



# Cortex XSOAR On-prem Documentation

Confidential - Copyright © Palo Alto Networks

## 1. Get Started with Cortex XSOAR

- 1.1. What is Cortex XSOAR?
- 1.2. Cortex XSOAR use cases
- 1.3. Cortex XSOAR architecture
- 1.4. Understand Cortex XSOAR licenses
- 1.5. Roles and responsibilities
- 1.6. Supported web browsers

## 2. Onboard and configure Cortex XSOAR

- 2.1. Plan your deployment
- 2.2. Onboarding checklist
- 2.3. Step 1. Install Cortex XSOAR
- 2.4. Step 2. Set up an engine
- 2.5. Step 3. Set up a remote repository
- 2.6. Step 4. Set up users and roles
- 2.7. Step 5. Install and configure content
  - 2.7.1. Install content packs
  - 2.7.2. Set up your use case with the Deployment Wizard
- 2.8. Post deployment
  - 2.8.1. User communication
    - 2.8.1.1. Configure notifications in Cortex XSOAR
    - 2.8.1.2. Customize system emails
  - 2.8.2. Configure system settings
    - 2.8.2.1. Configure server settings
    - 2.8.2.2. Configure security settings
- 2.9. Configure Cortex XSOAR

## 3. Cortex XSOAR Installation

- 3.1. Installation
- 3.2. System Requirements
- 3.3. High Availability for Cortex XSOAR
- 3.4. Install Cortex XSOAR from an OVA image
- 3.5. Install Cortex XSOAR from a VHD image
- 3.6. Post-installation
  - 3.6.1. Add the Cortex XSOAR license
  - 3.6.2. HTTPS with a signed certificate
  - 3.6.3. Load balancing for Cortex XSOAR
  - 3.6.4. Add or remove nodes in a cluster
  - 3.6.5. Scale up hardware resources
  - 3.6.6. Access logs and log bundles
  - 3.6.7. Open a support session
  - 3.6.8. Shut down Cortex XSOAR
- 3.7. Update Cortex XSOAR

## 4. Engines

- 4.1. What is an engine?
- 4.2. Engine requirements
- 4.3. Install an engine
  - 4.3.1. Engine air gap installation
  - 4.3.2. Docker
    - 4.3.2.1. Install Docker
      - 4.3.2.1.1. Install Docker distribution for Red Hat on an engine server
      - 4.3.2.1.2. Docker image security
      - 4.3.2.1.3. Docker FAQs
      - 4.3.2.1.4. Troubleshoot Docker issues
      - 4.3.2.1.5. Configure Docker pull rate limit
      - 4.3.2.1.6. Change the Docker installation folder
    - 4.3.2.2. Configure Docker integrations to trust custom certificates
    - 4.3.2.3. Docker hardening guide
  - 4.3.3. Podman
    - 4.3.3.1. Change container storage directory
    - 4.3.3.2. Install Podman
    - 4.3.3.3. Migrate From Docker to Podman

#### 4.3.3.4. Troubleshoot Podman

- 4.4. Manage engines
- 4.5. Upgrade an engine
- 4.6. Remove an engine
- 4.7. Configure engines
  - 4.7.1. Configure the engine to use a web proxy
  - 4.7.2. Configure the engine to call the server without using a proxy
    - 4.7.2.1. Use NGINX as a reverse proxy
    - 4.7.2.2. Configure an engine to use custom certificates
- 4.8. Use an engine in an integration
- 4.9. Run a script using an engine
- 4.10. Troubleshoot engines
- 4.11. Troubleshoot integrations running on engines

## 5. Remote Repository Management

- 5.1. Content management in Cortex XSOAR
- 5.2. Set up a private remote repository
- 5.3. Push content from a development tenant
- 5.4. Install content on a production tenant
- 5.5. Remote repository troubleshooting

## 6. Users and Roles Management

- 6.1. Users and roles in Cortex XSOAR
- 6.2. Roles management
  - 6.2.1. Role-based permissions
  - 6.2.2. Manage roles in the Cortex XSOAR tenant
- 6.3. User group management
- 6.4. Set up authentication
  - 6.4.1. Create users in Cortex XSOAR
  - 6.4.2. Authenticate users using SSO
  - 6.4.3. Set up Okta as the Identity Provider Using SAML 2.0
  - 6.4.4. Set up Azure AD as the Identity Provider Using SAML 2.0
- 6.5. User management
- 6.6. Configure a password policy

## 7. Marketplace

- 7.1. Cortex Marketplace
- 7.2. Content packs
- 7.3. Manage content packs
- 7.4. Set up your use case with the Deployment Wizard
- 7.5. Marketplace FAQs
- 7.6. Content pack update notifications
  - 7.6.1. Customize content pack notifications
- 7.7. Content pack contributions
  - 7.7.1. Create a content pack
  - 7.7.2. Resubmit a content pack

## 8. Integrations

- 8.1. Integration use cases
- 8.2. Configure integrations
- 8.3. Change the Docker image in an integration or script
  - 8.3.1. Connect your engine to an image registry
  - 8.3.2. Pull images from a private image registry
- 8.4. Manage credentials
- 8.5. Add an integration instance
  - 8.5.1. Fetch incidents from an integration instance
  - 8.5.2. Receive notifications on an incident fetch error
  - 8.5.3. Configure integration permissions
  - 8.5.4. Troubleshoot integrations

- 8.6. Integration commands in the CLI
- 8.7. Forward requests to long-running integrations

## 9. Incident configuration

- 9.1. Incident lifecycle
- 9.2. Incident Customization
  - 9.2.1. Use incident context data
  - 9.2.2. Create an incident type
  - 9.2.3. Create an incident field
    - 9.2.3.1. Incident field trigger scripts
    - 9.2.3.2. Create dynamic fields
    - 9.2.3.3. Troubleshoot incident fields
  - 9.2.4. Incident layout customization
    - 9.2.4.1. Examples of using scripts in incident layouts
- 9.3. Classification and mapping
  - 9.3.1. Create an incident classifier
  - 9.3.2. Create an incident mapper
- 9.4. Set up incident mirroring
- 9.5. Incident deduplication in Cortex XSOAR
- 9.6. Pre-process rules
- 9.7. Use post-processing scripts in an incident
- 9.8. Customize incident close reasons
- 9.9. Configure inline value fields
- 9.10. Export an incident to CSV using the UTF8-BOM format

## 10. Playbooks

- 10.1. What is a playbook?
- 10.2. Playbook development checklist
- 10.3. Plan your playbook
- 10.4. Develop your playbook
  - 10.4.1. Playbook tasks
    - 10.4.1.1. Playbook inputs and outputs
    - 10.4.1.2. Create a section header
    - 10.4.1.3. Create a standard task
    - 10.4.1.4. Create a conditional task
    - 10.4.1.5. Create a communication task
    - 10.4.1.6. Configure script error handling in a playbook
  - 10.4.2. Customize a playbook for a phishing use case example
- 10.5. Customize your playbook
  - 10.5.1. Configure general playbook settings
  - 10.5.2. Customize the SOC name
  - 10.5.3. Configure a sub-playbook
  - 10.5.4. Filter and transform data
    - 10.5.4.1. Filter considerations, categories, and built-in filters
    - 10.5.4.2. Transformer considerations, categories, and built-in transformers
  - 10.5.5. Extract indicators
  - 10.5.6. Extend context
  - 10.5.7. Set and update incident fields
  - 10.5.8. Playbook polling
- 10.6. Scripts
  - 10.6.1. Create a script
- 10.7. Debug your playbook
  - 10.7.1. Troubleshoot playbook performance
- 10.8. Manage playbook content
- 10.9. Add ad-hoc tasks to a Work Plan as part of your investigation
- 10.10. Best practices

## 11. Lists

- 11.1. What is a list?
- 11.2. Create a list
- 11.3. List commands
- 11.4. Use cases: JSON lists
- 11.5. Transform a list into an array

## 12. Jobs

- 12.1. Manage jobs
- 12.2. Create a time triggered job
- 12.3. Create a job triggered by a delta in a feed
- 12.4. Create jobs to process indicators example

## 13. SLAs

- 13.1. SLAs in Cortex XSOAR
- 13.2. Configure an SLA in an incident type
- 13.3. Configure Timer/SLA fields
- 13.4. Configure a playbook to run Timers/SLAs
- 13.5. Automate changes to incident fields using SLA scripts
- 13.6. Create SLA scripts
- 13.7. Use SLA and Timer field commands manually in the CLI
- 13.8. Configure the Global Risk Threshold
- 13.9. Search incidents for Timer/SLAs

## 14. Dashboards and Reports

- 14.1. Dashboards
  - 14.1.1. Dashboard actions
  - 14.1.2. Manage dashboards
- 14.2. Reports
  - 14.2.1. Manage reports
    - 14.2.1.1. Report scheduling examples
  - 14.2.2. Configure the timezone in a report
  - 14.2.3. Troubleshoot script timeout for reports
- 14.3. Widgets
  - 14.3.1. Widget customization
  - 14.3.2. Create a widget using the widget builder
    - 14.3.2.1. Create a widget using the widget builder examples
  - 14.3.3. Create a custom widget using a JSON file
  - 14.3.4. Create a custom widget using a script
    - 14.3.4.1. Script-based widget examples
  - 14.3.5. Create a widget from an incident
  - 14.3.6. Create a widget from an indicator
  - 14.3.7. Edit a widget
  - 14.3.8. Add a widget in the War Room
  - 14.3.9. Saved By Dbot (ROI) Widget

## 15. Incidents and indicators investigation

- 15.1. Incidents
- 15.2. Incident management
  - 15.2.1. Search for incidents
  - 15.2.2. Create an incident
  - 15.2.3. Export incidents
- 15.3. Investigate an incident
  - 15.3.1. Retain incidents
  - 15.3.2. Limit access to investigations using access control
  - 15.3.3. Incident Tasks
  - 15.3.4. Use the War Room in an investigation
  - 15.3.5. Schedule a command in the War Room
  - 15.3.6. Run commands in the CLI
  - 15.3.7. Evidence Handling
  - 15.3.8. Use the Work Plan in an investigation
  - 15.3.9. Link incidents
  - 15.3.10. Create an incident summary report
- 15.4. Manage indicators
  - 15.4.1. Query indicators
  - 15.4.2. View indicator relationships in an investigation

## 16. Threat Intel Management

- 16.1. Get started with Threat Intel Management

- 16.1.1. What is Threat Intel Management?
- 16.1.2. Threat Intel Management use cases
- 16.1.3. Indicator concepts
- 16.1.4. Indicator lifecycle
- 16.1.5. Roles and responsibilities in Threat Intel Management
- 16.2. Indicator configuration
  - 16.2.1. Customize indicator types, fields, and layouts
    - 16.2.1.1. Create an indicator type
      - 16.2.1.1.1. Indicator type profile
      - 16.2.1.1.2. File indicators
      - 16.2.1.1.3. Formatting scripts
      - 16.2.1.1.4. Enhancement scripts
      - 16.2.1.1.5. Reputation scripts
      - 16.2.1.1.6. Reputation commands
      - 16.2.1.1.7. Map custom indicator fields
    - 16.2.1.2. Create an indicator field
      - 16.2.1.2.1. Indicator fields structure
      - 16.2.1.2.2. Indicator field trigger scripts
    - 16.2.1.3. Indicator layout customization
  - 16.2.2. Indicator classification and mapping
  - 16.2.3. Indicator extraction
    - 16.2.3.1. Indicator extraction modes
    - 16.2.3.2. Create indicator extraction rules for an incident type
    - 16.2.3.3. Set the indicator extraction mode for a playbook task
    - 16.2.3.4. Disable indicator extraction for scripts or integrations
    - 16.2.3.5. Troubleshoot indicator extraction
  - 16.2.4. Configure the indicator timeline
  - 16.2.5. Configure indicator expiration
  - 16.2.6. Configure Threat Intel feed integrations
  - 16.2.7. Configure Threat Intelligence Management playbooks to process indicators
  - 16.3. Export indicators
  - 16.4. Customize Threat Intel Reports
    - 16.4.1. Create a Threat Intel Report type
    - 16.4.2. Create a Threat Intel Report field
    - 16.4.3. Create a Threat Intel Report layout
  - 16.5. Indicator management
    - 16.5.1. Query indicators with Unit 42 Intel data
  - 16.6. Indicator investigation
    - 16.6.1. Indicator verdict
    - 16.6.2. Extract and enrich an indicator
    - 16.6.3. Expire an indicator
    - 16.6.4. Manage indicator relationships
    - 16.6.5. Delete and exclude indicators
    - 16.6.6. Investigate files using sample analysis
    - 16.6.7. Use sessions and submissions in your investigation
  - 16.7. Manage Threat Intel Reports

## 17. Troubleshoot

  - 17.1. System diagnostics
  - 17.2. View service limit errors and warnings in the Guard Rails page
  - 17.3. Logs
  - 17.4. Integration logs
  - 17.5. Management audit logs
  - 17.6. Configure management audit notification forwarding

## 18. Reference

  - 18.1. Cortex XSOAR concepts
  - 18.2. How to search in Cortex XSOAR
  - 18.3. How to use markdown in Cortex XSOAR
  - 18.4. User details and preferences
  - 18.5. Server configurations
  - 18.6. New user FAQ
  - 18.7. Telemetry in Cortex XSOAR
  - 18.8. Product support lifecycle
  - 18.9. Keyboard shortcuts
  - 18.10. Cortex XSOAR navigation cheat sheet

## 1 | Get Started with Cortex XSOAR

### Abstract

View information about how to get started with Cortex XSOAR On-prem such as architecture, roles and responsibilities, and licenses.

Before diving in, understand Cortex XSOAR functionality and how it integrates with your needs. Review the available licenses, service limits, and other key details to optimize your Cortex XSOAR experience from the start.

### 1.1 | What is Cortex XSOAR?

#### Abstract

Learn about Cortex XSOAR features.

*Alert Alert Exclusion Analytics behavioral indicators of compromise Attack Surface Management Behavioral indicators of compromise Bring Your Own Machine Learning Broker Virtual Machine Broker Virtual Machine Fully Qualified Domain Name Causality Chain Causality Group Owner Causality View Cloud Detection and Response Cortex Copilot Cortex Data Model Cortex Query Language Dataset Elasticsearch Filebeat Endpoint Detection and Response Endpoint Protection Platform Exception Exception vs Alert Exclusion Extended Detection and Response External Dynamic List Filebeat Forensics Fully Qualified Domain Name Identity Threat Detection and Response Incident Indicators of compromise IT Metrics Dashboard Managed Threat Hunting Management, Reporting, and Compliance Master Boot Record Protection MITRE ATT&CK Framework Coverage Dashboard Next-Generation Firewall Notebooks On-write File Protection Playbook Prisma ScriptSecurity Orchestration, Automation, and Response Security Information and Event Management Threat Intelligence Platform User and Entity Behavior Analytics Unified Extensible Firmware Interface Protection Virtual Machine Vulnerability Assessment Windows Event Collector XSIAM Command Center*

Cortex XSOAR is the industry's first extended security orchestration and automation platform that simplifies security operations by unifying automation, case management, real-time collaboration, and threat intel management.

Cortex XSOAR ingests aggregated alerts and indicators of compromise (IOCs) from detection sources, such as security information and event management (SIEM) solutions, network security tools, threat intelligence feeds, and mailboxes, before executing automatable, process-driven playbooks to enrich and respond to these incidents. These playbooks coordinate across technologies, security teams, and external users for centralized data visibility and action.

With a Threat Intel Management license, Cortex XSOAR provides a Threat Intelligence Platform with actionable threat data from Unit 42. You can identify and discover new Malware families or campaigns and create and disseminate strategic intelligence reports.

For existing Cortex users, XSOAR is easily integrated into other Cortex solutions and is delivered from the same platform.

Terjadi error.

---

Cobalah menonton video ini di  
[www.youtube.com](http://www.youtube.com), atau aktifkan  
JavaScript jika dinonaktifkan di browser  
Anda.

## Why Cortex XSOAR?

- Improve SOC Efficiency by Automating Incident Response

Automate incident response workflows and repetitive tasks to free up analysts to focus on the most critical incidents with Cortex XSOAR. Use predefined playbooks or easily customize your own to automate SOC use cases such as indicator enrichment, alert deduplication, phishing response, ransomware response, threat intelligence feed management, malware investigation, and even IT operations such as employee onboarding and offboarding.

- Experience Better Performance, Reliability, and Scalability

Cortex XSOAR supports future growth, with rapid deployment to accelerate ROI. Fully integrated into the Cortex platform, Cortex XSOAR is delivered through a unified user interface for ease of use and consistency in workflow management.

- Ingest, Search, and Query All Security Alerts

When complex, real-time investigations require analyst intervention, ensure analysts have quick access to investigation data. Cortex XSOAR accelerates incident response by unifying incident and indicator data from multiple sources on a single easy-to-search platform.

- Improve Investigation Quality by Working Together

Collaborative investigation features provide a powerful toolkit to help analysts assist each other, run real-time security commands, and learn from each incident with auto-documentation of all actions. An ML-driven assistant learns from actions taken in the platform and offers guidance on analyst assignments and commands to execute actions.

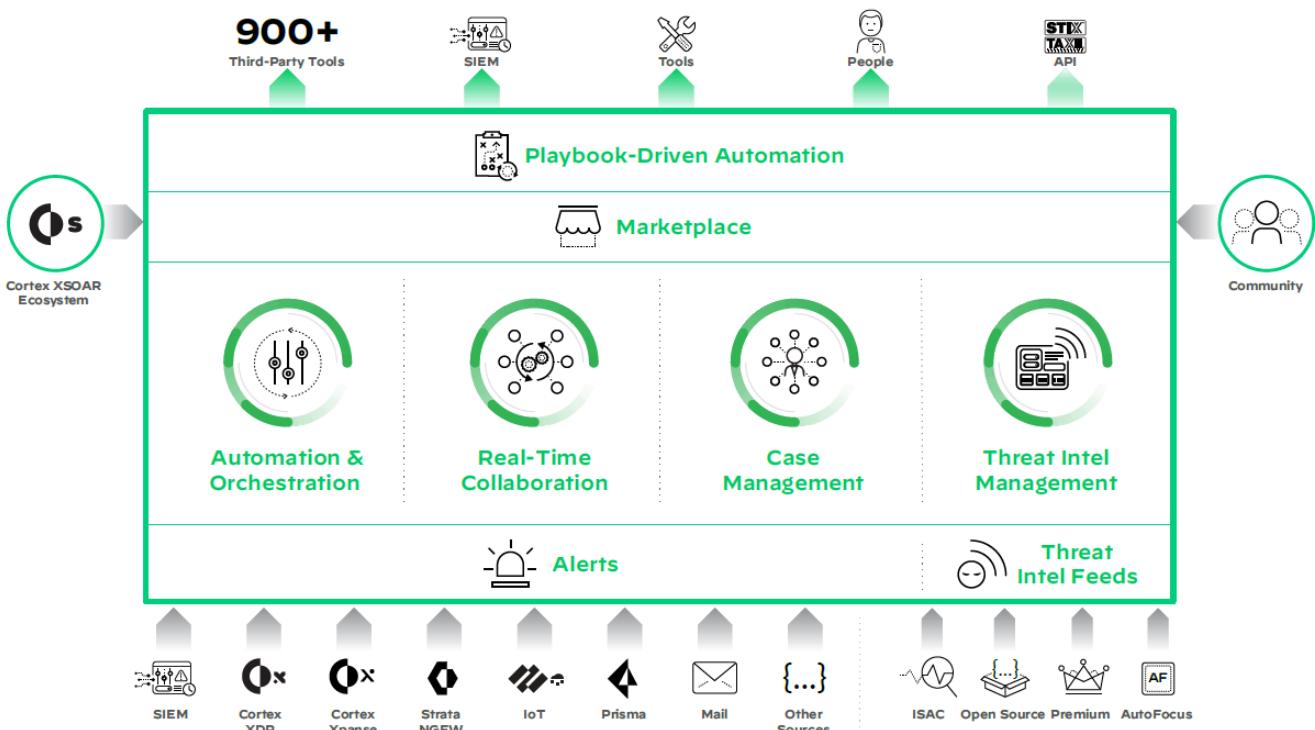
- Act on Threat intelligence with Agility and Confidence

Unify aggregation, scoring, and sharing threat intelligence with playbook-driven automation with native threat intelligence management. The built-in, high-fidelity threat intelligence can be boosted by layering additional third-party threat intel to better reveal and prioritize critical threats.

## How Cortex XSOAR Works

Cortex XSOAR ingests aggregated alerts and indicators of compromise (IoCs) from detection sources such as security information and event management (SIEM) solutions, network security tools, threat intelligence feeds, and mailboxes, before executing automatable, process-driven playbooks to enrich and respond to these incidents. These playbooks coordinate across technologies, security teams, and external users for centralized data visibility and action.

For existing Cortex users, XSOAR is easily integrated into other Cortex solutions and is delivered from the same platform. Cortex XSOAR ingests alerts from third-party products and Threat intel feeds and by installing content packs, you can automate the investigation and response process.



## 1.2 | Cortex XSOAR use cases

### Abstract

Recommended ways to automate your SOC in Cortex XSOAR.

### How Automation Makes Life Easier in the SOC

- Accelerate incident response:** Replacing low-level manual tasks with automations, security automation can shave off large chunks from incident response times while improving accuracy and analyst satisfaction.
- Standardize and scale processes:** Through stepwise, replicable workflows, security automation can help standardize incident enrichment and response processes that increase the baseline quality of response and is primed for scale.
- Unify security infrastructures:** A SOAR platform like Cortex XSOAR can act as a connective fabric that runs through hitherto disparate security products, providing analysts with a central console from which to action incident response.
- Increase analyst productivity:** Since low-level tasks are automated, and processes are standardized, analysts can spend their time in more important decision-making and charting future security improvements rather than getting mired in grunt work.
- Leverage existing investments:** By automating repeatable actions and minimizing console switching, security orchestration enables teams to coordinate among multiple products easily and extract more value out of existing security investments.
- Streamline incident handling:** By applying automation to incident ticket management via integrations with key ITSM vendors such as ServiceNow, Jira, and Remedy, as well as communication tools such as Slack, security teams can speed up incident handling and closure. Incidents can also be distributed automatically to the respective stakeholders based on predefined incident types.
- Improve overall security posture:** The sum of all aforementioned benefits is an overall improvement of the organization's security posture and a corresponding reduction in security and business risk.

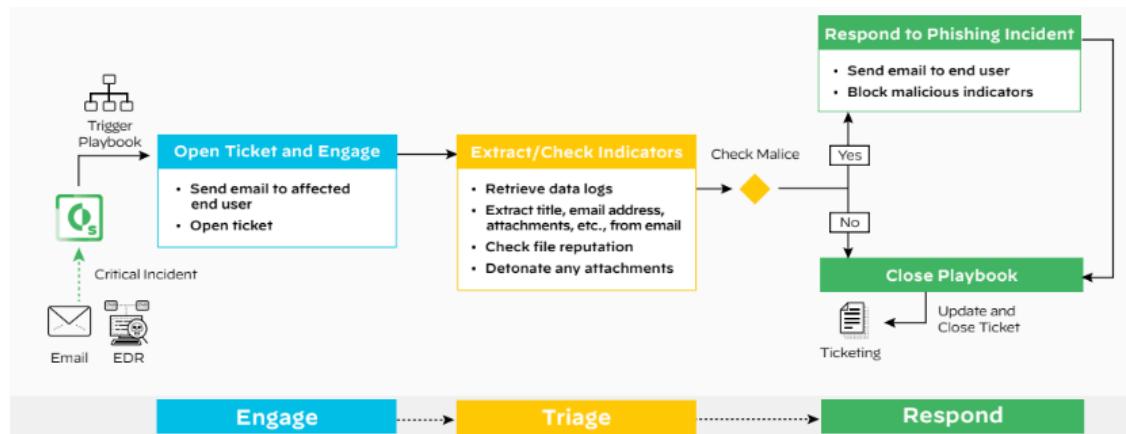
The following examples demonstrate how to automate repetitive tasks and streamline your security incident response processes for maximum efficiency. These are tried and tested automation use cases that have been leveraged by our own Palo Alto Networks SOC, ITOps, and our customers to gain operational efficiencies and scale.

#### Phishing response

Phishing emails are pernicious and one of the most frequent, easily executable, and harmful security attacks organizations still face today. Responding to a phishing email involves switching between multiple screens to coordinate a response, including responding to end users. These tasks can easily take around 45 minutes of your time per incident.

In Cortex XSOAR phishing playbooks can help you execute repeatable tasks at machine speed, identify false positives, and prime your operations for standardized phishing responses at scale. More importantly, the quick identification and resolution of false positives gives you more time to deal with genuine phishing attacks and prevents them from slipping through the cracks. Cortex XSOAR has machine learning intelligence built in, allowing you to "train" the phishing engine to recognize future phishing attacks.

#### Workflow Automating phishing response



#### Engage

Cortex XSOAR can ingest suspected phishing emails as incidents from various detection sources such as SIEMs, EDRs, email security, or phishing services. If you aggregate all suspected phishing emails in a common mailbox, these emails can be ingested as incidents via a mail listener integration.

When the email is ingested, a playbook is triggered, going through the steps to automate enrichment and response. To keep end users updated, the playbook sends an automated email to the affected user and lets them know the suspected phishing email is being investigated.

#### Triage

In the triage process, the playbook can perform extraction and enrichment of indicators of compromise (IoC) extraction.

By investigating the email, such as title, email address, and attachments, the playbook assigns incident severity by cross-referencing these details against external threat databases. Following this, the playbook extracts IoCs from the email and checks for any reputational red flags from threat intelligence tools that your team uses.

When enrichment is finished, the playbook checks if any malicious indicators are found. Based on this check, different response branches can arise.

#### Respond

Different playbook branches execute depending on whether malicious indicators were detected in the suspected phishing email.

If malicious indicators are detected, the playbook sends an email to the affected user with further instructions. The playbook also scans all organizational mailboxes/endpoints to identify other instances of that email and deletes all instances to avoid further damage. Finally, the playbook adds the malicious IoCs to block lists/watchlists on the SOC's other tools. If no malicious indicators are detected, there are still precautions to be taken before confirming that the email is harmless. The playbook checks if there are any attachments in the email that can be sent for detonation in a sandbox.

Threat intel analyses are then presented in an incident war room for the analyst to do a final check. Once the analyst is satisfied that the email isn't malicious, the playbook sends an email to the affected user apprising them of the false alarm. The incident ticket is marked closed.

You easily eliminate 10 or more steps your security team has to touch, saving them hours responding to phishing alerts.

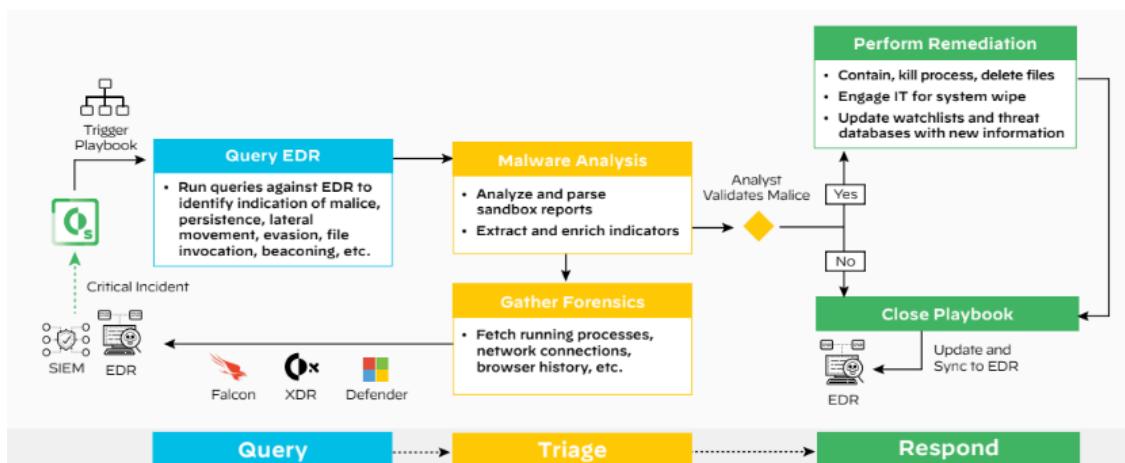
For more information, see the Phishing content pack.

#### Malware investigation and response

Determining if alerts for unknown activity from your endpoint security tools are malicious often involves coordinating between multiple security tools. It's a cross-referencing nightmare with multiple consoles open simultaneously and valuable time spent performing repetitive data collection tasks. Decreasing the investigation and response time means less dwell time for malicious activity to wreak havoc in your network.

Automation playbooks can unify processes across SIEMs and endpoint tools in a single workflow, performing repetitive steps before bringing analysts in for important decision-making and investigative activities.

#### Workflow automating malware investigation and response



#### Query

An incoming endpoint security alert triggers a series of playbooks that automatically query for evidence of malice, such as:

- Is there evidence of an attempted lateral movement?
- Is there evidence of persistence? Did the process create any scheduled jobs? Did it write to the registry? Was Autorun updated?
- Is the file digitally signed?
- How did the file get onto the machine?
- What is the process execution chain?
- What triggered the execution of the file?
- Was the network traffic blocked at the firewall?

The findings are presented in the incident for an analyst review, eliminating the need to manually collect and piece the evidence together.

#### Triage

Detonating suspicious files in sandboxes for malware analysis is an ever-present and important investigative step during incident response. However, it's taxing for security analysts to coordinate across consoles while executing this repetitive task because malware analysis tools are isolated from other security

products. Transferring results from one console to another for documentation is time-consuming and increases the chances of errors.

In this scenario, playbooks can be run concurrently to automate the file detonation process as an isolated workflow or with other enrichment activities. Playbooks can parse through the results of the sandbox detonation and be configured to run specific queries against the EDR tool. As playbooks document the result of all actions on a central console, the need for manual post-incident documentation is also eliminated.

Another aspect of malware analysis involves gathering forensic data, such as all the processes running on a machine, which can be automated. During an investigation, it is critical to understand what is happening on the endpoint when the alert is detected. Sometimes it can be minutes or even hours before an analyst looks at a detected alert, at which point the state of the endpoint is likely different, which makes the re-creation of what happened more challenging. These playbooks can communicate continuously with the same endpoint tools to run queries on processes, network connections, browser history, etc. to track incident status.

#### Respond

If the file is malicious, the playbook updates relevant watchlists/block lists with that information. From here, the playbook can branch into other actions such as quarantining infected endpoints, killing malicious processes, removing infected files, opening tickets, and reconciling data from third-party threat feeds.

After the queries have been run, the playbook updates the endpoint tool database with new indicator information, so repeat offenses are eliminated.

For more information, see the Malware Investigation and Response content pack.

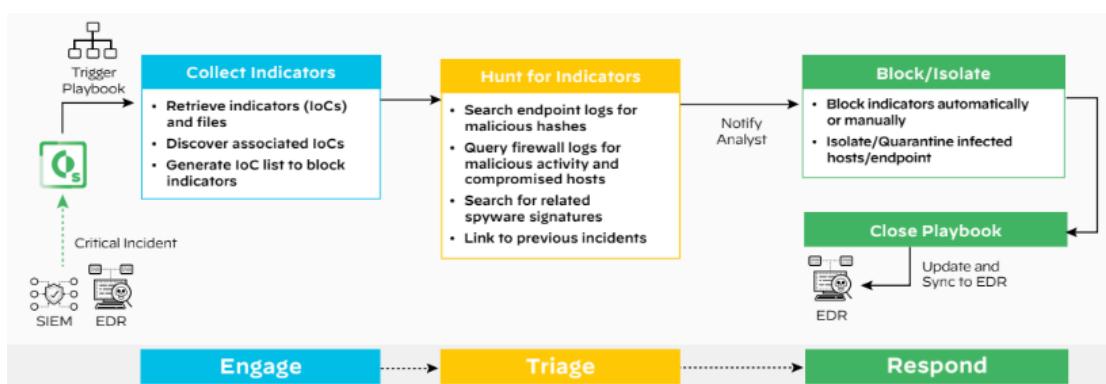
#### Zero-day threat response

Zero-day threats and ransomware breaches are constantly in the news, such as SolarWinds SUNBURST, HAFNIUM Microsoft zero-day exploit, Nobelium threat actor, Kaseya supply chain ransomware attack, and Log4j vulnerability.

Every time a critical vulnerability is reported, it's an all-hands-on-deck effort to ensure that your organization is not exposed to the potential exploits of the vulnerability. Your executive team likely has heard it in the news and needs an assessment of exposure for the organization. Speed is essential if potential malicious activity is detected.

Automation can help you quickly process, collect, hunt for indicators, and perform quick response actions upon finding IoCs.

#### Workflow automating zero-day threat response



#### Engage and Triage

In the case of a breach alert, the process of retrieving and discovering associated IoCs is as repetitive as it is important. Your analysts risk getting mired in this work while the attack continues to manifest. Isolated security tools result in a struggle to reconcile threat data across platforms to get an overall understanding of malicious activity and spread.

By running this playbook at the outset of incident response, your team can query endpoints, firewalls, and other incidents in seconds, avoiding wasted time that can be used towards locking down defenses.

#### Respond

The playbook executes initial response actions based on indicator malice. For example, the playbook can block indicators, isolate, or quarantine infected hosts, or feed malicious indicators back into threat intelligence databases and tool watchlists to avoid future attacks using the same indicators.

We provide specific rapid breach response playbooks for high-profile breaches to help you speed up your investigation efforts. For more information, see the Rapid Breach Response content pack.

#### Remote user access provisioning

Remote work has become the norm, and your business is increasingly moving to the cloud, which has increased the threat exposure and attack surface your team has to account for.

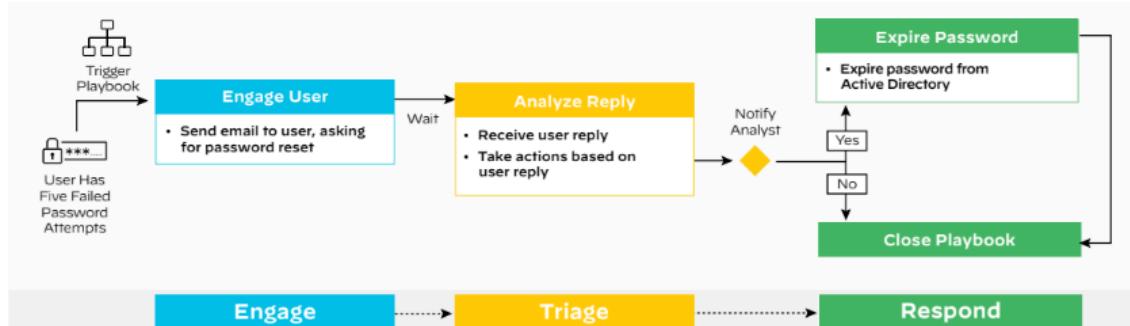
Automation can play a role in many areas, including aiding investigations into unsuccessful login attempts and other access violations, monitoring the health of VPNs, and updating dynamic allow/deny IP domain lists to ensure business continuity.

## Failed user logins

Despite the increased sophistication of security measures, it's possible for attackers to brute-force their way into accounts by obtaining the email address and resetting the password. This behavior is difficult to preempt, as there are high chances of it being innocuous (a genuine employee resetting their password). Constant communication between you and end users to separate the anomalies from the usual is critical.

At user-defined triggers (such as five failed login attempts), a playbook can execute and verify whether the case is genuine or malicious.

### Workflow automating multiple failed logins alert response



#### Engage

The playbook sends an automated email to the affected user, notifying them of the five failed login attempts and asking them to confirm that the behavior was theirs. The email requests the user to reply with "Yes/No," and spells out the ensuing action for each response.

#### Triage

You can analyze the replies to automated emails and execute different playbook branches.

#### Respond

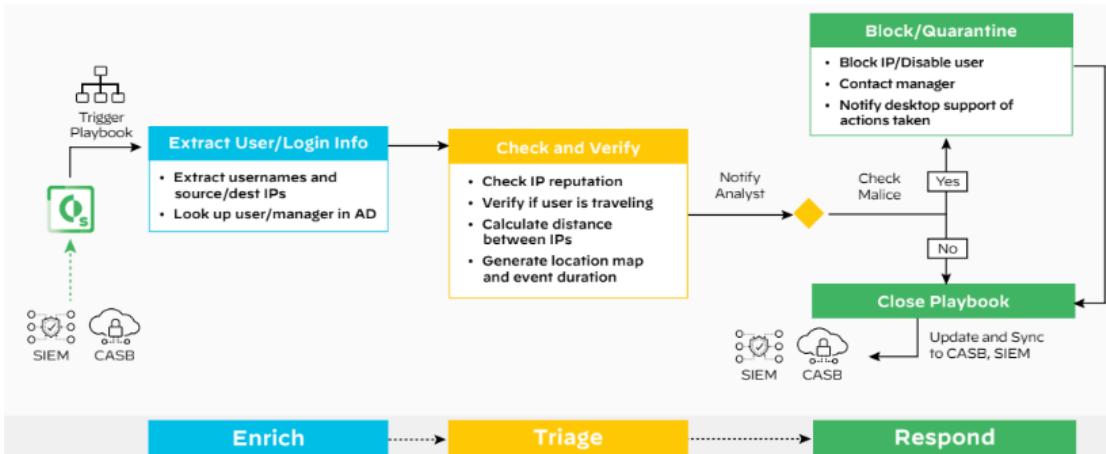
If the end-user behavior is genuine, the playbook resets the password on Active Directory and sends a new email to the affected user with revised login credentials.

If the end users confirm they made the failed login attempts, the playbook sends a new email notifying them of these account takeover attempts. The playbook can also execute investigative actions, such as extracting the IP/location where the failed attempts were made and quarantining the affected endpoint.

#### Logins from unusual locations

With the ability to work from anywhere, it's difficult to spot a malicious access attempt from a genuine case of employee access from multiple geographical locations. With increased cloud adoption, multiple sources of geographical presence exist to verify, heaping more work on your security team and presenting a window of opportunity to attackers.

### Workflow automating suspicious login activity response



To combat "impossible travel" (simultaneous logins from distant locations), which is flagged by the playbook and the trigger action, a modified failed user login playbook would enrich IP information by checking the IP address reputations using threat intelligence sources and calculating the distance between IPs, generate a location map and login time duration. When the analyst decides the activity is malicious, the playbook executes a series of containment steps, such as disabling user accounts, blocking malicious IPs at the firewall, and notifying IT Support of actions taken.

For more information, see the

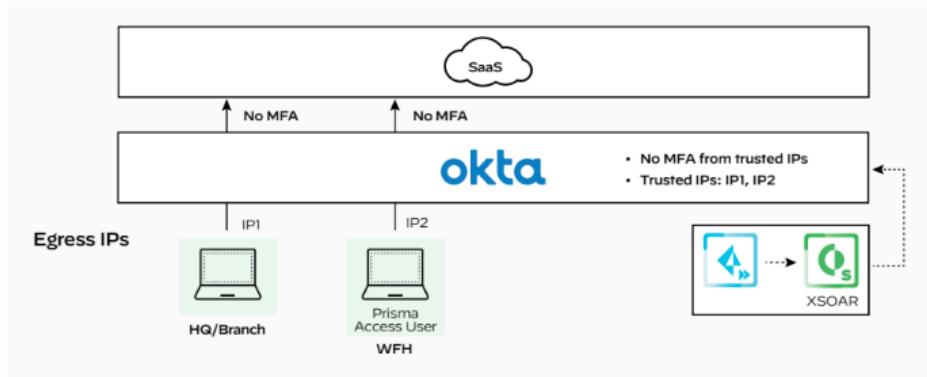
### Enforcing multifactor authentication

Multifactor authentication (MFA) is often required when end users connect from untrusted or unknown IPs. Trusted network IPs are defined in identity and access management (IAM) systems like Okta, so users connecting from a trusted network such as HQ or branch offices do not need MFA, but if they connect from a coffee shop, they would be required to authenticate with MFA.

However, in SASE solutions such as Prisma Access, due to auto-scaling or provisioning of new locations, the list of assigned IPs for an enterprise often changes. So, if these egress IPs are not listed or added to their IAM, any user connecting to these new IPs to access their software-as-a-service (SaaS) applications, even if they are on a trusted network, will be required to use MFA. This can result in an inconsistent end-user experience.

With the integration between Cortex XSOAR and Prisma Access, an automated playbook can “listen” to auto-scaling and new provisioning events, immediately pick up the new list of Prisma Access egress IPs, and automatically update the IAM. This provides a seamless login experience for users connecting from a trusted network.

### Enforcing multifactor authentication



### Monitoring VPN tunnel health

On a security team's busy day, there is no time to proactively monitor for potential connectivity downtime as the staff is usually busy firefighting and triaging critical incidents. Among other things, this makes it difficult to keep track of the health status of all VPN tunnels to ensure 100% uptime for end users.

In this case, an automated VPN tunnel monitoring playbook can be scheduled to poll Prisma Access connection statuses regularly and send a Slack alert to the security or ITOps team if a tunnel is down.

### Monitoring VPN tunnel status

With the new normal of remote work, these automation use cases can help streamline operations and help your ITOps and security teams scale to address remote access security incidents and keep track of remote activity.

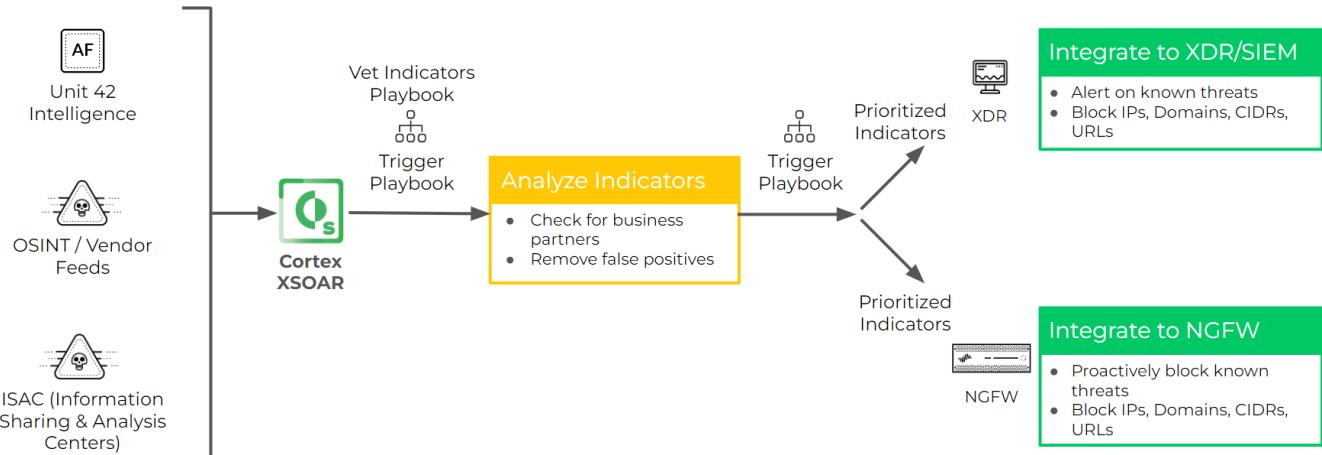
For more information, see the Prisma Access - Connection Health Check playbook in the Palo Alto Networks - Strata Cloud Manager content pack.

### Threat Intel Management

As you ingest alerts, you can automatically enrich them with the latest threat intel from your feeds. This gives you context for how external and emerging threats are impacting your environment and also helps you quickly hone in on critical threats.

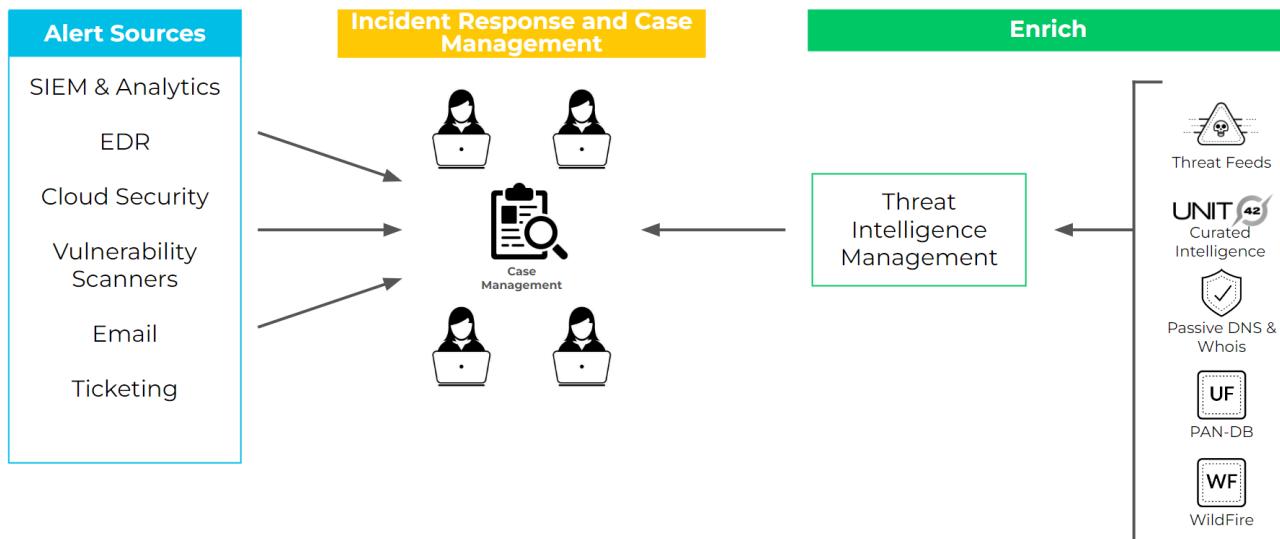
### Proactive blocking of threats

The indicators collected from many different threat feeds need to be aggregated, normalized, scored, and prioritized before they can be pushed to enforcement points. A threat intel platform can automate these feed management functions, ensuring that your external dynamic lists (EDLs) are always up to date per the latest threats.

**COLLECT****MANAGE****TAKE ACTION**

Continuous incident enrichment

As you investigate incidents, you need threat intel context on associated indicators. Curated threat intelligence, such as those from Unit 42 Intel threat research that comes packaged with the Threat Intel Management (TIM) module, helps you automate indicator enrichment, giving your analysts early warning and rich context into emerging threats in the wild that might be impacting your network.



Generating Weekly OSINT (Open Source Intelligence) and Other Threat Reports

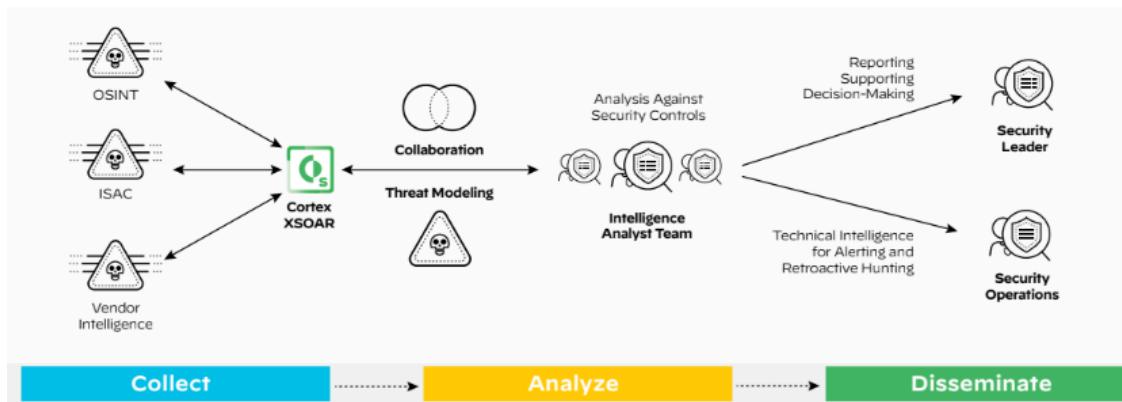
Your threat intel team produces and disseminates threat intelligence reports to various business units/stakeholders to keep them up to date on the latest threats targeting their industry. Most intelligence is still shared via unstructured formats such as email and blogs, so your threat analysts may go through hours of manual work aggregating and digging for known malware families, curated news, and industry-specific threats, as well as providing analyses on why each threat is relevant to the business. Cortex XSOAR TIM provides automated workflows and a central repository for intelligence analysts to create, collaborate, and share curated intelligence reports with stakeholders.



## COLLECT → MANAGE → TAKE ACTION

### External threat landscape modeling

Threat intelligence teams need to understand the details of attacks and how their organizations may be vulnerable. The intel team builds profiles of threat actors, identifying if there are related attacks and which techniques and tools the threat actor used. This information is shared with stakeholders, including security operations and leadership.



### MITRE ATT&CK mapping

The MITRE ATT&CK framework was created to organize the real-world industry observations of threat actors into a standardized language of tactics, techniques, and procedures (TTPs) to help organizations share information and recommendations, which can be used to harden security programs.

Given the breadth and depth of the framework, understanding, consuming, and mapping the tactics and techniques within the MITRE ATT&CK framework into reliable and usable remediation steps can be a complicated and time-consuming task.

The set of playbooks in the MITRE ATT&CK - Courses of Action content pack helps you automatically map your incident response to MITRE ATT&CK techniques and sub-techniques in an organized and automated manner, which ensures your organization not only blocks specific reported IoCs but also takes a more holistic approach to preventing future attacks. With Cortex XSOAR, you can leverage prebuilt automation playbooks to cross-reference every incident with the tactics and techniques of the MITRE ATT&CK framework.

This content pack provides manual or automated remediation of MITRE ATT&CK techniques and kill chain. Security analysts choose the techniques relevant to their security program and run the prebuilt playbooks that leverage expert remediation workflows. This can be found in the built-in MITRE ATT&CK dashboard.

When used with Unit 42's feed ingesting Actionable Threat Objects and Mitigations (ATOMs), your team gets notified when there is a new threat actor report, with recommendations for immediate remediation action. This allows your security team to apply industry threat response protocols and best practices to block specific reported IoCs and take a more holistic approach to prevent future attacks.

### Cloud security incident response

In cloud security, there are many infrastructures and products to deal with. The security of your cloud is often a shared responsibility between you, your cloud service provider, and other teams. Cloud SecOps teams report that cloud security incidents are treated on a case-by-case basis, and the remediation process is high-touch and manual. There is often no correlation between cloud platforms and on-premises security.

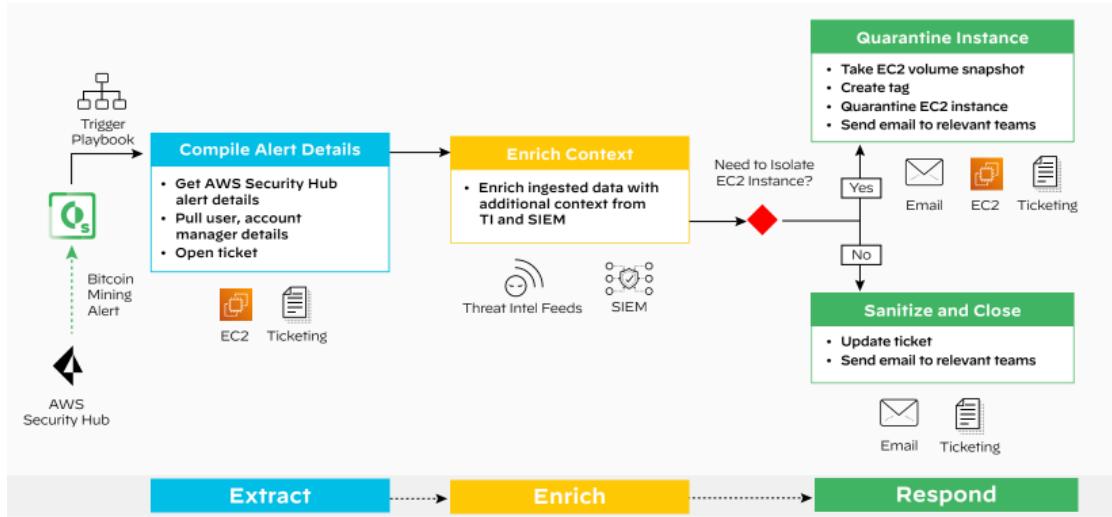
Cortex XSOAR can unify processes across multi-cloud and on-premises security infrastructures, providing your security teams with a single console to execute the incident response. We also integrate with cloud-based identity management tools, enabling role-based and keyless deployment of services

without the need for credential management.

#### Cloud threat detection

With the move towards digital currency and the acceptance of cryptocurrency for financial transactions, cryptojacking isn't declining anytime soon.

For example, you may automate a response to a cryptomining alert. Cortex XSOAR can ingest cloud security alerts from AWS, Google Cloud, Microsoft Azure, or Prisma Cloud to fully or partially automate incident response.



#### Extract

The playbook extracts indicators (IPs, URLs, hashes, and so on) from the incident data. It can also open a ticket for the incident.

#### Enrich

The playbook enriches indicators with reputation data from threat intelligence tools that the SOC uses. It also enriches the ingested data with additional context from SIEMs and other non-cloud-based event management tools to identify the full extent of the suspected attack. The playbook checks if the indicators are identified as malicious.

#### Respond

The playbook obtains the instance and security group details and security group details, takes volume snapshots, and creates a tag for the EC2 instance to be isolated. These steps are classic digital incident response and forensics actions, but carried out in the cloud. What we are doing is moving the EC2 instance into a separate virtual PC (VPC) as we would on a virtual LAN (VLAN) in the on-premises world, getting a list of running processes, analyzing the results, and also sending an email to the analyst for review.

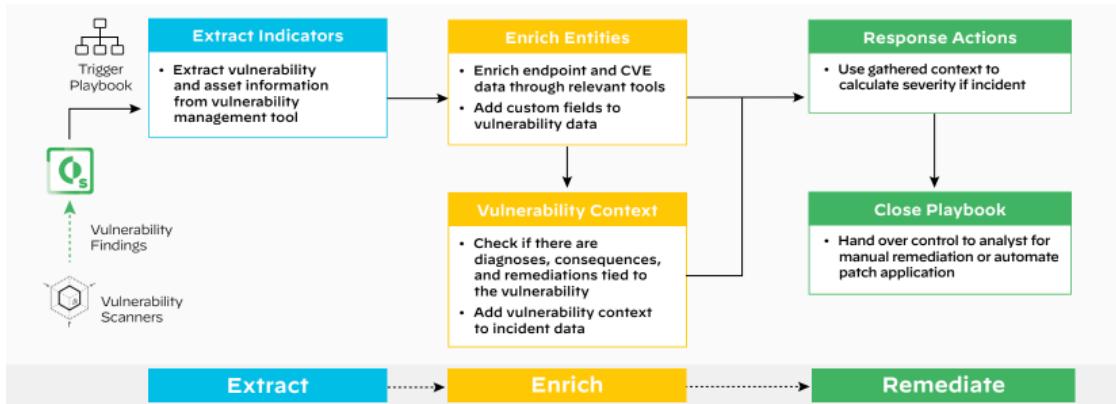
If the indicators are not identified as malicious, the playbook can ask a security analyst to review the information and verify that it's not dangerous before closing the incident as a false positive.

Automation cuts analyst time and increases responses by eliminating manual tasks, inter-team coordination, and security product changes. Also, you can enforce standard operating procedures across different teams for cloud security incident response. Other automation use cases include automating incident response for common cloud security incidents like password and security group misconfigurations, access key compromises, unpatched vulnerabilities, and unusual activity like port scans/port sweeps. View more automation content packs in Marketplace.

#### Vulnerability Management

Vulnerability management is a strategically important process that covers both the proactive and reactive aspects of security operations. Since vulnerability management encompasses all computing and internet-facing assets, security teams often grapple unsuccessfully with correlating data across environments, spending too much time unifying context and not enough time remediating the vulnerability.

Security orchestration playbooks can automate enrichment and context addition for vulnerabilities before passing them to the appropriate teams for patch remediation. This maintains a balance between automated and manual processes by ensuring that analyst time is not spent executing repetitive tasks but on making critical decisions and drawing inferences.



## Extract

The playbook ingests asset and vulnerability information from a vulnerability management tool such as Tenable or Qualys. The related information from the incident is extracted, and related indicators are created and enriched.

The playbook then enriches endpoint and CVE data through relevant tools. It also adds custom fields to the incident if the newly gathered data requires them.

To provide the analyst with a richer vulnerability context, the playbook queries the vulnerability management tool for any diagnoses, consequences, and remediations tied to the vulnerability. If any vulnerability context is found, it's added to the incident data. Based on the gathered context, the playbook then calculates the severity of the incident.

## Remediate

Playbooks can also use vulnerabilities to inform threat priority and initiate the patching process. Response actions can be taken by playbooks, including:

- Checking if assets (IP, domain, or certificate) associated with the issue are excluded in the exclusions list and closing the incident automatically.
- Enriching indicators and calculating the severity of the issue.
- Adding associated assets (IP, domain, or certificate) to the exclusions list.
- Tagging associated assets and updating the status of the issue.

The playbook now hands over control to the security analyst for manual investigation and remediation of the vulnerability.

## Attack surface management

Vulnerability scanners are great for monitoring your known assets, but what about your unknown assets? To uncover these blind spots, your organization needs an automated attack surface management (ASM) solution like Cortex Xpanse that continuously discovers and monitors the entirety of IPv4 space to provide a complete and accurate inventory of your global internet-facing assets and misconfigurations.

Together with Cortex XSOAR, Xpanse enables you to automate the identification and remediation of web-facing exposures to reduce your mean time to detect and respond (MTTD and MTTR).



The integration enables the fetching and mirroring of Xpanse issues into Cortex XSOAR incidents as well as the ingestion of indicators (IPs, domains, and certificates), referring to the corporate network perimeter as discovered by Xpanse. Leveraging both technologies, your security team can respond to asset vulnerabilities and incidents with automated orchestration playbooks. You can trigger scans to enrich incidents and automatically generate tickets for on-premises and cloud assets.

## Discover

Scan the internet and accurately attribute unknown assets using multiple sources to reduce false positives and map your full attack surface.

## Enrich

Use automated playbooks to enrich incidents using Xpanse asset information and threat intelligence indicators, helping you reduce MTTD and MTTR across your cloud native, hybrid, and on-premises environments.

## Remediate

Improve your team's efficiency with a host of integrations and prebuilt scripts to automate attack surface management. For more details, see the Cortex Xpanse by Palo Alto Networks content pack.

### Network operations automation

In this use case, we will pivot from the SOC to the NOC. A flexible and scalable SOAR platform can be applied to any workflow or process, and our own Palo Alto Networks operations teams are using Cortex XSOAR internally to automate their manual processes.

One area where we have seen great benefits is network operations, where manual but necessary tasks are a time burden for the ITOps and NetOps teams.

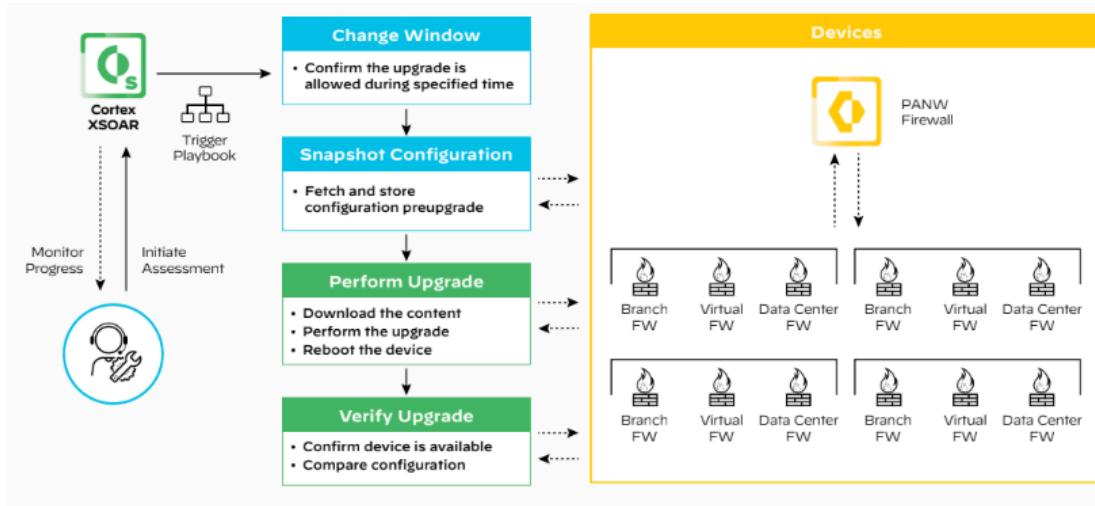
#### Manual Firewall Device Onboarding and Upgrades

It's a tedious and manual process to upgrade and validate all firewalls distributed across your network. There is significant time investment needed in the process where your team needs to download the firewall update, install, reboot, and verify that the upgrade was successful. For enterprises with over 100 firewalls distributed across their organization, this process is not scalable and is done infrequently.

"We manage about 450 firewalls. It takes us two hours to upgrade each firewall. We can only do a few at a time to ensure everything upgrades correctly." – Insurance industry customer

With Cortex XSOAR, you can onboard and upgrade all your devices within the environment and automatically verify upgrade status. There is still time required to download and reboot the system, but your NetOps team no longer has to "babysit" the process. Snapshots of the configuration can be captured to enable rollbacks if necessary. Once the upgrade is complete, verification steps can be performed to ensure the firewall is functioning properly.

There are many more automation use cases that can be deployed to streamline network operations, from policy and rule change management to monitoring network health and outages, but your NetOps teams will derive great efficiency benefits just from starting their automation journey with this key use case.



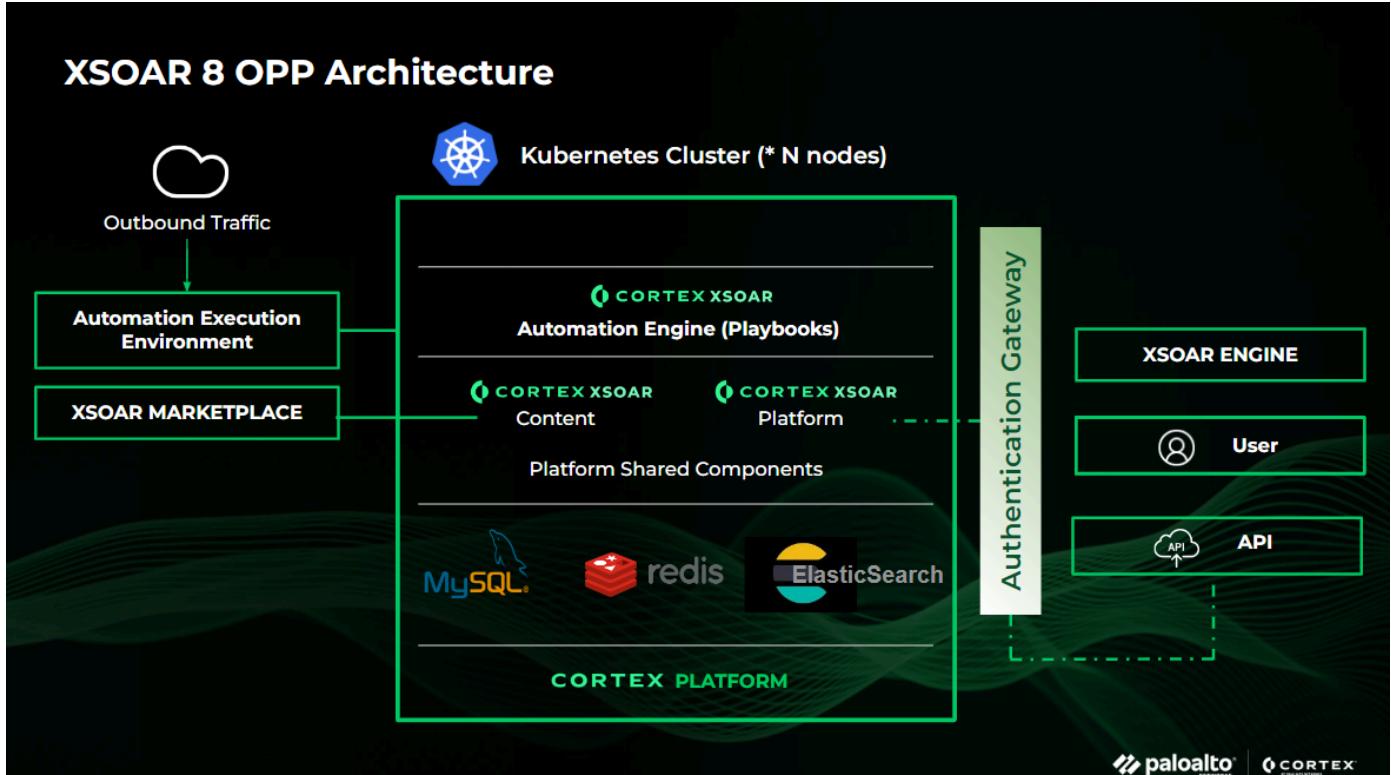
## 1.3 | Cortex XSOAR architecture

### Abstract

Describes the XSOAR On-prem architecture.

The following diagram describes the high-level architecture for Cortex XSOAR:

Cortex XSOAR installation is implemented by your IT team or Cortex XSOAR administrators. Cortex XSOAR uses the following:



- Rational store using MySQL
- Cache and synchronization using Redis
- Data warehousing using Elasticsearch

Cortex XSOAR is provided as an Kubernetes cluster, a set of nodes (VMs) that runs containerized applications that package Cortex XSOAR with its dependencies and some necessary services. You can decide how many nodes/VMs to include in the cluster when running the Administrative tool. You can decide between a standalone environment (one or two nodes) or a multi-nodes cluster (three or more nodes).

Playbooks are executed on dedicated and isolated workers and workloads do not share compute resources.

## 1.4 | Understand Cortex XSOAR licenses

### Abstract

The Cortex XSOAR license is downloaded from Cortex Gateway and determines which components users can use and how many users can access the tenant.

Cortex XSOAR requires a yearly license per user. Multi-year licenses are available.

### License usage

This table describes the types of Cortex XSOAR licenses which are used in the following circumstances:

Version	Usage	License
Cortex XSOAR (Enterprise) Edition	Built for customers who need a complete security automation solution.	Includes the SOAR Enterprise and TIM Enterprise licenses.
Cortex XSOAR Threat Intel Management Edition	Built for Threat Intelligence and Security Operations teams who need threat intelligence-based automation.	Includes the TIM Enterprise license only.
Cortex XSOAR Starter Edition	Built for Security Operations and Incident Response customers who need case management with collaboration and playbook-driven automation.	Includes the SOAR Enterprise license only.

## License quota

The following table describes the license quotas of each version in Cortex XSOAR.

	XSOAR TIM (TIM Only)	XSOAR Starter Edition (SOAR Only)	XSOAR (SOAR + TIM)
Integrations	Unlimited	Unlimited	Unlimited
Incident Management	30-day history	Unlimited	Unlimited
Incident Triggered Automations	166 daily	Unlimited	Unlimited
Job Triggered Automations	Unlimited	Unlimited	Unlimited
Intel Feeds	Unlimited	5 active feeds, 100 indicators/fetch	Unlimited
Threat Intel Library	Unlimited	Intelligence detail view and relationships data are not included	Unlimited
Unit 42 Intelligence	Unlimited UI access, 5k/day API points	Not included	Unlimited UI access, 5k/day API points

### NOTE:

Intel feed quotas are based on the selected Fetches Indicators field in the integration instance settings, not the enabled status. Disabling an integration instance does not affect the Intel feed quota. For example, if the AWS Feed is enabled and is fetching indicators and you don't want to include this in your quota, open the integration settings and clear the Fetches Indicators checkbox.

## Cortex XSOAR users

Cortex XSOAR has audit users and full users.

### Audit user

Audit users have read-only permission in Cortex XSOAR, meaning they do not have the ability to edit system components and data, or run commands, scripts, and playbooks. Audit users can view incidents, dashboards, and reports.

### Full user

Full users have read-write permission in Cortex XSOAR, meaning they have the ability to view and edit system components and data. They can investigate incidents, run scripts and playbooks, chat in the War Room, and more. Full users' access to Cortex XSOAR is determined by their assigned role.

## 1.5 | Roles and responsibilities

### Abstract

Learn about the typical core roles that make up a SOC team.

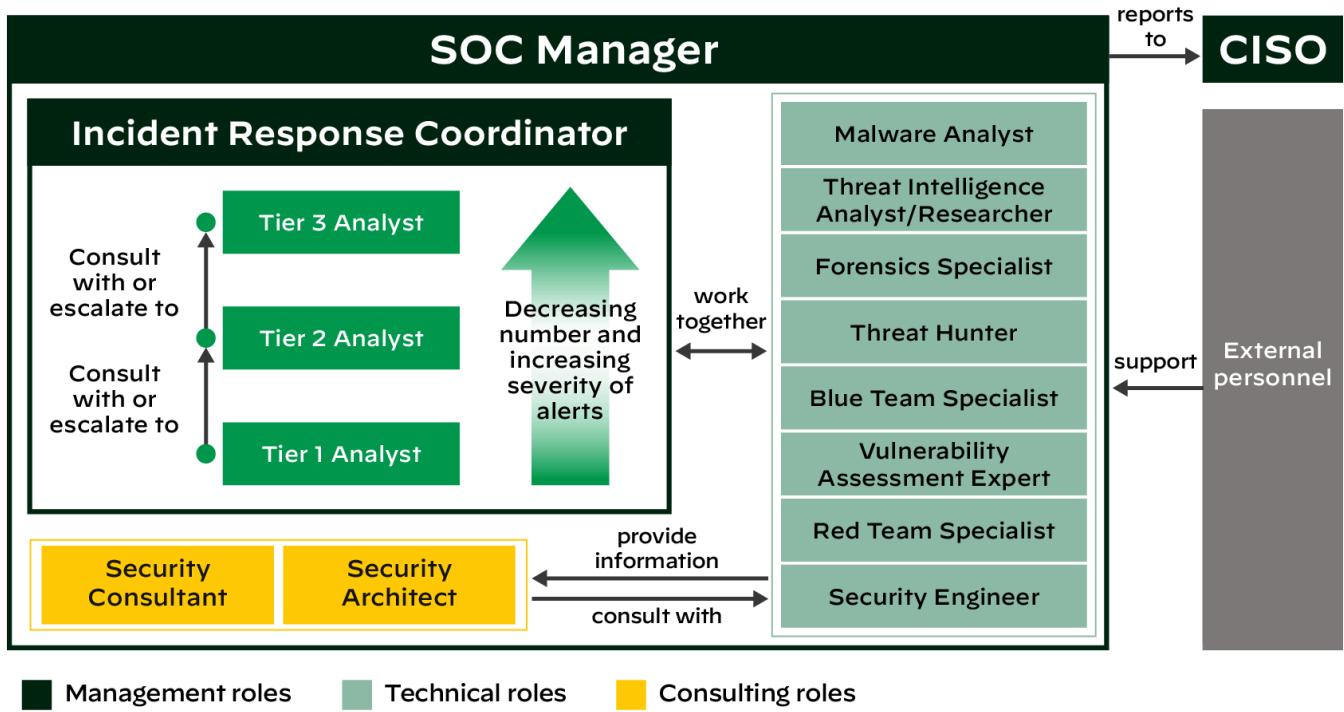
### What is the role of a Security Operations Center?

Security Operations Centers (SOCs) were created to facilitate collaboration among security personnel, with a primary focus on security monitoring and alerting, including the collection and analysis of data to identify suspicious activity and improve the organization's security. A SOC can streamline the security incident handling process as well as help analysts triage and resolve security incidents more efficiently and effectively. In today's digital world, a SOC can be located in-house, in the cloud (a virtual SOC), staffed internally, outsourced, for example, to an MSSP or MDR, or a mix of these. SOCs can provide continuous protection with uninterrupted monitoring and visibility into critical assets across the attack surface. They can provide a fast and effective response, decreasing the time elapsed between when the compromise first occurred and the mean time to detection.

## Roles and responsibilities

Typical core roles that make up a SOC team consist of different tiers of SOC analysts and dedicated managers:

- **Tier 1 - Triage specialist:** Mainly responsible for collecting raw data as well as reviewing incidents. They confirm, determine, or adjust the criticality of incidents and enrich them with relevant data. For every alert, the triage specialist has to identify whether it's justified or a false positive, as incident fatigue is a real issue. An additional responsibility at this level is identifying other high-risk events and potential incidents. All these need to be prioritized according to their criticality. If problems occurring cannot be solved at this level, they have to be escalated to tier 2 analysts. Triage specialists are often managing and configuring the monitoring tools.
- **Tier 2 - Incident responder:** Reviews the higher-priority security incidents escalated by triage specialists and does a more in-depth assessment using threat intelligence, such as indicators of compromise and updated rules. Incident responders need to understand the scope of an attack and be aware of the affected systems. The raw attack telemetry data collected at tier 1 is transformed into actionable threat intelligence at this second tier. Incident responders are responsible for designing and implementing strategies to contain and recover from an incident. If a tier 2 analyst faces major issues with identifying or mitigating an attack, additional tier 2 analysts are consulted, or the incident is escalated to tier 3.
- **Tier 3 - Threat hunter:** Most experienced workforce in a SOC. Threat hunters handle major incidents escalated to them by the incident responders. They also perform or at least supervise vulnerability assessments and penetration tests to identify possible attack vectors. Their most important responsibility is to proactively identify possible threats, security gaps, and vulnerabilities that might be unknown. They should also recommend ways to optimize the deployed security monitoring tools as they gain reasonable knowledge about a possible threat to the systems. Additionally, any critical security alerts, threat intelligence, and other security data provided by tier 1 and tier 2 analysts need to be reviewed at this tier.
- **SOC manager:** Supervises the security operations team. SOC managers provide technical guidance if needed, but most importantly, they are in charge of managing the team. This includes hiring, training, and evaluating team members; creating processes; assessing incident reports; and developing and implementing necessary crisis communication plans. They also oversee the financial aspects of a SOC, support security audits, and report to the chief information security officer (CISO) or a respective top-level management position.



## 1.6 | Supported web browsers

Cortex XSOAR supports the following web browsers:

Browser	Version
Chrome	95.x and later
Firefox	93.x and later

Browser	Version
Safari	13.x and later
Microsoft Edge	Latest version

## 2 | Onboard and configure Cortex XSOAR

### Abstract

Follow the steps to successfully onboard and configure Cortex XSOAR On-prem

Get up and running quickly. Our intuitive Onboard section guides you through essential setup steps like installation, remote repository configuration, and content management. Once you're set up, customize Cortex XSOAR to match your requirements.

### 2.1 | Plan your deployment

#### Abstract

Learn more about deployment considerations and onboarding steps for Cortex XSOAR.

Before you start your Cortex XSOAR deployment, consider the following:

- Do you need Cortex XSOAR to communicate with internal or external applications that may be blocked by a firewall or proxy?

You may need to create an engine to enable communication or for load balancing.

- Do you want to deploy a single node (standalone) or a cluster of three or more nodes?

When deciding how many nodes to deploy, consider the following:

- Currently, if you deploy a single node (standalone), you can't switch to a cluster of three or more nodes.

- If you deploy a cluster of three or more nodes, you can implement out-of-the-box high availability (HA) by replicating data between the nodes in the cluster. For more information, see High Availability for Cortex XSOAR.

- Do you need a repository for content development?

Add your private repository to Cortex XSOAR.

The remote repository enables developing and testing content in a development environment before using it in a production environment.

Production and development are separate Kubernetes clusters with no dependency between them. For example, you can deploy a three-node cluster for production and a standalone node for development. Or if you want to implement HA with three nodes for production and for development, you need a total of six nodes, three for production and three for development.

- How do you want users to access Cortex XSOAR? Do you need to set up SSO?

- Do you need to restrict user roles to certain actions?

- How do you want to communicate with users in Cortex XSOAR?

Which mail sender do you use? Do you want to integrate a communication app, such as Slack?

- What steps do you currently take in your day-to-day SOC operations? Which integrations will enable you to automate your most important and time consuming procedures?

### 2.2 | Onboarding checklist

#### Abstract

Activate, provision, grant access, and configure Cortex XSOAR.

We recommend that you review the following steps to successfully deploy and onboard Cortex XSOAR:

Step	Details	See More
Step 1: Install Cortex XSOAR	Install Cortex XSOAR by downloading the image file from the Cortex Gateway.	See topic
Step 2: Set up an engine	Use an engine for load balancing and proxies.	See topic
Step 3. Set up a remote repository	Set up a dev/prod environment with a private remote repository.	See topic
Step 4. Set up users & roles	Configure users, roles, and user groups, and set up authentication.	See topic
Step 5. Install and configure content	Install content packs and configure integrations for your use case.	See topic
Post-deployment	Configure user notifications and customize system emails. Configure system settings.	See topic

## 2.3 | Step 1. Install Cortex XSOAR

### Abstract

Learn how to install Cortex XSOAR On-prem, including system requirements, and adding a license.

### Before you begin

- Review the System Requirements for installation.
- Have a basic understanding of how to deploy OVA or VHD file formats.
- Add DNS records that point the following host names to the cluster IP address.
  - Cluster FQDN: The Cortex XSOAR DNS name for accessing the UI. For example, `xsoar.mycompany.com`.
  - API-FQDN: The Cortex XSOAR DNS name that is mapped to the API IP address. For example, `api-xsoar.mycompany.com`.
  - ext-FQDN: The Cortex XSOAR DNS name that is mapped to the external IP address. For example, `ext-xsoar.mycompany.com`.

### Install Cortex XSOAR

1. From the Cortex Gateway, in the Available for Activation section, use the serial number to locate the tenant to download.

2. Click Download On Prem.

3. If you want to use a production and a development tenant with a private remote repository, select Dev.

If you don't select it now, you can install a development tenant later.

4. Download one of the following image files.

- OVA: Supported by VMWare.
- VHD: Supported by Microsoft Hyper-V.

You can deploy a single node (standalone) or a cluster (three or more nodes).

5. Depending on the image file, do one of the following:

- Install Cortex XSOAR from an OVA image
- Install Cortex XSOAR from a VHD image

6. After installation, add the Cortex XSOAR license.

7. Optionally perform post-installation maintenance, including scaling up hardware resources and using your own X.509 certificate for a secure HTTP connection.

8. If you want to install a development machine, install the image files on the development virtual machine.

For more information, see Cortex XSOAR Installation.

## 2.4 | Step 2. Set up an engine

### Abstract

Set up a Cortex XSOAR engine on a remote machine.

Engines are installed on a remote machine and used mainly for the following:

- Integration instances for on-prem applications. For example, the GitLab v2 integration enables you to run commands on GitLab instances.
- Execute scripts and commands that require access to on-prem resources. For example, the Active Directory v2 integration enables you to run commands in Active Directory.
- Generic Indicator export service. In Cortex XSOAR, you can configure an EDL to share a list of Cortex XSOAR indicators with other products in your network, such as a firewall or SIEM. For example, your Palo Alto Networks firewall can add IP address and domain data from the EDL to block or allow lists.
- Load balancing which enables the distribution of the command execution load.

1. Review the engine requirements. For more information, see Engine requirements.

2. Install an engine. For more information, see Install an engine.

If you want to install an air-gapped engine, see Engine air gap installation.

To learn more about engines, requirements, and installation, see Engines.

## 2.5 | Step 3. Set up a remote repository

### Abstract

Set up a content management system with a development environment to create and test content before using it in a production environment.

When you set up a remote repository, you can add any private content repository that is Git-based, including GitHub, GitLab, and Bitbucket. Also, On-prem repositories are supported.

Although you can set up multiple development tenants, in a cluster of tenants that includes one production tenant and one or more development tenants, only one development tenant can push content. The production tenant and any other development tenants pull from the one development tenant that is configured to push content. After the remote repository is enabled in the production tenant, by default, the first development tenant that has been installed is set to push content to the remote repository. When you create additional development tenants, they are set to pull content from the remote repository.

If the content repository option is disabled for the production or development tenant, the tenant becomes standalone and does not push or pull content.

### Before you begin

- If you are changing your remote repository settings, back up existing content to your local computer by navigating to Settings & Info → Settings → System → Server Settings → Custom Content and click Export all custom content.
- You must have Instance Administrator or Account Admin permission.

### Install a development tenant

1. If you haven't done so already, download and install the Cortex XSOAR development tenant.

- a. From the Cortex Gateway, in the Available for Activation section, use the serial number to locate the development tenant to download.
- b. Click Re-Download On Prem.
- c. Select Dev and click Next.
- d. Select the relevant image file.
- e. Agree to the terms and conditions and select Download.
- f. Install Cortex XSOAR according to the image file downloaded.

2. After you have installed the development tenant, you can now set up the private remote repository. For more information, see Set up a private remote repository.

To learn more about remote repositories, requirements, and configuration, see Content management in Cortex XSOAR.

## 2.6 | Step 4. Set up users and roles

### Abstract

View the permissions, and predefined roles in Cortex XSOAR On-prem

Cortex XSOAR uses role-based access control (RBAC) to manage roles with specific permissions for controlling user access. RBAC helps manage access to Cortex XSOAR components, so that users, based on their roles, are granted the minimal access required to accomplish their tasks.

### Task 1. Create roles

Roles enable you to define permissions for specific components, such as incident data, playbooks, scripts, and jobs. For example, you can create a role that allows users to edit the properties of incidents, but not delete incidents. You can create new roles or customize out-of-the-box roles.

If you assign one or more roles to an incident, only users with those roles can view and interact with the incident. For example, you might have an incident with sensitive data that should only be accessible to Tier-1 analysts and managers.

Roles can also be used to define permissions for integration commands. On the Integration Permissions page, you can assign roles to specific integration instances (all commands for that instance) or specific integration instance commands. For example, you could assign the Generic Export Indicators Service integration instance the Account Admin role, or you could restrict certain commands in the Core Rest API to a specific role. For more information, see Integration Permissions.

1. Review out-of-the-box roles and role-based permissions.
2. Create a role.

For more information about out-of-the-box roles, permissions, and how to create roles, see Roles management.

### Task 2. User groups

While roles can be assigned directly to users, we recommend instead creating user groups. Each user group has a single role associated with it, but each user group can contain multiple users and user groups can be nested within each other, enabling you to further refine your RBAC requirements. Users can belong to multiple user groups.

For more information about user groups and how to create them, see User group management.

After adding users, assign users to user groups or assign users to direct roles.

### Authentication

You can create users locally or by using SAML Single Sign-On (SSO) in the tenant. After you create users, they authenticate by either:

- Using a username and password
- Using SSO

For more information about setting up authentication, see Set up authentication.

### Manage users

You can manage users including resetting passwords, sending invitations, and removing users.

By default, users do not have roles assigned and do not automatically have access to tenant data until you assign them a role or add them as members of a user group that has an assigned role.

For more information about how to manage users, see User management.

## 2.7 | Step 5. Install and configure content

### Abstract

What content includes in Cortex XSOAR.

### What is content?

In Cortex XSOAR, content includes the following:

Content	Description
Integrations	Third-party tools and services that the Cortex XSOAR platform works with to orchestrate and automate SOC operations. You can trigger events from these integrations that become incidents in Cortex XSOAR. After the incidents are created, you can run playbooks on these incidents to enrich them with information from other products in your system.
Playbooks	You can automate many security processes, including handling investigations and managing tickets and security responses that were previously handled manually. Playbooks enable you to organize and document security monitoring, orchestration, and response activities. When an incident is ingested, if a playbook runs, an incident is created.
Dashboards, reports, and widgets	Dashboards and reports consist of visualized data powered by fully customizable widgets, which enable you to analyze data from inside or outside Cortex XSOAR in different formats such as graphs, pie charts, or text. Reports allow you to share similar data outside of Cortex XSOAR via email. Reports can be scheduled to run at a specific time to capture data where the start/end time is important.
Classifiers and mappers	Classification determines the type of incident/indicator that is created for events ingested from a specific integration. You create a classifier and define that classifier in an integration. Mappers map the fields from your third-party integration to the fields that you defined in your incident/indicator layouts.
Incident types, fields, and layouts	All incidents that are ingested into Cortex XSOAR are assigned an incident type when they are classified. Each incident type has a unique set of data that is relevant to that specific incident type. Fields and layouts ensure that you see relevant information that is relevant to the incident type.
Indicator types, fields, and layouts	Indicators are categorized by indicator type, which determines the indicator layout and fields that are displayed and which scripts are run on indicators of that type.
Scripts	Perform a specific action, and are comprised of commands associated with an integration. Write scripts in either Python or JavaScript. Scripts are used as part of tasks, which are used in playbooks and commands in the War Room.

Content is organized into content packs to support specific security orchestration use cases, which are either preinstalled or downloaded from Marketplace. Content packs are created by Palo Alto Networks, technology partners, contributors, and customers.

After downloading and installing content packs, you can then start customizing the content to suit your use case. For example, although Cortex XSOAR comes with a Mail Sender integration already configured, you may want to set up your own Mail Sender integration, such as EWS.

For more information about installing and configuring content packs, see [Manage content packs](#).

### 2.7.1 | Install content packs

#### Abstract

#### Install a content pack

You can only install one content pack at a time. Cortex XSOAR automatically adds any content that is required to install the content pack. You can also add any optional content packs that use the content pack you want to install.

If you receive an error message when you try to install a content pack, you need to fix the error before installing. If a warning message is issued, you can still download the content pack, but you should fix the problem otherwise the content may not work correctly.

Before you install a content pack you should review the content pack to see what it includes and what are the various dependencies. Following is the information you can view:

- **Details:** General information about the content pack such as installation, content, version, author, and status.
- **Content:** The content to be installed, such as scripts or integrations.
- **Dependencies:** Details of any required content packs and optional content packs that may need to be installed with your content pack.
- **Version History:** View the currently installed version, earlier versions, available updates, and revert if required.

How to install a content pack in Marketplace

1. Go to Marketplace → Browse and locate the content pack you want to install.
2. Click the required content pack and review the contents.
3. Click Install to add the content pack to the Cart.
4. (Optional) If the content pack includes optional content, select the content packs you want to add.

The Cart displays the number of items you are installing including any required content packs. You can log in and out, but the content packs remain in the Cart until you click either Empty cart or Install.

5. Click Install.
6. After installation, click Refresh content.

You can now start configuring your content. If you have installed an integration, configure the integration including setting up an integration instance. For more information, see Configure integrations.

## 2.7.2 | Set up your use case with the Deployment Wizard

### Abstract

The Deployment Wizard guides you step-by-step to quickly adopt your use case.

The Deployment Wizard can be used to set up your use case for the **Malware Investigation and Response** content pack and the **Phishing** content pack. In order to work with your content pack you need to set up your integrations. The Deployment Wizard guides you through:

- Configuring the integrations that will be used to fetch events (fetching integrations). These events will be mapped as incidents.
- Configuring the main playbook and its input parameters. For example, the Setup Malware playbook pane opens showing the recommended primary playbook for the incident type you selected when configuring the fetching integration. The playbook configuration includes all the input parameters to configure that will change the playbook behavior, for example, whether to use sandbox detonation or whether to perform isolation response. You can open the playbook by clicking the link on the bottom.
- Configuring any supporting integrations. such as an email integration

The default fetching integration for your content pack depends on which fetching integration(s) are installed. For example:

Content Pack	Default Fetching Integration In Order Of Priority
Malware Investigation and Response	1. Palo Alto Networks Cortex XDR - Investigation and Response 2. CrowdStrike Falcon 3. Microsoft Defender for Endpoint
Phishing	1. Gmail 2. EWS v2 (Make sure you also install the Microsoft Exchange On-Premise pack) 3. O365 Outlook Mail (Using Graph API) 4. Gmail Single User 5. O365 Outlook Mail Single User (Using Graph API)

### Prerequisites

To access the Deployment Wizard for the first time, you need to first install or update your Malware Investigation and Response content pack or your Phishing content pack in Marketplace. The Deployment Wizard tab appears in Marketplace after the content pack installation or update is completed.

### For example:

- For the Malware Investigation and Response content pack, you need at least one incident fetching content pack (mandatory). You can also optionally install sandbox, messaging, case management, and data enrichment and threat intelligence content packs.
- For the Phishing content pack, you need at least one email gateway content pack (mandatory). You can also optionally install sandbox, EDR systems, network devices, email security gateways, mail sender, and data enrichment and threat intelligence content packs.

#### How to set up your use case with the Deployment Wizard

1. In Marketplace, select the content pack for your use case (for example, Malware Investigation and Response or Phishing) and click Install or Update (if the pack is already installed).
2. In the Select Content Packs window, select one or more content packs from the required categories. You can also install other supportive content packs from other categories if needed. These items will be automatically added to the cart.
3. Click Continue and then Install or Update.
4. When the content pack finishes installing or updating, click Refresh content.

The Deployment Wizard tab appears.

##### NOTE:

After you start running your use case you can return to this tab and make changes to the configurations, such as your integration's credentials or playbook parameters.

5. Click Let's Start in the small dialog box that appears next to the Deployment Wizard tab.

The tab opens showing the use case deployment flow.

6. Step 1: Fetching Integration - Click the displayed fetching integration. If the integration is new, select New instance. If you want to use an existing instance, select it from Update existing instance. The integration will stay disabled until you complete all steps of the wizard.

##### NOTE:

You must define the incident type in order to set the playbook in the next step.

A list of What needs to be done guides you through the required fetching integration instance settings configurations. Scroll down to see the complete list.

After you save your settings, the wizard initiates a test connection. If the connection succeeds, the Fetching Integration step turns green and moves to the next step (Set Playbook).

7. Step 2: Set Playbook - Select Configure Playbook & Parameters.

##### NOTE:

The wizard displays the recommended playbook. If for the fetching integration setup you chose an incident type that uses a different playbook from the recommended one, the incident type will be detached.

8. Click Done.

9. Step 3: Supporting Integrations - Configure any installed supporting integrations in the content pack.

If a supporting integration is already installed and connected, it appears with a green check. Otherwise, click the integration to configure it.

##### NOTE:

After you save the settings, the integration instance is automatically enabled.

10. Step 4: What's Next - Select Turn on Use Case to start the fetching process and running the playbooks and scripts.

## 2.8 | Post deployment

After you complete the onboarding steps, there are additional optional configuration options.

- **User communication:** You can choose a mail sender and customize system emails. Users can set which notifications they want to receive and through which channels, such as email or Slack.
- **System settings:** Cortex XSOAR offers a wide variety of system settings. For example, you can enhance security, set a custom logo and login message, and choose a timezone.

### 2.8.1 | User communication

In Cortex XSOAR, you can configure mail and messaging integrations to send notifications to users and you can customize the subject and body of system emails.

Users can choose which notifications to receive and whether to receive notifications via email, Slack, Microsoft Teams, or other communication tools. For more information, see User details and preferences.

#### 2.8.1.1 | Configure notifications in Cortex XSOAR

Cortex XSOAR can send out notifications and emails to users through the following:

- By email using a mail sender
- By a message notification such as Slack.

#### Mail Sender integrations

A mail integration enables Cortex XSOAR to send emails and can be used for system notifications and playbooks. For example, when adding users to Cortex XSOAR, an email invitation is sent to users to log in. When you use the mail integration for playbook tasks, you can pass arguments such as to, subject, body, etc. to customize the contents of your email.

1. Go to Marketplace.
2. Search for and download a mail sender content pack (such as Microsoft Exchange On-Premise).
3. Go to Settings & Info → Integrations → Instances.
4. Locate the mail sender integration (for example, EWS v2) and click Add Instance.
5. Configure your mail sender integration and select Enable to enable your mail sender integration.
6. If you configure multiple email integrations, select the Do not use in CLI by default option in the integration instances that should not be used to send emails. This ensures that the email will only be sent in the instance you are expecting when running the send-mail command from the CLI or within a playbook.

#### Multiple sender integrations

When there are multiple instances of a mail sender in Cortex XSOAR, you can choose which email sender should send the notification by configuring the `server.notification.using.sendmail` key in the advanced server configuration settings.

If you do not configure the advanced server setting, Cortex XSOAR uses the first email integration it finds to send the system notifications.

1. Navigate to Settings & Info → Settings → Server Settings → Server Configuration → Add Server Configuration.
2. Add the following key and enter the mail sender instance name:

Key	Value
<code>server.notification.using.send-mail</code>	The mail sender instance name.

#### Configure a messaging integration

If your organization uses a messaging service, such as Slack or Microsoft Teams, we recommend installing the relevant content pack.

The Slack content pack enables you to send messages and notifications to your Slack team and integrates with Slack's services to execute create, read, update, and delete operations for employee lifecycle processes. For more information, see Slack content pack. For more information about Microsoft Teams, see Microsoft Teams content pack.

#### 2.8.1.2 | Customize system emails

##### Abstract

Customize subject and message body for Cortex XSOAR system emails and choose HTML and/or text format.

Cortex XSOAR sends notifications to users. You can customize the subject and the contents of the email, and choose whether to send the email in HTML format. The following are the default message subjects and the default message contents:

Message Type	Default Subject	Default Message Body
mentionNew	Message from Cortex XSOAR Security Operations Server	<code>{{.username}} added you to investigation {{.invName}}.\nYou were mentioned: {{.parentContent}}.</code>
mentionNewNoContent	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} added you to investigation {{.invName}}.</code>

Message Type	Default Subject	Default Message Body
mentionOld	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} mentioned you in investigation {{ .invName}}: {{ .parentContent}}.</code>
assign	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} assigned task #{{ .taskId}} in investigation {{ .invName}} to you.</code>  <b>NOTE:</b> The assign message type is only relevant for Playbook tasks.
todoAssign	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} assigned To-Do task {{ .title}} in investigation {{ .invName}} to you.</code>
taskCompleted	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} completed task #{{ .taskId}} in investigation {{ .invName }}.</code>
taskUpdated	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} updated task #{{ .taskId}} in investigation {{ .invName }}.</code>
investigationClosed	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} has closed investigation {{ .invName }}.</code>
investigationWaiting	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}}, {{ .invName }} has stopped and is waiting your instructions."</code>
investigationError	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}}, {{ .invName }} has stopped because of an error.</code>
investigationDeleted	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} has deleted investigation {{ .invName }}.</code>
incidentOpened	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} has reported {{ .incTermArticle}} {{ .incTermSingular}} {{ .invName }}.</code>
incidentChanged	Message from Cortex XSOAR Security Operations Server	<code>{{ .username}} has updated {{ .incTermArticle}} {{ .incTermSingular}} {{ .invName }}.</code>
incidentStatusChanged	Playbook has stopped on {{ .runStatus}} for {{ .invName}} (#{{ .incID}})	<code> {{ .incTermCapitalSingular}} playbook task " {{ .taskId}}" stopped on {{ .runStatus}}. {{ .incTermCapitalSingular}} Id: #{{ .incID}} {{ .incTermCapitalSingular}} Name: {{ .invName}}{{ .incTermCapitalSingular}} SLA: {{ .SLA}}{{ .incTermCapitalSingular}} Severity: {{ .severity}}Task: #{{ .taskID}}Task Name: {{ .taskName}}Task SLA: {{ .TaskSLA}}</code>

Message Type	Default Subject	Default Message Body
incidentAssigned	Message from Cortex XSOAR Security Operations Server	{{.username}} has assigned you {{.incTermArticle}} {{.incTermSingular}} {{.invName}}.
taskCompletedWithNotes	Message from Cortex XSOAR Security Operations Server	{{.username}} completed task #{{.taskId}} in investigation {{.invName}}.\nCompletion note was: {{.taskComment}}
incidentReminderSLA	Message from Cortex XSOAR Security Operations Server	FYI, {{.incTermSingular}} #{{.invID}}" {{.remindedOn}}" - SLA expiration is approaching. ({{{.SLA}}})
MessageTypeTaskSLA	Message from Cortex XSOAR Security Operations Server	FYI, task "{{.remindedOn}}" (from investigation {{.invName}}) - due date is approaching. ({{{.SLA}}})
newContentAvailable	Message from Cortex XSOAR Security Operations Server	A content update: {{.release}} for your Demisto Server is available.\n{{.releaseNotes}}
failedFetchIncidents	Integration instance {{ .instance}} ({{{.brand}}}) failed fetching new {{ .incTermPlural}}	Integration instance {{.instance}} ({{{.brand}}}) failed fetching new {{.incTermPlural}} at {{{.date}}}.\nerror message is:\n{{.error}}
engineDisconnected	Cortex XSOAR Engine Disconnected	Engine '{{.name}}' ({{{.host}}}) is disconnected. Engines will not process integration automations until it is reconnected.
externalFormSubmit	{{{.subject}}}	""
externalAskSubmit	{{{.subject}}}	""
jobRunning	Message from Cortex XSOAR Security Operations Server	A previous instance of job {{.invName}} is already running.

Change the email subject

You can customize the subjects of system emails.

1. Go to Settings & Info → Settings → System → Server Settings → Server Configuration.
2. Add the key **messages.subject.formats.<MessageType>**, where **<MessageType>** is the type of message, such as **assign** or **taskCompleted**. For the value, enter your custom subject. You can use any of the default variables, for example **.invName** in your subject.

Examples:

Key	Value
<b>messages.subject.formats.assign</b>	You were assigned to an incident

Key	Value
messages.subject.formats.taskcompleted	Task completed in {{.invName}}

Change the email body

You can customize the content of the system messages, and include variables such as .username and .invName in your body content.

You can send HTML or non HTML messages. If you have users who can only receive plain text, use the key `messages.formats.<MessageType>`, where `<MessageType>` is the type of message, such as `assign` or `taskCompleted`. Enter your custom body text as the value. If you have users who can receive HTML emails, use the key `messages.HTML.formats.<MessageType>`, where `<MessageType>` is the type of message. Enter your custom body text as the value. To set custom body text for both text and HTML messages, add both keys/values for each message you want to customize.

1. Go to Settings & Info → Settings → System → Server Settings → Server Configuration.
2. Add the key `messages.formats.<MessageType>` or `messages.HTML.formats.<MessageType>`. For the value, enter your custom email body.

Examples:

Key	Value
<code>messages.HTML.formats.assign</code>	<code>{{.username}}</code> added you to investigation {{.invName}}.\nPlease log in and review.
<code>messages.formats.assign</code>	<code>{{.username}}</code> added you to investigation {{.invName}}.\nPlease log in and review.

## 2.8.2 | Configure system settings

You can configure security settings, such as session expiration, approved domains and IP ranges, and the option to disable inactive users, in the Security Settings page.

You can configure server settings, such as a keyboard shortcut for fast navigation, the timezone and the timestamp format, logo, login message, and specific server configurations, from the Server Settings page. You can also import and export custom content.

### 2.8.2.1 | Configure server settings

Abstract

Define keyboard shortcuts, timezone, logo, server configurations, and more.

You can create a more personalized user experience by defining your server settings. Go to Settings & Info → Settings → Server Settings.

#### NOTE:

By default, the keyboard shortcuts, timezone, and timestamp format options appear on the Preferences table of the User Details page. To instead display these settings in the Server Settings page, add the `UI.show.timezone.in.server.settings` server config, set to true. Keyboard shortcuts, timezone, and timestamp format are not set universally and only apply to the user who sets them.

Server Setting	Description
Keyboard Shortcuts	<p>Use a shortcut to search, investigate, and initiate actions. To change the shortcut letter, click the letter in the box, type a letter, and then save.</p> <p><b>NOTE:</b></p> <p>The shortcut value must be a keyboard letter (A to Z).</p>
Timezone	Select the timezone to display your Cortex XSOAR data, which affects the timestamps displayed in Cortex XSOAR, such as auditing logs, and exported files.

Server Setting	Description
Timestamp Format	The timestamp format is displayed in data tables, auditing logs, and exported files. The setting is configured per user and not per tenant.
Appearance	<p>By default, the full-size Cortex XSOAR logo displays on the sign-in page, on the navigation bar when expanded, on reports, on the artifact viewer, and on communication task forms and emails. A minimized version of the default Cortex XSOAR logo displays at the top of the navigation bar when it is collapsed. You can replace the default logos with a custom logo to match your organization's branding in the Cortex XSOAR platform. Supported file formats are PNG, JPEG, SVG, and GIF. You can add the following:</p> <ul style="list-style-type: none"> <li>• Full-size logo: Upload your logo (displayed when the navigation bar is not collapsed).</li> <li>• Minimized logo: Upload your logo for the top of the navigation bar when it is collapsed (minimized).</li> </ul> <p><b>NOTE:</b></p> <p>If you define a full-size logo, but not a minimized logo, no logo will display when the navigation bar is collapsed.</p>
Telemetry Collection	<p>Cortex XSOAR uses telemetry to collect specific usage data, which is analyzed and used to improve Cortex XSOAR and to identify common usage to help drive the product roadmap.</p> <p>You can select the following:</p> <ul style="list-style-type: none"> <li>• All: Includes data that helps improve operational efficiency, optimizes resource allocation, enhances the overall user experience of Cortex XSOAR, and data relevant for debugging. For more information, see Telemetry in Cortex XSOAR.</li> <li>• System diagnostics only: Captures data relevant to debug issues only. Cortex XSOAR sends error logs stack traces and infrastructure metrics that could help debug technical issues, CPU, memory, etc.</li> <li>• None: No telemetry is transmitted, apart from essential information according to your license, such as usage.</li> </ul> <p><b>NOTE:</b></p> <p>Only users with Role → Components → Administration → View/Edit permission can change the telemetry scope, such as Administrators.</p>
Custom Content	<p>You can do the following:</p> <ul style="list-style-type: none"> <li>• Export all custom content: Exports custom content, such as playbooks and scripts as a content bundle, which you can import for use in another Cortex XSOAR tenant.</li> <li>• Upload custom content: Imports custom content created from a Cortex XSOAR tenant.</li> </ul>
Login Message	<p>You can display a custom message to users before every login to Cortex XSOAR. For example, you can add a message that includes terms and conditions specific to your organization to help adhere to the National Institute of Standards and Technology (NIST) security standards and reduce cybersecurity risk. The message is by default disabled.</p> <p><b>NOTE:</b></p> <p>You must have administration rights to access this feature. The message supports markdown.</p>
Server Configuration	<p>Add to customize your Cortex XSOAR environment on the tenant level. You can also use custom server configurations where you experience issues or need to troubleshoot situations in your environment. For a list of server configurations, see Server configurations.</p>

#### 2.8.2.2 | Configure security settings

##### Abstract

Configure security settings such as session expiration, user login expiration, and dashboard expiration.

You can configure security settings such as how long users can be logged into Cortex XSOAR, and from which domains and IP ranges users can log in.

Go to Settings & Info → Settings → System → Security Settings.

Settings	Options	Description
Session Expiration	User Login Expiration	The number of hours (between 1 and 24) after which the user login session expires. You can also choose to automatically log users out after a specified period of inactivity.
	Dashboard Expiration	Whether the Dashboard page expires at the same time as the user login session or after seven days. This is useful when you view a dashboard on a separate screen.  For example, if you select seven days for dashboards and eight hours for login expiration and you are currently viewing the Dashboard page, the dashboard expiration takes priority (seven days). This ensures that the Dashboard page continues to display the widgets for an extended period.
Allowed Sessions	Approved Domains	The domains from which you want to allow user access (login) to Cortex XSOAR. You can add or remove domains as necessary.
	Approved IP Ranges	The IP ranges from which you want to allow user access (login) to Cortex XSOAR. You can also choose to limit API access from specific IP addresses.
User Expiration	Deactivate Inactive User	Deactivate an inactive user, and also set the user deactivation trigger period. By default, user expiration is disabled. When enabled, enter the number of days after which inactive users should be deactivated.
Allowed Domains	Domain Name	Enables you to specify one or more domain names that can be used in your distribution list for audit forwarding.

## 2.9 | Configure Cortex XSOAR

### Abstract

Configure engines, playbooks, scripts, dashboards, etc., for your use case.

As soon as you have completed onboarding with Cortex XSOAR, you can start configuring the tenant to match your use cases.

Section	Details	See More
Engines	If you have not done so already, you can configure and manage engines, such as using an engine as a web proxy and setting up Docker hardening.	Engines
Marketplace	You may want to install additional content packs, delete, update, revert, and set up notifications.	Marketplace

Section	Details	See More
Integrations	Configure integrations, including fetching incidents, managing credentials, troubleshooting, and more.	Integrations
Incidents	Customize incident fields, layouts, and types, set up preprocessing and post-processing rules, limit access to an investigation, etc.	Incident configuration
Playbooks	Learn how to customize your playbooks including creating tasks, sub-playbooks, and polling.	Playbooks
Lists	Create lists and add them to playbooks or scripts.	Lists
Jobs	Run playbooks based on certain events or on a specific time and date.	Jobs
SLAs	Incorporate SLA fields in your investigations so you can view how much time is left before the SLA becomes past due, as well as configure actions to take when the SLA is passed its due date.	SLAs
Indicators	Customize indicator fields, layouts, and types, classify and map fields, and delete and exclude indicators.	Indicator configuration
Dashboards, reports, and widgets	Customize and create widgets to add to your dashboard and reports.	Dashboards and Reports

After you have configured Cortex XSOAR, analysts can start to investigate incidents and indicators.

## 3 | Cortex XSOAR Installation

### Abstract

Install Cortex XSOAR On-prem and complete post-installation steps. Learn how to upgrade Cortex XSOAR.

Download one of the image files from the Cortex Gateway, install, and complete the post-installation steps. Learn how to upgrade Cortex XSOAR.

### 3.1 | Installation

#### Abstract

Download the image files from the Cortex Gateway and install Cortex XSOAR on your machine.

Install Cortex XSOAR by downloading and deploying an image to one or more virtual machines and then setting up and installing Cortex XSOAR on the virtual machines.

#### Deploy an image

Cortex XSOAR supports the following image files, which you download from the Cortex Gateway:

- OVA: Supported by VMWare.

For more information about installing Cortex XSOAR by deploying an OVA image file, see [Install Cortex XSOAR from an OVA image](#).

- VHD: Supported by Microsoft Hyper-V.

For more information about installing Cortex XSOAR by deploying a VHD image file, see [Install Cortex XSOAR from a VHD image](#)

You can deploy the image on your platform as a single node (standalone) or a cluster of three or more nodes, where each node is a dedicated virtual machine (VM).

After you deploy and log into your virtual machine, a textual UI menu is displayed enabling you to configure IP settings, proxy, and trust between all nodes in a cluster and then to install Cortex XSOAR.

## Set up and install standalone

Installing on a standalone node involves:

- Setting up one VM.
- From the textual UI Cluster Installation menu, install the cluster from one node, including setting the IP address of the node.

## Set up and install a cluster

Installing on a cluster involves:

- Set up multiple VMs.
- From the textual UI:
  - In the Connect Nodes menu, connect the VMs (establishing trust between all nodes in the cluster).
  - In the Cluster Installation menu, select one node to install the cluster from, including setting the IP addresses of each node.

### **NOTE:**

When deciding how many nodes to deploy, consider the following:

- Currently, if you deploy a single node (standalone), you can't switch to a cluster of three or more nodes.
- If you deploy a cluster of three or more nodes the system automatically implements built-in High Availability (HA), enabling workload distribution and data replication across the nodes. This ensures a continuous operation. For more information, see [High Availability for Cortex XSOAR](#).

## 3.2 | System Requirements

### Abstract

Verify that your Cortex XSOAR deployment meets the minimum system requirements.

Cortex XSOAR requires the following hardware, URLs, and bandwidth. Verify you meet all minimum system requirements.

### Hardware requirements

The Cortex XSOAR tenant has specific minimum VM hardware requirements depending on the scale.

### **NOTE:**

The performance benchmark indicates how many playbook runs per hour can be supported.

The benchmark values provided are for a single node running the Phishing playbook. This value can vary according to playbook size and complexity.

### **IMPORTANT:**

A hypervisor host running VMs must have enough hardware resources to support all Cortex XSOAR VMs that are planned to run on it.

Each VM (node) in a cluster needs to have the same resources. For example, a 3 VM cluster planned to run on a host must have at least 3 times the listed specs for memory, CPU cores, and storage. Leave additional space for overhead virtualization operations.

To fully leverage High Availability, deploy each VM on a different hypervisor. This ensures that if one hypervisor fails, the other VMs continue to operate.

The following requirements apply for a single node (standalone), or for each node in a cluster.

Component	Small Scale	Medium Scale	Large Scale
CPU per VM	16 CPU cores	32 CPU cores	48 CPU cores
Memory per VM	64 GB RAM	128 GB RAM	192 GB RAM
Storage per VM	256 GB boot disk plus an additional separate 775 GB data disk. These disks must be SSD.	256 GB boot disk plus an additional separate 1.3 TB* data disk. These disks must be SSD.	256 GB boot disk plus an additional separate 1.8 TB* data disk. These disks must be SSD.

Component	Small Scale	Medium Scale	Large Scale
Performance benchmark running the Phishing playbook	Up to 200 playbook runs per hour	200 - 350 playbook runs per hour	Up to 500 playbook runs per hour

\*1 TB = 1024 GB

#### Required ports for cluster communication

##### SSH

Relevant for Standalone (one VM) and three VMs (3-node cluster).

Port	Protocol
22	TCP

##### Control plane

Relevant for three VMs (3-node cluster).

Name	Port	Protocol
etcd client port	2379	TCP
etcd peer port	2380	TCP
Kubernetes API	6443	TCP
Kubelet API	10250	TCP
kube-scheduler	10257	TCP
kube-controller-manager	10259	TCP

##### Worker node

Relevant for three VMs (3-node cluster).

Name	Port	Protocol
kube nodeport range	30000:32767	TCP

##### Intra-node communication

#### IMPORTANT:

To prevent degraded storage performance, make sure all nodes are synchronized with no NTP offset.

Relevant for three VMs (3-node cluster).

Name	Port	Protocol
Calico with IPv4 Wireguard	51820	UDP

**Required URLs**

You need to allow the following URLs for Cortex XSOAR to operate properly.

**NOTE:**

If you use SSL inspection and experience difficulty connecting to the required URLs or to integration URLs, exclude the required URLs from SSL offloading on the firewall/proxy.

Function	Service	Port	Direction
Web interface	HTTPS	443	Inbound
Engine connectivity	HTTPS	443 (configurable)	Inbound
Integrations	Integration-specific ports		Outbound
Unit42 Intel Inventory (TIM license)	<a href="https://unit42intel.xsoar.paloaltonetworks.com">https://unit42intel.xsoar.paloaltonetworks.com</a>	443	Outbound
Marketplace	<ul style="list-style-type: none"> <li>• <a href="https://marketplace.xsoar.paloaltonetworks.com/">https://marketplace.xsoar.paloaltonetworks.com/</a> Download content packs and view the Marketplace (to view content pack images, the domain should also be reachable from the browser).</li> <li>• <a href="https://storage.googleapis.com/marketplace-dist/">storage.googleapis.com</a> Download content packs and view the Marketplace. This domain stores content pack artifacts (to view content pack images, the domain should also be reachable from the browser). It is possible to further limit the url prefix to: <a href="https://storage.googleapis.com/marketplace-dist/">https://storage.googleapis.com/marketplace-dist/</a></li> <li>• <a href="https://api.demisto.com">api.demisto.com</a> Download content Packs and view the Marketplace (this file maps the Marketplace URL to the Cortex XSOAR version).</li> </ul> <p><b>NOTE:</b></p> <p>You must add <a href="https://marketplace.xsoar.paloaltonetworks.com">marketplace.xsoar.paloaltonetworks.com</a>, <a href="https://storage.googleapis.com">storage.googleapis.com</a>, and <a href="https://api.demisto.com">api.demisto.com</a> otherwise you cannot access the Marketplace.</p> <ul style="list-style-type: none"> <li>• <a href="https://xsoar-authentication-proxy.paloaltonetworks.com">xsoar-authentication-proxy.paloaltonetworks.com</a> Login and register users.</li> <li>• <a href="https://xsoar-contrib.pan.dev">xsoar-contrib.pan.dev</a> Contribute content packs.</li> </ul>	443	Outbound
On-prem Gateway	<a href="https://onpremgw.crtx.[region].paloaltonetworks.com">onpremgw.crtx.[region].paloaltonetworks.com</a>	443	Outbound

Function	Service	Port	Direction
Download packages required for installation	<ul style="list-style-type: none"> <li>• deb.debian.org</li> <li>• security.debian.org</li> <li>• debian.map.fastlydns.net</li> </ul>	80	Outbound

#### Bandwidth requirements

The minimum required download bandwidth is 10Mbit/s. This is required for successful Cortex XSOAR upgrades and Marketplace operations.

### 3.3 | High Availability for Cortex XSOAR

If you deploy a cluster of three or more nodes, the system automatically implements built-in High Availability (HA). A cluster of three or more nodes enables distributing the workload and replicating data across the nodes to ensure continuous operation. If one node fails, the remaining nodes take over the tasks and data handling, minimizing downtime and maintaining service availability. This setup also allows for load balancing and redundancy, enhancing the overall resilience of the system.

#### IMPORTANT:

For built-in High Availability to work:

- To fully leverage High Availability, deploy each VM on a different hypervisor. This ensures that if one hypervisor fails, the other VMs continue to operate.
- The IPs of all VMs (nodes) in a cluster as well as the virtual IP must be in the same subnet.

### 3.4 | Install Cortex XSOAR from an OVA image

#### Abstract

Download an OVA image from Cortex Gateway, deploy the image, and use the textual user interface to configure network, IP, and environment settings, and to install a Cortex XSOAR tenant.

#### PREREQUISITE:

- To be able to download the Cortex XSOAR 8 images from Cortex Gateway, you need a license (or evaluation license via sales) assigned to your CSP account.
- Review the System requirements for deploying a Cortex XSOAR tenant.
- Have a basic understanding of how to deploy OVA file formats.
- Add DNS records that point the following host names to the cluster IP address.
  - Cluster FQDN - The Cortex XSOAR DNS name for accessing the UI. For example, xsoar.mycompany.com.
  - API-FQDN - The Cortex XSOAR DNS name that is mapped to the API IP address. For example, api-xsoar.mycompany.com.
  - ext-FQDN - The Cortex XSOAR DNS name that is mapped to external IP address. For example, ext-xsoar.mycompany.com.

#### Task 1. Select and download the OVA image and license from Cortex Gateway

#### TIP:

In Google Chrome, to download the image and license files together, you may need to set the browser Settings → Privacy and security → Site settings → Additional permissions → Automatic downloads to the default behavior Sites can ask to automatically download multiple files.

1. Log in to Cortex Gateway. For your Cortex XSOAR license, select Download On Prem.

By default, the Production-Standalone license is selected. You can also select Dev.

Production and development are separate Kubernetes clusters with no dependency between them. For example, you can deploy a three-node cluster for production and a standalone node for development. Or you can support small scale for development and large scale for production.

2. Click Next.

3. Select the OVA image format to download.

OVA is supported by VMWare.

4. Select the checkbox to agree to the terms and conditions of the license and click Download.

Two files download: A zipped license file containing one or more JSON license files with instructions, and a zipped image file of the type you selected (.ova, .vhd)

5. Extract (unzip) the license and image files.

## Task 2. Deploy your virtual machine

The following is an example of deploying your VM on VSphere from an OVA image. For more details, see Deploying OVF Templates.

If you set your Cortex XSOAR environment as a standalone (single node), you cannot add nodes to it and switch to a cluster. If you deploy three nodes, you can later add nodes and expand the cluster. For more information, see Add or remove nodes in a cluster.

1. Copy the downloaded image file into your hypervisor.

2. Wherever the templates are located, right click one of the templates and select Deploy OVF Template.

### NOTE:

Although you can create a virtual machine directly from the OVA image file, deploying an OVF template enables creating multiple configured virtual machines from one downloaded OVA instead of downloading the same OVA for each virtual machine, which can be time consuming.

3. Right click the template file and select New Virtual Machine.

4. Follow the wizard instructions to define the virtual machine properties, including:

1. Select the storage for the virtual machine configuration and disk files.

- Batch configure or configure per disk.
- Set the virtual disk format.
- Set the VM storage policy.
- Disable storage DRS.

2. Select Customize this virtual machine's hardware from the clone options and go to the Customize hardware step. This includes CPU, memory, hard disk space, network adapter, and other settings.

### IMPORTANT:

Every virtual machine is provided with a 256 GB hard disk to run the OS. However, you also need to add an extra hard disk for each virtual machine instance you want to deploy to run the application.

All virtual machines in a cluster must have the same storage size.

To ensure successful deployment, make sure the hard disks meet performance requirements detailed in the System requirements.

1. Select ADD NEW DEVICE → Hard Disk.

2. Set the disk space for the extra hard disk to 775 GB.

3. Choose the Thin Provision hard disk type (for SSD).

4. If the virtual machine is running, reset it.

3. Click FINISH.

4. Go to the folder the virtual machine was deployed to and select the virtual machine name you defined.

5. In the Summary tab Guest OS section, click the console to launch the new virtual machine.

5. Repeat from Step 3 for each additional virtual machine in the cluster.

6. Log in to each virtual machine console. When logging in for the first time, enter the default user name admin and password admin and then create a new password.

### NOTE:

The password must be at least eight characters long and contain at least:

- One lower case letter
- One upper case letter
- One number, or one of the following special characters: !@#%

The textual UI menu appears with all the configuration and installation options.

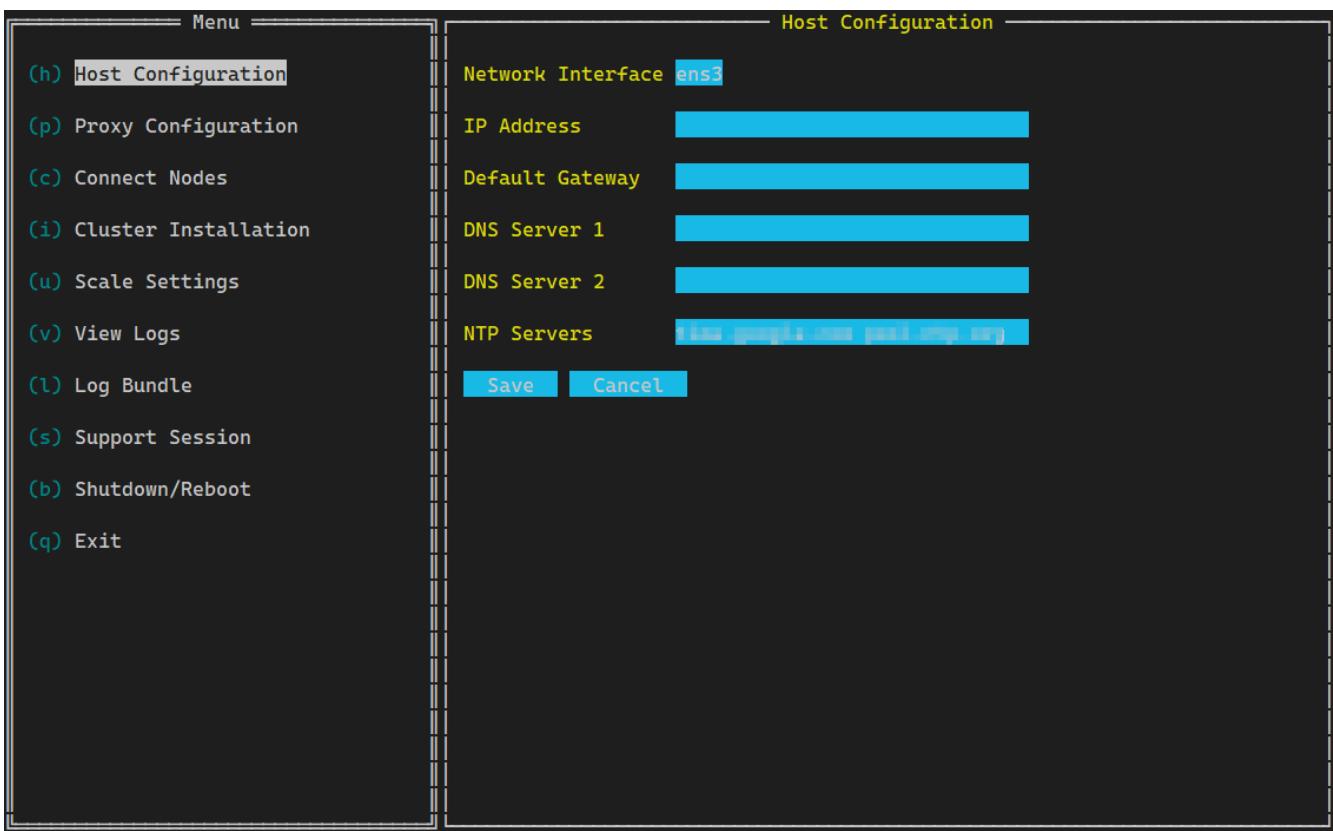
## Task 3. Configure Cortex XSOAR tenant network and IP settings

You need to configure network and IP settings in each node in a cluster. For standalone, there is just a single node.

**NOTE:**

When choosing the network settings, either use private IPs or a public IP covered by an access policy defined in a security group.

1. In the textual UI menu, select Host Configuration.
2. Configure the following network and IP settings for each node/virtual machine.
  - Network interface - A list of available interfaces on the node that the textual UI runs on. For example, ens160
  - IP address - IP address for this node. After deployment, this field will not be editable. For example, 10.196.37.10
  - Default gateway - IP address of the default gateway for this interface. For example, 10.196.37.1
  - DNS server 1 - IP address of the DNS server. For example, 10.196.4.10
  - DNS server 2 (optional) - IP address of a secondary DNS server. For example, 10.196.4.11
  - NTP - The IP address of NTP server that the node will be synced with. By default, the nodes get an out-of-the-box NTP server, you can override the value.



3. Select Save.

#### Task 4. (Optional) Configure proxy settings

If you want to use a proxy, define the proxy address and port settings. The proxy can be set at any point, during Cortex XSOAR deployment or at a later stage.

1. From the textual UI menu, select Proxy Configuration.
2. Configure the following settings.

- Proxy Address

**NOTE:**

Enter the address as IP:port without a http:// or https:// prefix.

- Proxy Port

3. Select Save.

#### Task 5. Establish trust between all nodes in a cluster

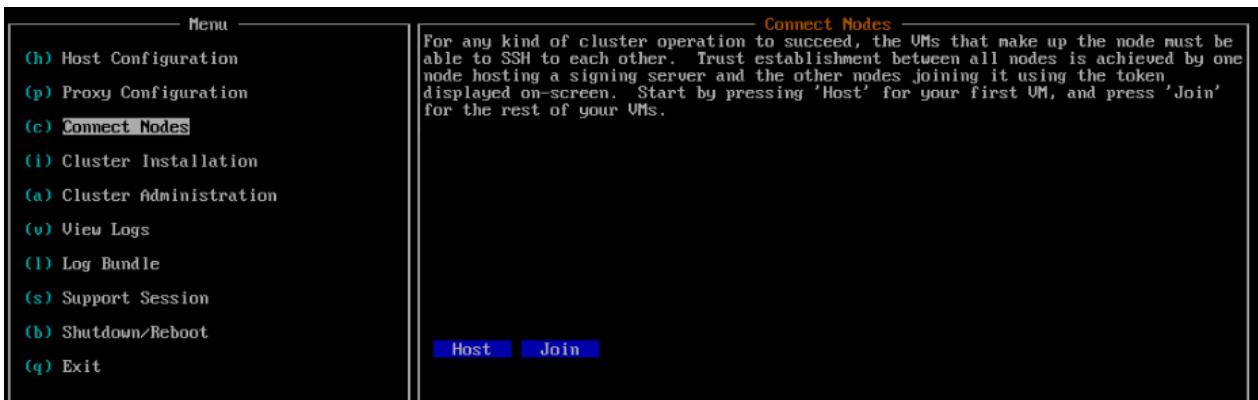
This task is not relevant for a standalone deployment (single node).

For each VM (node) in a cluster, the nodes must have SSH connections between them. Establish trust between all the nodes in a cluster by declaring one node as host for a signing server and the other nodes connecting to it using a token displayed on screen by the host.

#### **IMPORTANT:**

You need to set the host again and reestablish trust between all the nodes if you want to add more nodes to the cluster after completing installation.

1. In the textual UI menu for the VM you want to be the host, select Connect Nodes.
2. Select Host.



A message displays that this action cancels prior trust established with other nodes. Select Yes to continue.

This node becomes the host, and a token is generated on the screen. Copy the token.

#### **NOTE:**

Keep this window open until trust is established between all nodes to enable the host to listen for the token from the other nodes.

3. In the textual UI for each additional node (VM) in the cluster:

- a. Select Connect Nodes.
- b. Select Join.
- c. Paste the Token generated for the host.
- d. Enter the Host IP Address.
- e. Select Submit.



A message displays that this action cancels prior trust established with other nodes. Select Yes to continue.

4. Select OK.
5. After trust is established between all the nodes in the cluster, go back to the host node and close the listening window.

#### Task 6. Install Cortex XSOAR

You are now ready to install Cortex XSOAR on the VMs deployed on your hypervisor.

#### **PREREQUISITE:**

Add DNS records that point the following host names to the cluster IP address.

- Cluster FQDN - The Cortex XSOAR DNS name for accessing the UI. For example, `xsoar.mycompany.com`.
- API-FQDN - The Cortex XSOAR DNS name that is mapped to the API IP address. For example, `api-xsoar.mycompany.com`.
- ext-FQDN - The Cortex XSOAR DNS name that is mapped to external IP address. For example, `ext-xsoar.mycompany.com`.

#### 1. From the textual UI menu, select Cluster Installation.

The virtual machine you use to run the installer will deploy Cortex XSOAR on all virtual machines in a cluster.

For a single virtual machine (standalone), configure the settings for a single node.

#### 2. Configure the following settings.

##### **IMPORTANT:**

For High Availability to work, the IPs of all VMs (nodes) in a cluster as well as the virtual IP must be in the same subnet.

- Cluster Nodes: A list of IPs of all virtual machines/nodes in the cluster, separated by a space. For example, `10.196.37.10 10.196.37.11`  
`10.196.37.12`
  - For an OCI deployment: Copy the IP of each VM from the Private IPv4 address in the OCI Instance information tab and paste it in this field, separated by a space.
  - For an AWS deployment: Copy the IP of each VM from the Private IPv4 address field in the AWS EC2 → Instances → Instance summary page and paste it in this field, separated by a space.
- Cluster FQDN: The Cortex XSOAR environment DNS name. For example, `<subdomain>.<domain name>.<top level domain>`
  - For an OCI deployment: Copy the FQDN from the Internal FQDN field in the OCI Instance information tab and paste it in this field.
  - For an AWS deployment: Copy the FQDN from the Public IPv4 DNS field in the AWS EC2 → Instances → Instance summary page and paste it in this field.

##### **NOTE:**

This name must be registered in your DNS server so the FQDN will be resolved to the IP of the node if it is a single node, or to the IP of the entire cluster if using the built-in virtual IP feature. If you use your own load balancer, you need to register the FQDN to match the IP of the load balancer.

Cortex XSOAR supports only static IP addresses for each virtual machine in the cluster, it does not support a DHCP (dynamic IP) network interface.

- Virtual IP (optional): The Cortex XSOAR environment virtual IP for the multi-node cluster. It must be a different IP than the IPs of the nodes. It is a virtual interface assigned to one of the nodes to provide load balancing to the cluster. For more details, see Load balancing for Cortex XSOAR.

##### **IMPORTANT:**

For Cloud deployments such as OCI or AWS, do not fill in this field (Cortex XSOAR does not support Virtual IPs in Cloud deployments).

- Cluster Region: The region the cluster is located in. For example, US.
- Cortex XSOAR Admin Email, Password, and Confirm Password: These are credentials for the first user to log in to Cortex XSOAR .

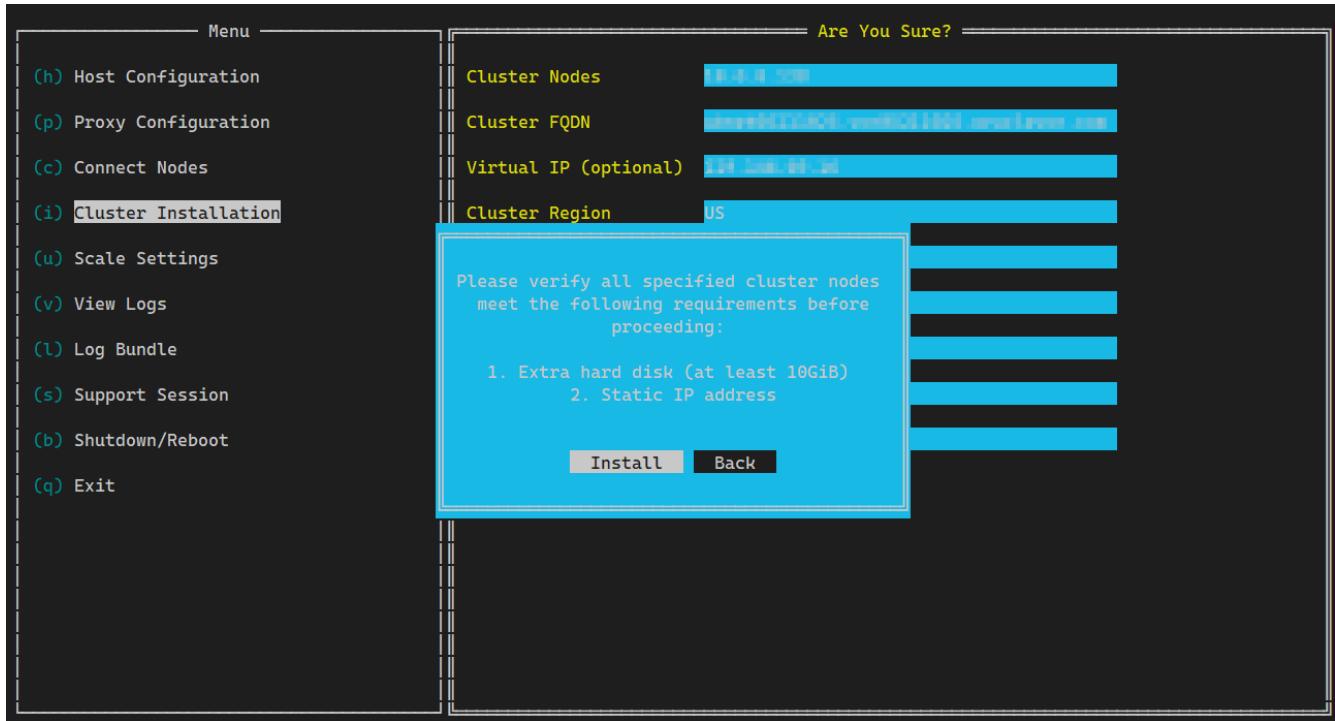
##### **NOTE:**

The password must be at least eight characters long and contain at least:

- One lower case letter
- One upper case letter
- One number, or one of the following special characters: !@#%

#### 3. Select Install.

Verify all nodes meet the required hardware and network requirements, and select Install again.



The virtual machine you use to run the installer will deploy Cortex XSOAR on all virtual machines in a cluster.

#### Task 7. Verify you can log in to Cortex XSOAR

After installation completes, verify that you can log in to Cortex XSOAR and upload your license. If you do not upload your license, all pages are disabled.

1. Log in to Cortex XSOAR.

When you log in for the first time, use the Admin password and email you set during installation.

2. Upload your license to Cortex XSOAR.

For more information, see Add the Cortex XSOAR license.

### 3.5 | Install Cortex XSOAR from a VHD image

#### Abstract

Download a VHD image from Cortex Gateway, deploy the image, and use the textual user interface to configure network, IP, and environment settings, and to install a Cortex XSOAR tenant.

#### PREREQUISITE:

- To be able to download the Cortex XSOAR 8 images from Cortex Gateway, you need a license (or evaluation license via sales) assigned to your CSP account.
- Review the System requirements for deploying a Cortex XSOAR tenant.
- Have a basic understanding of how to deploy VHD file formats.
- Add DNS records that point the following host names to the cluster IP address.
  - Cluster FQDN - The Cortex XSOAR DNS name for accessing the UI. For example, `xsoar.mycompany.com`.
  - API-FQDN - The Cortex XSOAR DNS name that is mapped to the API IP address. For example, `api-xsoar.mycompany.com`.
  - ext-FQDN - The Cortex XSOAR DNS name that is mapped to external IP address. For example, `ext-xsoar.mycompany.com`.

#### Task 1. Select and download the VHD image and license from Cortex Gateway

#### TIP:

In Google Chrome, to download the image and license files together, you may need to set the browser Settings → Privacy and security → Site settings → Additional permissions → Automatic downloads to the default behavior Sites can ask to automatically download multiple files.

1. Log in to Cortex Gateway. For your Cortex XSOAR license, select Download On Prem.

By default, the Production-Standalone license is selected. You can also select Dev.

Production and development are separate Kubernetes clusters with no dependency between them. For example, you can deploy a three-node cluster for production and a standalone node for development. Or you can support small scale for development and large scale for production.

2. Click Next.
3. Select the VHD image format to download.

VHD is supported by Microsoft Hyper-V.

4. Select the checkbox to agree to the terms and conditions of the license and click Download.

Two files download: A zipped license file containing one or more JSON license files with instructions, and a zipped image file of the type you selected (.ova, .vhd)

5. Extract (unzip) the license and image files.

## Task 2. Deploy your virtual machine

The following is an example of deploying your VM on Hyper-V from a VHD image.

If you set your Cortex XSOAR environment as a standalone (single node), you cannot add nodes to it and switch to a cluster. If you deploy three nodes, you can later add nodes and expand the cluster. For more information, see Add or remove nodes in a cluster.

1. Open the Hyper-V manager.
2. Create a new hard disk. This additional hard drive will contain the application data.
  - a. In the Hyper-V manager menu select Action → New → Hard disk → next.
  - b. Select VHD and then fixed size.
  - c. Name the new drive and set its location to the dedicated hard disk you prepared to contain the application data.
  - d. Select Create a new blank virtual hard disk and set its size. For more details, see the System requirements.
  - e. Click finish. This may take a few minutes to complete.

3. Create a new virtual machine.
  - a. In the Hyper-V manager menu select Action → New → Virtual Machine and follow the instructions.
  - b. Name your machine.
  - c. Choose Generation 1.
  - d. Disable storage DRS for the virtual machine.
  - e. Set the memory size. For more details, see the System requirements.

### IMPORTANT:

Every virtual machine is provided with a 256 GB hard disk to run the OS. However, you also need to add an extra hard disk for each virtual machine instance you want to deploy to run the application.

All virtual machines in a cluster must have the same storage size.

To ensure successful deployment, make sure the hard disks meet performance requirements detailed in the System requirements.

- f. Choose the relevant virtual network switch.
- g. Choose Use an existing virtual Hard Disk and browse to the location of the VHD image.
- h. Click finish.

4. Configure the virtual machine.
  - a. Right click the virtual machine and select Settings.
  - b. Under Processor, set the number of processors. For more details, see the System requirements.
  - c. Under IDE controller 0 → Hard drive, click add → virtual hard disk. Choose the hard disk created in Step 2.

5. Start the virtual machine.

6. Repeat this procedure from Step 2 for each additional virtual machine in the cluster.

7. Log in to each virtual machine console. When logging in for the first time, enter the default user name admin and password admin and then create a new password.

### NOTE:

The password must be at least eight characters long and contain at least:

- One lower case letter
- One upper case letter
- One number, or one of the following special characters: !@#%

The textual UI menu appears with all the configuration and installation options.

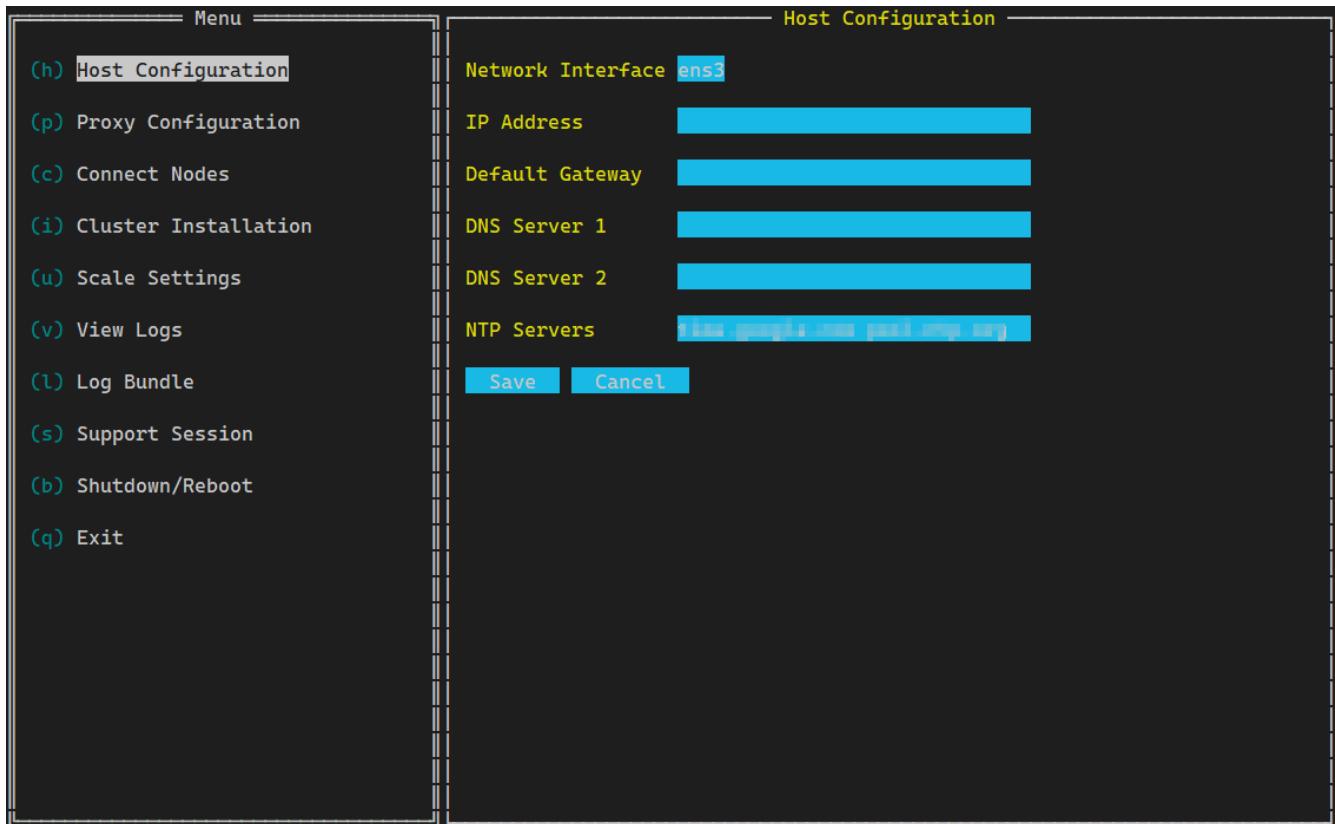
### Task 3. Configure tenant network and IP settings for each node

You need to configure network and IP settings in each node in a cluster. For standalone, there is just a single node.

#### NOTE:

When choosing the network settings, either use private IPs or a public IP covered by an access policy defined in a security group.

1. In the textual UI menu, select Host Configuration.
2. Configure the following network and IP settings for each node/virtual machine.
  - Network Interface: A list of available interfaces on the node that the textual UI runs on. For example, ens160
  - IP Address: IP address for this node. After deployment, this field will not be editable. For example, 10.196.37.10
  - Default Gateway: IP address of the default gateway for this interface. For example, 10.196.37.1
  - DNS Server 1 - IP address of the DNS server. For example, 10.196.4.10
  - DNS Server 2 (optional) - IP address of a secondary DNS server. For example, 10.196.4.11
  - NTP Servers: The IP address of NTP server that the node will be synced with. By default, the nodes get an out-of-the-box NTP server, you can override the value.



3. Select Save.

### Task 4. (Optional) Configure proxy settings

If you want to use a proxy, define the proxy address and port settings. The proxy can be set at any point, during Cortex XSOAR deployment or at a later stage.

1. From the textual UI menu, select Proxy Configuration.
2. Configure the following settings.

- Proxy Address

**NOTE:**

Enter the address as IP:port without a http:// or https:// prefix.

- Proxy Port

## 3. Select Save.

## Task 5. Establish trust between all nodes in a cluster

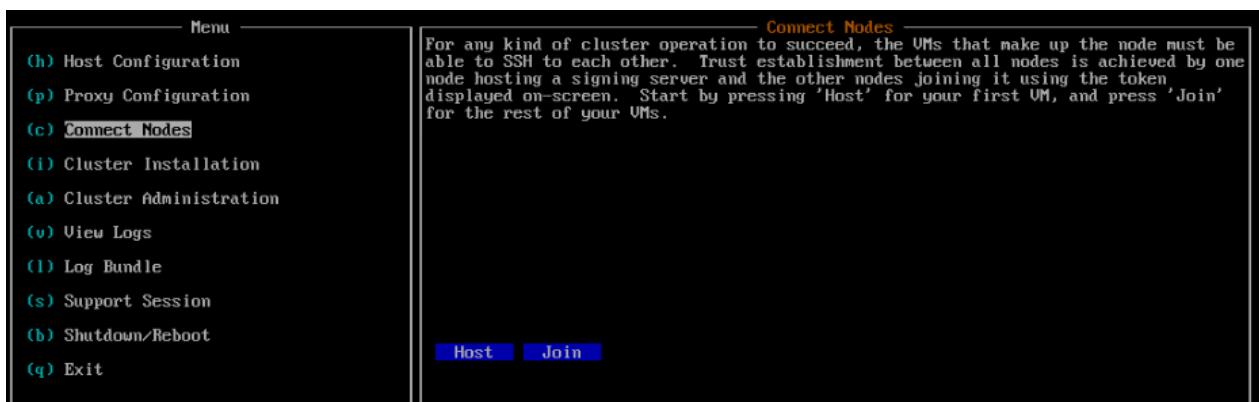
This task is not relevant for a standalone deployment (single node).

For each VM (node) in a cluster, the nodes must have SSH connections between them. Establish trust between all the nodes in a cluster by declaring one node as host for a signing server and the other nodes connecting to it using a token displayed on screen by the host.

**IMPORTANT:**

You need to set the host again and reestablish trust between all the nodes if you want to add more nodes to the cluster after completing installation.

1. In the textual UI menu for the VM you want to be the host, select Connect Nodes.
2. Select Host.



A message displays that this action cancels prior trust established with other nodes. Select Yes to continue.

This node becomes the host, and a token is generated on the screen. Copy the token.

**NOTE:**

Keep this window open until trust is established between all nodes to enable the host to listen for the token from the other nodes.

3. In the textual UI for each additional node (VM) in the cluster:
  - a. Select Connect Nodes.
  - b. Select Join.
  - c. Paste the Token generated for the host.
  - d. Enter the Host IP Address.
  - e. Select Submit.



A message displays that this action cancels prior trust established with other nodes. Select Yes to continue.

4. Select OK.
5. After trust is established between all the nodes in the cluster, go back to the host node and close the listening window.

## Task 6. Install Cortex XSOAR

You are now ready to install Cortex XSOAR on the VMs deployed on your hypervisor.

### PREREQUISITE:

Add DNS records that point the following host names to the cluster IP address.

- Cluster FQDN - The Cortex XSOAR DNS name for accessing the UI. For example, `xsoar.mycompany.com`.
- API-FQDN - The Cortex XSOAR DNS name that is mapped to the API IP address. For example, `api-xsoar.mycompany.com`.
- ext-FQDN - The Cortex XSOAR DNS name that is mapped to external IP address. For example, `ext-xsoar.mycompany.com`.

1. From the textual UI menu, select Cluster Installation.

The virtual machine you use to run the installer will deploy Cortex XSOAR on all virtual machines in a cluster.

For a single virtual machine (standalone), configure the settings for a single node.

2. Configure the following settings.

### IMPORTANT:

For High Availability to work, the IPs of all VMs (nodes) in a cluster as well as the virtual IP must be in the same subnet.

- Cluster Nodes: A list of IPs of all virtual machines/nodes in the cluster, separated by a space. For example, `10.196.37.10 10.196.37.11 10.196.37.12`
  - For an OCI deployment: Copy the IP of each VM from the Private IPv4 address in the OCI Instance information tab and paste it in this field, separated by a space.
  - For an AWS deployment: Copy the IP of each VM from the Private IPv4 address field in the AWS EC2 → Instances → Instance summary page and paste it in this field, separated by a space.
- Cluster FQDN: The Cortex XSOAR environment DNS name. For example, `<subdomain>.<domain name>.<top level domain>`
  - For an OCI deployment: Copy the FQDN from the Internal FQDN field in the OCI Instance information tab and paste it in this field.
  - For an AWS deployment: Copy the FQDN from the Public IPv4 DNS field in the AWS EC2 → Instances → Instance summary page and paste it in this field.

### NOTE:

This name must be registered in your DNS server so the FQDN will be resolved to the IP of the node if it is a single node, or to the IP of the entire cluster if using the built-in virtual IP feature. If you use your own load balancer, you need to register the FQDN to match the IP of the load balancer.

Cortex XSOAR supports only static IP addresses for each virtual machine in the cluster, it does not support a DHCP (dynamic IP) network interface.

- Virtual IP (optional): The Cortex XSOAR environment virtual IP for the multi-node cluster. It must be a different IP than the IPs of the nodes. It is a virtual interface assigned to one of the nodes to provide load balancing to the cluster. For more details, see Load balancing for Cortex XSOAR.

### IMPORTANT:

For Cloud deployments such as OCI or AWS, do not fill in this field (Cortex XSOAR does not support Virtual IPs in Cloud deployments).

- Cluster Region: The region the cluster is located in. For example, US.
- Cortex XSOAR Admin Email, Password, and Confirm Password: These are credentials for the first user to log in to Cortex XSOAR .

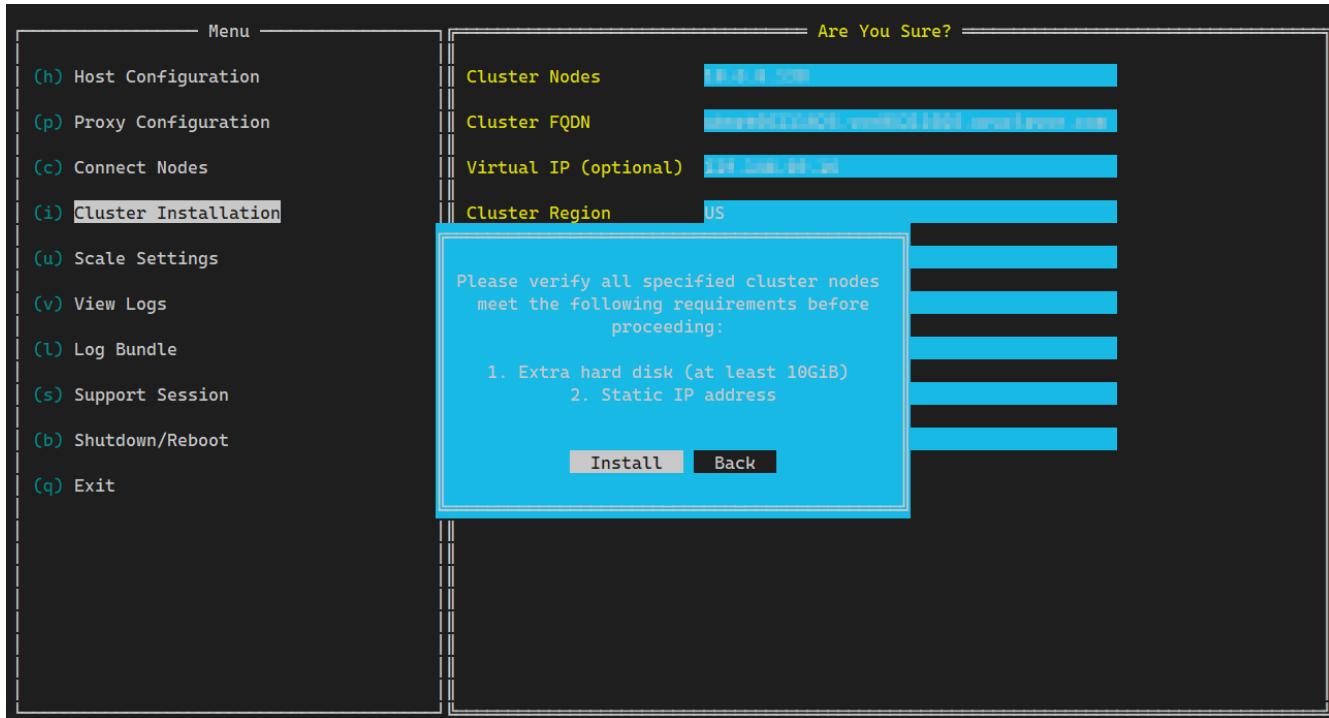
### NOTE:

The password must be at least eight characters long and contain at least:

- One lower case letter
- One upper case letter
- One number, or one of the following special characters: !@#%

3. Select Install.

Verify all nodes meet the required hardware and network requirements, and select Install again.



The virtual machine you use to run the installer will deploy Cortex XSOAR on all virtual machines in a cluster.

#### Task 7. Verify you can log in to Cortex XSOAR

After installation completes, verify that you can log in to Cortex XSOAR and upload your license. If you do not upload your license, all pages are disabled.

1. Log in to Cortex XSOAR.

When you log in for the first time, use the Admin password and email you set during installation.

2. Upload your license to Cortex XSOAR.

For more information, see Add the Cortex XSOAR license.

## 3.6 | Post-installation

### Abstract

Perform optional post-installation maintenance activities from the VM textual UI menu.

After Cortex XSOAR installation is complete, do the following:

- Add the Cortex XSOAR license
- Set up HTTPS with a signed certificate

From the VM textual UI menu you can optionally optimize or troubleshoot system performance with the following:

- Load balancing for Cortex XSOAR
- Add or remove nodes in a cluster
- Scale up hardware resources
- Access logs and log bundles
- Open a support session
- Shut down Cortex XSOAR

### 3.6.1 | Add the Cortex XSOAR license

#### Abstract

Download the Cortex XSOAR license from Cortex Gateway. The license determines which components users can use and how many users can access the tenant.

Cortex XSOAR requires a yearly license per user. Multi-year licenses are available.

After purchasing a license, the activation card for the license is visible in Cortex Gateway. When you install Cortex XSOAR from Cortex Gateway download both the image file and the license. After installing Cortex XSOAR, you must upload the license to Cortex XSOAR. Until you upload a valid license, you will be unable to use Cortex XSOAR.

In the License page (Settings & Info → Cortex XSOAR License) you can see the following:

- License type
- Expiration date
- Number of licensed users
- Number of active users

1. Locate the license file you downloaded from the Cortex Gateway.

If you selected Dev/Prod, you should have two license files, one for each environment. Each should be uploaded separately to the corresponding environment.

2. Upload the license to Cortex XSOAR.

1. Go to Settings & Info → Cortex XSOAR License.
2. In the Upload License section, either drag and drop your license file or browse your files to select the license file. The license file is in JSON format.

**NOTE:**

If you upload a new license while your current license is still valid, the new license will override your existing license for the same product. Other products' licenses will not be affected.

### 3.6.2 | HTTPS with a signed certificate

#### Abstract

Use HTTPS with a signed certificate in Cortex XSOAR. Concatenate the certificate chain.

By default, the tenant uses a self-signed certificate for a secure HTTP connection. TLS versions 1.2 and 1.3 are supported.

#### Create a self-signed certificate

We recommend using a self-signed certificate only for development environments. Follow these steps to create a self-signed certificate.

#### Task 1. Create the certificate

1. Open an SSH session to the Cortex XSOAR tenant.

```
ssh viewer@<host IP address>
```

2. Generate the private key and the certificate. For example:

```
openssl req -newkey rsa:4096 -x509 -sha256 -days 3650 -out example.crt -keyout example.key
```

**NOTE:**

- While the example is generic, you might need to create your certificates and keys with different parameters according to your internal company policies or compliance with regulations.
- If you prefer to create a key without a passphrase, add the **-nodes** flag

Flag	Description
<code>-newkey rsa:4096</code>	Generates a 4096-bit RSA new private key. The default RSA key is 2048 bits.
<code>-x509</code>	Creates an X.509 certificate.
<code>-sha256</code>	Uses 256-bit SHA (Secure Hash Algorithm).

Flag	Description
-days 3650	The number of days for which to certify the certificate. 3650 is ten years. You can use any positive integer.
-out example.csr	Specifies the file name for the newly created certificate signing request. You can specify any file name.
-keyout example.key	Specifies the file name for the newly created private key. You can specify any file name.

#### Task 2. Apply the certificate

1. Open an SSH session to the Cortex XSOAR tenant.

```
ssh viewer@<host IP address>
```

2. Apply the key and certificate files that should be used as the HTTPS certificate for the tenant.

```
sudo ./sbin/set_ssl_certificate --key example.key --cert example.crt
```

#### Install or renew a custom certificate from a Certificate Authority

If you want to use your own certificate (X.509 certificates), you can install or renew a custom certificate. For security reasons, the default certificate for a production environment must be replaced with your private key and a certificate from a Certificate Authority (CA). For development environments, you either use a self-signed certificate or a certificate from a CA.

#### Task 1. Create a private key and a Certificate Signing Request (CSR)

The following example is one way to create a private key and a CSR on a Linux-based system.

##### NOTE:

While this example is generic, you might need to create your certificates and keys with different parameters according to your internal company policies or compliance with regulations.

1. Open an SSH session to the Cortex XSOAR tenant.

```
ssh viewer@<host IP address>
```

2. Generate the certificate signing request and the private key. The certificate signing request is for the URL that will be publicly available for everyone and also includes all public-facing aliases.

```
openssl req -newkey rsa:4096 -x509 -keyout example.key -out example.csr
```

##### NOTE:

- The FQDN must be provided as the Common Name (CN) when generating the CSR and private key.
- To create a key without a passphrase, add the **-nodes** flag.

Flag	Description
-newkey rsa:4096	Creates a new certificate request and a 4096 bit RSA key. The default RSA key is 2048 bits.
-sha256	Uses 256-bit SHA (Secure Hash Algorithm).
-out example.csr	Specifies the file name for the newly created certificate signing request. You can specify any file name.
-keyout example.key	Specifies the file name for the newly created private key. You can specify any file name.

Flag	Description
-addext	Adds desired DNS aliases to the certificate.

3. Save the cert.key file.

4. Send the CSR to the Certificate Authority (CA). The CA should send the certificate by email in multiple formats. For example, example.crt.

**CAUTION:**

Cortex XSOAR tenant does not support PKCS#8 encrypted PEM files. To validate that the file is in a format that is supported, view the encrypted .key file (you can use one of the following commands - vi / less / cat) and check that the DEK-Info header exists.

A certificate with the DEK-Info header begins with the following:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,B94C43E0E49D267EB3AA84DC19EB41ED
VcNSY7T...
```

If the DEK-Info header is not similar to the example above, the file is likely in the wrong format (PKCS#8).

You can convert the .key file to the proper format by running the following command:

```
openssl rsa -in oldcert.key -out cert.key -aes256
```

You don't have to use aes256, you can use des3 or whichever encryption method you prefer.

After you run this command, view the .key file and verify that the DEK-Info header is similar to the example above. This should allow the .key file to be read.

5. For the certificate PEM file, you must concatenate the certificate chain one after the other in the file.

**NOTE:**

- If you are using an intermediate certificate, the order is:
  1. SSL certificate
  2. Intermediate certificate
  3. CA certificate
- If you are not using an intermediate certificate, the order is:
  1. SSL Certificate
  2. CA Certificate
- Only the certificate itself is needed, for example the text between and including "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----".

## Task 2. Apply the certificate to Cortex XSOAR

To replace the default internal certificate with a private key and a certificate from a CA:

1. Open an SSH session to the Cortex XSOAR tenant.

```
ssh viewer@<host IP address>
```

2. Apply the key and certificate files that should be used as the HTTPS certificate for the server.

```
sudo ./sbin/set_ssl_certificate --key /path/to/example.key --cert /path/to/example.crt
```

## Troubleshoot creating a private key and CSR

If the browser does not show the new certificate, after the newly generated certificate key pair is applied, do one or more of the following:

- Check whether the FQDN of the Cortex XSOAR tenant is the same as the CN field of the certificate, or any of the DNS fields in the Certificate Subject Alternative NAME (SAN).
- On your browser on which you are trying to load Cortex XSOAR, clear cookies and other data. For example, in Chrome, go to Settings → Advanced → Clear Browsing data → Clear data.
- If the Cortex XSOAR tenant is behind a load balancer, reupload the certificate on the load balancer. For example, if the Cortex XSOAR tenant is behind ELB (Elastic Load Balancing), re-import the certificate on ELB on the Amazon Certificate Manager AWS console.

### 3.6.3 | Load balancing for Cortex XSOAR

If you deploy a cluster of three or more nodes, you can optionally set up an internal load balancer by defining a virtual IP in the textual UI tool installation step, or you can set up an external load balancer.

To set up an external load balancer:

1. From the textual UI menu, select Cluster Installation.
2. Set the access to the cluster nodes through port 443.
3. Use HTTP 10254 with the path /healthz as the health endpoint.
4. Select Install.

### 3.6.4 | Add or remove nodes in a cluster

#### Abstract

Add, drain, remove, taint, or uncordon a node in a cluster under the Cluster Administration textual UI menu item.

If you deployed your Cortex XSOAR environment starting with three nodes, using the textual UI menu in your VM you can add a node, taint a node, remove a node, drain a node, and uncordon a node.

#### **IMPORTANT:**

- If you deployed your Cortex XSOAR environment as a standalone (single node), you cannot add nodes to it and switch to a cluster.
- If you remove one of the control planes (the original three nodes in a cluster), you cannot perform actions such as upgrade or scaling up. You need to add a new node to replace the control plane and assign it the same IP address as the control plane that was removed.

#### Add a node

Add a node to a cluster to increase its capacity, improve performance, or enhance redundancy for better load distribution.

1. From the textual UI menu in your VM, select Cluster Administration.
2. Reestablish trust between all nodes in the cluster, including the new node. For more information, see Task 5. Establish trust between all nodes in a cluster.
3. Select Add Node.
4. Enter the IP Address and click Add.

#### Taint a node

Tainting a node marks the node as out of service for internal K8s functions. Taint a node to stop applications from running on it.

1. From the textual UI menu in your VM, select Cluster Administration.
2. Select the IP address of the node you want to taint.
3. Select Taint.

In the list of nodes in the Cluster Administration menu, the node IP will display as Ready,SchedulingDisabled.

#### Remove a node

Remove a node from a cluster to reduce resources, perform maintenance, or decommission the node, ensuring the cluster operates efficiently without unnecessary or malfunctioning components.

1. From the textual UI menu in your VM, select Cluster Administration.
2. Drain the node.
  - a. Select the IP address of the node you want to drain.
  - b. Select Drain.
3. Remove the node.
  - a. Select the IP address of the node you want to remove.
  - b. Select Remove.

In the list of nodes in the Cluster Administration menu, the node IP will display as Ready,SchedulingDisabled.

In the list of nodes in the Cluster Administration menu, the node IP will display as Ready.

#### Drain a node

Draining a node pauses the node activity in the cluster and marks it as unschedulable. Draining a node safely removes workloads from it, ensuring that running applications are gracefully terminated or moved to other nodes without disrupting service availability before you perform maintenance on the node.

1. From the textual UI menu in your VM, select Cluster Administration.
2. Select the IP address of the node you want to drain.
3. Select Drain.

In the list of nodes in the Cluster Administration menu, the node IP will display as Ready,SchedulingDisabled.

#### Uncordon a node

Uncordon a node in a cluster to make it available again for scheduling new workloads, for example after maintenance or troubleshooting is complete.

1. From the textual UI menu in your VM, select Cluster Administration.
2. Select the IP address of the node you want to uncordon
3. Select Uncordon.

In the list of nodes in the Cluster Administration menu, the node IP will display as Ready.

### 3.6.5 | Scale up hardware resources

#### Abstract

The Scale Settings textual UI menu item enables scaling up resources for CPU, memory, and disk size.

Using the textual UI menu in your VM, you can easily scale up your Cortex XSOAR hardware environment based on your organizational and usage growth.

1. Choose the scale you want and make sure your hardware resources meet the system requirements. For more information, see System requirements.

#### NOTE:

If you are working with more than one node, all the nodes in the cluster must meet the same hardware requirements.

2. From the textual UI menu, select Scale Settings.
3. Select Scan scale options to run a scan to evaluate the cluster recommended scale.

Based on the results, the system indicates the current scale size and gives you the option to increase the scale size.

The supported scale sizes are:

- Small: 16 CPU, 64 GB memory, 1 TB hard disk (1TB = 1024 GB)
- Medium: 32 CPU, 128 GB memory, 1.5 TB hard disk
- Large: 48 CPU, 192 GB memory, 2 TB hard disk

#### NOTE:

The recommended scale is determined by the node with the least hardware resources.

### 3.6.6 | Access logs and log bundles

#### Abstract

Access logs and log bundles from the VM textual UI menu.

Using the textual UI menu in your VM, you can optionally view logs and log bundles.

#### NOTE:

The viewer SSH user can fetch the log bundles and view logs.

Select View Logs to see logs relevant for the textual UI session. These are not logs relevant for the user session in the Cortex XSOAR UI.

Select Log Bundle to download a log bundle for support or engineering. You can also download this log bundle from the Cortex XSOAR UI.

### 3.6.7 | Open a support session

#### Abstract

Open a support session from the VM textual UI menu.

There may be issues related to your Cortex XSOAR deployment that cannot be resolved by the information provided by the log bundles. You can establish a support session with support or engineering to troubleshoot and resolve these issues. Opening a support session temporarily allows the support/engineering person access to your system as a user with elevated permissions in order to troubleshoot or debug issues.

1. (Optional) Share your screen with the support/engineering person.
2. From the textual UI menu in your VM, select Support Session.
3. If the Tenant ID field is not populated, upload the tenant license to Cortex XSOAR and then select Refresh. For more information, see Add the Cortex XSOAR license.

The Tenant ID displays the tenant license.

4. Perform secure authentication.

1. Provide the value from the Challenge field to the support/engineering person who is assigned to help you. They will then provide you with a token.

Clicking Refresh generates a new challenge.

2. Enter the token in the Token field and select Submit.

The support/engineering person now has access to your system in a secure support shell as a user with elevated permissions.

When support/engineering finishes their troubleshooting or debugging activities, they log out of the system. The shell closes and you return to the textual UI menu.

### 3.6.8 | Shut down Cortex XSOAR

#### Abstract

Shut down a session from the VM textual UI menu.

You may need to shut down Cortex XSOAR in order to perform maintenance or troubleshooting activities.

You can gracefully shut down or reboot a cluster by selecting the Shutdown/Reboot menu item from the textual UI.

#### IMPORTANT:

Do not do a hard shutdown.

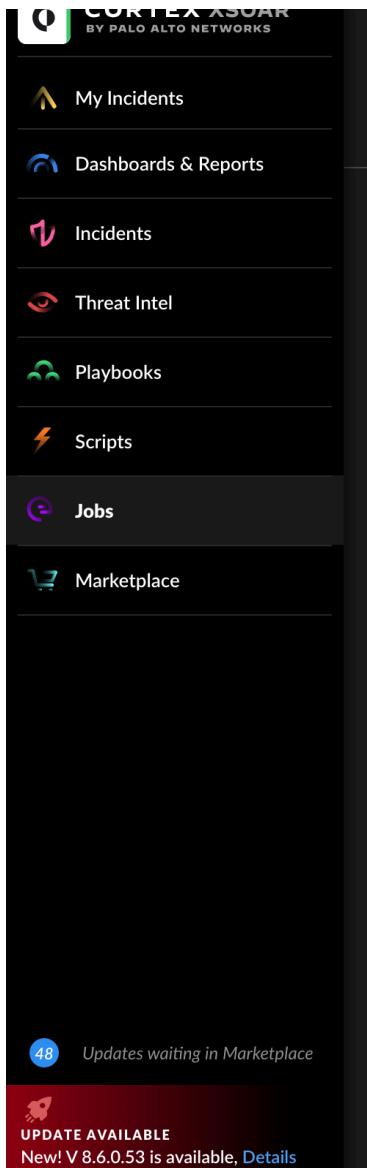
## 3.7 | Update Cortex XSOAR

#### Abstract

Upgrade your Cortex XSOAR On-prem tenant to the latest version.

Cortex XSOAR checks several times a day to determine whether there is a new version available and downloads an update file if it is available. If the update file of the new version of the tenant was downloaded successfully to the tenant, an update notification appears in the left menu pane for all admin users. Only administrators receive the update notifications and can perform an update.

The notification indicates the new version number that is available and contains a link that opens the About page.



You can hide the notification in the main pane for your user so that the next time you log in to the tenant, the notification in the menu pane will not appear until a different new version is available. If a different user logs in to the tenant, the notification will appear. However, the update icon will still appear next to the user's About option providing you with the ability to update the version.

**NOTE:**

When upgrading Cortex XSOAR you update every version consecutively. For example, if you have installed version 8.5, you update to 8.6 and then 8.7. After you update to version 8.6, you will receive a notification for the next version.

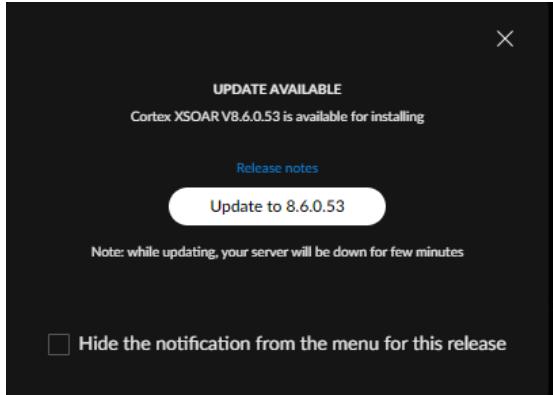
After updating, you can't rollback to a previous version.

1. In the Cortex XSOAR left menu pane, click Continue in the update notification, or go to <Your Username> → About.

The About page appears with a new banner that indicates that an update is available with the new version number.

2. Click Continue.

An Update Available message appears.



3. If you are not going to update now, you can click the Hide the notification from the menu for this release to hide the update notification for this version. The next time you log in to the tenant, the notification will not appear in the main menu of the UI, but an icon will appear in the About option.

4. Click Update to <version number> to proceed with the update process.

A message appears indicating that the system will be down while the data and settings are updated.

5. Click Update Now. When the update is done, if this is a major release, a new version message appears.

## 4 | Engines

### Abstract

Install, manage, configure, and troubleshoot engines.

Install an engine in your remote network, enabling effortless communication with Cortex XSOAR. Easily configure and manage the engine to fit your specific needs, and explore how to leverage it for seamless integrations.

### 4.1 | What is an engine?

An engine is a proxy server application that is installed on a remote machine and enables communication between the remote machine and the Cortex XSOAR tenant. You can run playbooks, scripts, commands, and integrations on the remote machine and the results are returned to the tenant.

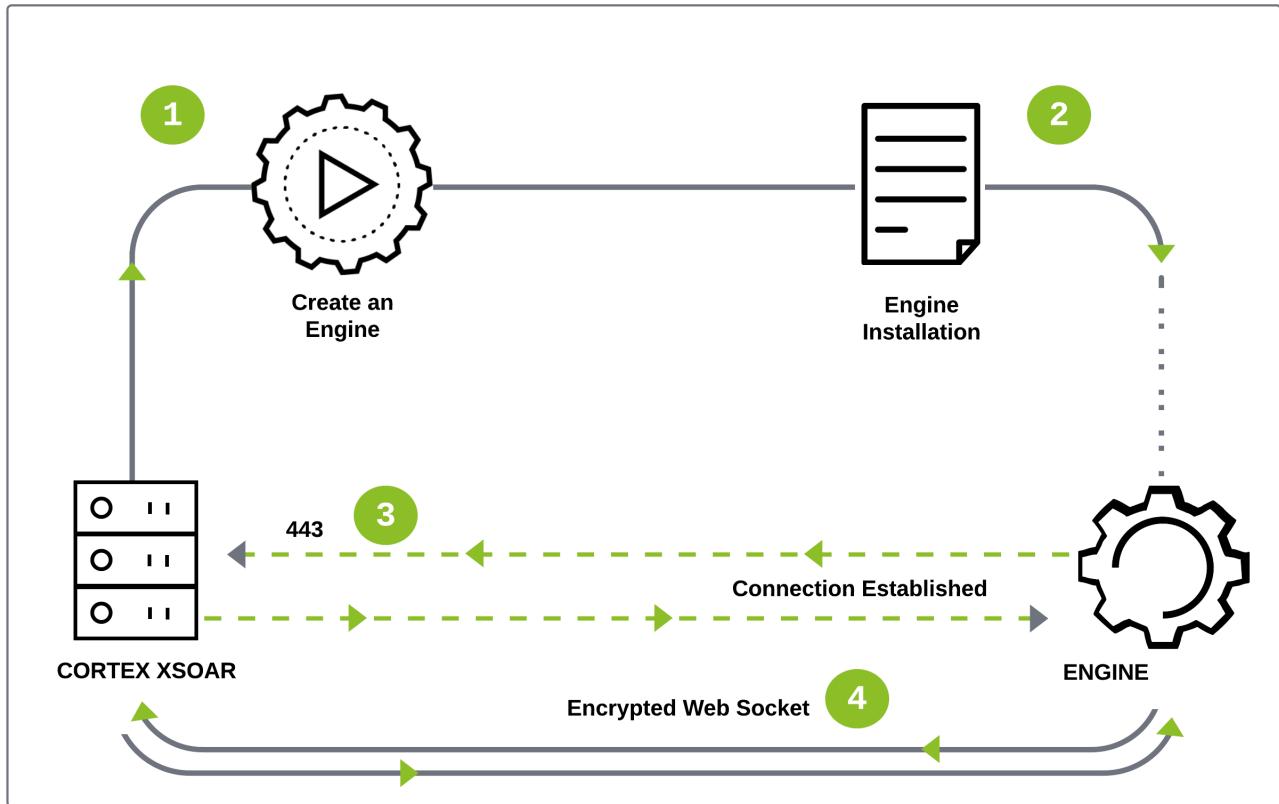
While the Cortex XSOAR tenant includes a user interface that allows security analysts to create and manage playbooks, investigate incidents, and perform other tasks, the engine operates behind the scenes to execute these playbooks and automate security actions. The separation between the user interface and the engine allows for the scalable and efficient execution of security automation and orchestration.

You can install multiple engines on the same machine (Shell installation only) which is useful in a dev-prod environment where you do not want to have numerous engines in different environments and to manage those machines.

#### NOTE:

You cannot share a multiple-engine installation with a single-engine installation.

## Engine architecture



Within the network, you need to allow the engine to access the Cortex XSOAR's IP address and listening port (by default, TCP 443). The engine always initiates the communication to Cortex XSOAR.

## Engine use cases

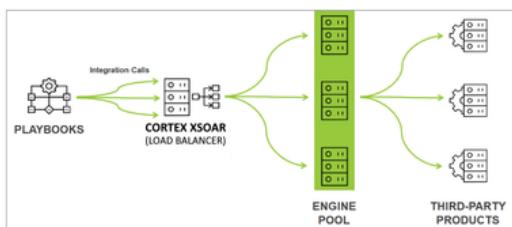
An engine can be used for the following purposes:

- **Engine proxy**

Cortex XSOAR engines enable you to access internal or external services that are otherwise blocked by a firewall or a proxy. For example, if a firewall blocks external communication and you want to run the Rasterize integration, you need to install an engine to access the Internet.

- **Engine load-balancing**

Engines can be part of a load-balancing group, which enables the distribution of the command execution load. The load-balancing group uses an algorithm to efficiently share the workload for integrations that the group is assigned to, thereby speeding up execution time. In general, heavy workloads are caused by playbooks that run a high number of commands.

**NOTE:**

When you add an engine to a load-balancing group, you cannot use that engine separately. The engine does not appear in the engines menu when configuring an integration instance but you can choose the load-balancing group.

## 4.2 | Engine requirements

## Abstract

Hardware, OS, and required URLs for engines.

You can install engines on all Linux machines. Docker/Podman needs to be installed before installing an engine. If you are using the shell installer for an engine, Docker/Podman is installed automatically.

**NOTE:**

The Cron package is required for installing engines on a Linux machine.

#### Engine hardware requirements

If your hard drive is partitioned, we recommend a minimum of 50 GB for the /var partition.

Component	Dev Environment Minimum	Production Minimum
CPU	8 CPU cores	16 CPU cores
Memory	16 GB RAM	32 GB RAM
Storage	100 GB	100 GB

**NOTE:**

If using Podman, we recommend reserving 150 GB for container storage, either in the /home partition or a different storage directory that you have set using the rootless\_storage\_path key. For more information, see Change container storage directory.

#### Operating system requirements

You can deploy a Cortex XSOAR engine on the following operating systems:

Operating System	Supported Versions
Ubuntu	18.04, 20.04, 22.04
RHEL	8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9.0, 9.1, 9.2, 9.3
Oracle Linux	7.x
Amazon Linux	2

**NOTE:**

Centos 8.x reached End of Life (EOL) on December 31, 2021, and is no longer a supported operating system.

Centos 7.x reached End of Life (EOL) on June 30, 2024, and is no longer a supported operating system.

#### Engine required URLs

You need to allow the following URLs for Cortex XSOAR engines to operate properly.

The endpoint URL is: `wss://api-<tenant domain>/xsoar/d1ws`

FUNCTION	SERVICE	PORT	DIRECTION
Integrations		Integration-specific ports	Outbound
Engine connectivity	HTTPS	443 (configurable)	Outbound

FUNCTION	SERVICE	PORT	DIRECTION
Docker	<ul style="list-style-type: none"> <li>• <a href="https://registry-1.docker.io">https://registry-1.docker.io</a></li> <li>• <a href="https://registry.fedoraproject.org">https://registry.fedoraproject.org</a></li> <li>• <a href="https://registry.access.redhat.com">https://registry.access.redhat.com</a></li> <li>• <a href="https://docker.io">https://docker.io</a></li> <li>• <a href="https://registry.docker.io">https://registry.docker.io</a></li> <li>• <a href="https://auth.docker.io">https://auth.docker.io</a></li> </ul> <p>This URL may change according to Docker's discretion.</p> <ul style="list-style-type: none"> <li>• <a href="https://production.cloudflare.docker.com">https://production.cloudflare.docker.com</a></li> </ul> <p>This URL may change according to Docker's discretion.</p>	443	Outbound

## 4.3 | Install an engine

### Abstract

Install, deploy and configure Cortex XSOAR engines.

When you install the engine, the `d1.conf` is installed on the engine machine, which contains engine properties such as proxy, log level, and log files. If Docker/Podman is already installed, the `python.engine.docker` and `powershell.engine.docker` keys are set to `true`. If Docker or Podman is not available when the engine is installed, the key is set to `false`. If so, you need to set the key to `true` after installing Docker and Podman. Verify that `python.engine.docker` and `powershell.engine.docker` configuration keys are present in the `d1.conf` file.

#### NOTE:

If you are using DEB, RPM, or Zip installation, install Docker or Podman.

### Installation types

Cortex XSOAR supports the following file types for installation on the engine machine:

- Shell: For all Linux deployments, including Ubuntu, and SUSE. Automatically installs Docker/Podman, downloads Docker/Podman images, enables remote engine upgrade, and allows installation of multiple engines on the same machine.

The installation file is selected for you. Shell installation supports the purge flag, which by default is false. To uninstall an engine, run the installer with the purge flag enabled.

#### NOTE:

When upgrading an engine that was installed using the Shell installation, you can use the Upgrade Engine feature in the Engines page. For Amazon Linux 2 type engines, you need to upgrade these engine types using a zip type engine and not use the Upgrade Engine feature.

If you use the shell installer, Docker/Podman is automatically installed.

- DEB: For Ubuntu operating systems.
- RPM: RHEL operating systems.

#### NOTE:

Use DEB and RPM installation when shell installation is not available. You need to manually install Docker or Podman and any dependencies.

- Zip: Used for Amazon Linux 2 machines.
- Configuration: Configuration file for download. When you install one of the other options, this configuration file (`d1.conf`) is installed on the engine machine.

#### IMPORTANT:

For DEB/RPM engines, Python (including 3.x) and the containerization platform (Docker/Podman) must be installed and configured. For Docker or Podman to work correctly on an engine, IPv4 forwarding must be enabled.

### How to install an engine

1. Create an engine.

- Select Settings & Info → Settings → Integrations → Engines → Create New Engine.
- In the Engine Name field, add a meaningful name for the engine.
- Select one of the installer types from the list.
- (Optional) (Shell only) Select the checkbox to enable multiple engines to run on the same machine.

If you have an existing engine, and you did not select the checkbox, and now you want to install another engine on the same machine, you need to delete the existing engine.

- (Optional) Add any required configuration in JSON format.
- Click OK to create the engine.

2. For shell installation, do the following:

**TIP:**

For Linux systems, we recommend using the shell installer. If using Amazon Linux 2, use the zip installer (see step 4).

- Move the .sh file to the engine machine using a tool such as SSH or PuTTY.
- On the engine machine, grant execution permission by running the following command:

```
chmod +x <engine-file-path>
```

- Install the engine by typing one of the following commands:

With tools: `sudo <engine-file-path>`

Without tools: `sudo <engine-file-path> -- -tools=false`

If you receive a `permissions denied` error, it is likely that you do not have permission to access the `/tmp` directory.

3. For RPM/DEB installation, do the following:

- Move the file to the required machine using a tool such as SSH or PuTTY.
- Type one of the following installation commands:

Machine Type	Install Command
RHEL (RPM)	<code>sudo rpm -Uvh d1-2.5_15418-1.x86_64.rpm</code>
Ubuntu (DEB)	<code>sudo dpkg --install d1_xxx_amd64.deb</code>

- Start the engine by running one of the following commands:

Machine Type	Start Command
RHEL (RPM)	<code>sudo systemctl start d1</code>
Ubuntu (DEB)	<code>sudo service d1 restart</code>

4. For Zip installation on Amazon Linux 2, run the following commands:

- Create the engine folder.

```
mkdir /usr/local/demisto
```

- Unzip the engine files to the folder created in the previous step.

```
unzip ./d1.zip -d /usr/local/demisto
```

c. Allow the process to bind to low numbered ports.

```
setcap CAP_NET_BIND_SERVICE=+eip /usr/local/demisto/d1_linux_amd64
```

d. Change the owner of /usr/local/demisto to the demisto user.

```
chown -R demisto:demisto /usr/local/demisto
```

e. In /etc/systemd/system edit the d1.service file as follows (adjust the directory and the name of the binaries file if needed).

```
[Unit]
Description=Demisto Engine Service
After=network.target
[Service]
Type=simple
User=demisto
WorkingDirectory=/usr/local/demisto
ExecStart=/usr/local/demisto/d1_linux_amd64
EnvironmentFile=/etc/environment
Restart=always
[Install]
WantedBy=multi-user.target
```

f. Run the following commands:

```
chown root:root /etc/systemd/system/d1.service
```

```
chmod 644 /etc/systemd/system/d1.service
```

g. Run the engine process.

```
systemctl start d1
```

h. Verify that the engine is running.

```
systemctl status d1
```

5. Verify that the engine you created is connected.

a. Select Settings & Info → Settings → Integrations → Engines.

b. Locate your engine on the Engines page and check that it is connected.

6. When the engine is connected, you can add the engine to a load-balancing group by clicking Load-Balancing Group on the Engines page.

If you want to add the engine to a new group, click Add to new group from the list.

When the engine is in the load-balancing group, it cannot be used as an individual engine and does not appear when configuring an engine from the list.

7. (Optional) After installing the engine, you may want to set up a proxy, set up Docker hardening, configure the number of workers for the engine, or perform other related engine configurations. For more information, see the Configure Engines section. You can also configure an integration instance to run on the engine you created.

#### **NOTE:**

If the installer fails to start due to a permissions issue, even if running as root, add one of the following two arguments when running the installer:

- --target <path> - Extracts the installer files into the specified custom path.
- --keep - Extracts the installer files into the current working directory (without cleaning at the end).

If using installer options such as -- -tools=false, the option should come after the --target or --keep arguments. For example:

```
sudo ./d1-installer.sh --target /some/temp/dir -- -tools=false
```

### 4.3.1 | Engine air gap installation

#### Abstract

Install a Cortex XSOAR engine offline when you don't have access to the Internet (tested on RHEL v8).

An air gap is a security measure that involves isolating a computer or network and preventing it from establishing an external connection. An air-gapped computer is physically segregated and incapable of connecting wirelessly or physically with other computers or network devices.

Use these instructions to install an engine on a machine without internet connectivity.

On a machine that has internet access, you need to download dependencies, Docker images, and from the Cortex XSOAR tenant, the engine installation files. You then need to transfer and install the files to the machine without internet access.

#### Download dependencies for offline installation

Install the following top level dependencies according to your operating system. These dependencies may be dependent upon other OS libraries.

##### **NOTE:**

Always verify that your dependencies are updated and take into account that they might change across releases.

#### RPM dependencies

The following dependencies are required for Red Hat deployments.

- systemd
- xmlsec1
- xmlsec1-openssl
- rpm-build
- libcap
- dnf-utils
- file
- fontconfig
- expat
- libpng
- freetype
- git
- makeself

Run the following commands:

- sudo yum check update
- sudo yum install <name of the dependency>
- (Red Hat v8 & above) If using Podman, you need to run:
  - sudo yum -y install slirp4netns fuse-overlayfs
  - sudo yum -y module install container-tools

#### Debian dependencies

The following dependencies are required for Debian and Ubuntu deployments:

- systemd
- xmlsec1
- rpm
- libcap2-bin
- file
- libfontconfig1
- libfreetype6
- git
- makeself

Run the following commands

- sudo apt-get update
- sudo apt-get install <name of the dependency>

#### Download Docker images offline

To download Docker images you need to use the `download_packs_and_docker_images` script to download the docker image according to the content pack integration you want to use, such as AWS-ILM, Cybereason, and EWS.

The `download_packs_and_docker_images` script enables you to download the latest content packs Docker images in a zip folder to your machine. The script is located in the `Utils` folder in the GIT Content repository. If you do not have access to the GIT Content repository, you can download the script from here. For detailed information and how to download the Docker images, see [download packs offline](#).

#### Install an engine offline

1. On a machine with internet access, download the following:

- a. Dependencies for your deployment type.
- b. Relevant Docker images.

2. In the Cortex XSOAR tenant, download the engine installation file.

- a. Select **Settings & Info** → **Settings** → **Integrations** → **Engines** → **Create New Engine**.

- b. In the **Engine Name** field, add a meaningful name for the engine.

- c. Select one of the installer types from the list.

For Linux systems we recommend using the **Shell** installer.

- d. (Optional) If you want to add the engine to a load balancing group, from the list, select the group.

The list only appears after you have created and connected an engine and created a load balancing group. To add the engine to a new group, select **Add new group** from the list.

The engine cannot be used as an individual engine and does not appear when configuring an engine from the list.

- e. (Optional) (Shell only) Select the checkbox to enable multiple engines to run on the same machine.

If you have an existing engine, you did not select the checkbox, and you want to install another engine on the same machine, you need to delete the existing engine.

- f. (Optional) Add any required configuration in JSON format.

- g. Click **Create New Engine**.

3. On the machine you want to install the engine, do the following:

- a. Transfer the files downloaded in steps 1 and 2.

- b. Verify that the required dependencies in step 1a is installed successfully by running one of the following commands.

- (Red Hat) `repoquery -a --installed`
- (Ubuntu or Debian) `apt list --installed`

- c. Install the engine.

1. Grant execution permission by running the following command:

```
chmod +x <engine-file-path>
```

2. Install the engine by running the following command:

```
sudo ./d1-<engine-name>-<XSIAM-version>-xxxxxx.sh -- -tools=false -do-not-start-engine=true
```

For example, `sudo ./d1-engine1-8.35-318874.sh -- -tools=false -do-not-start-engine=true`

If you receive a **permissions denied** error, it is likely that you do not have permission to access the `/tmp` directory.

- d. (Red Hat v8 & above) If you have not already done so, install and configure Podman, by following the steps in [Migrate From Docker to Podman](#) (from step 2 onwards).

- e. Load the Docker images that you downloaded in step 1b, by doing one of the following:

- (Ubuntu, Debian, Red Hat v7 & below) Run the following command:

```
sudo docker load -i <YOUR_DOCKER_FILE>.zip
```

- (Red Hat v8 & above) Do the following:

1. Ensure that the Docker file has `demisto:demisto` ownership.

2. Ensure that you are in the root directory (`cd /`).

3. Run the following commands:

```
sudo -su demisto
```

```
podman load -i <YOUR_DOCKER_FILE>.zip
```

4. (Optional) To verify that images are able to run, use the `podman images` command. You can also run the `podman images -q "demisto/python:1.3-alpine"` command to validate specific images and identify any issues.

4. Start the engine, by running the following command:

```
sudo systemctl start d1
```

**NOTE:**

For multiple engines the d1 service name may differ.

5. (Optional) After installation has completed, do the following:

- a. Confirm that the engine status is active, by running the `systemctl status d1` command.

- b. Validate that the engine is connected and running by going to Settings & Info → Settings → Integrations → Engines.

- c. Run the engine on a sample integration. For example, go to Settings & Info → Settings → Integrations → Instances and search for the Hello World (Community Contribution) integration. Add or edit the instance and in the Run on field, select the engine.

- d. Run a simple command to test that the engine is working properly using the integration.

For example, `!helloworld-say-hello name"Hello"`

### 4.3.2 | Docker

#### Abstract

Cortex XSOAR Docker installation, configuration, security, and troubleshooting guides.

Docker is a software framework for building, running, and managing containers.

**NOTE:**

This section is relevant when installing an engine.

Cortex XSOAR maintains a repository of Docker images, available in the Docker hub under the Cortex organization.

Each Python/PowerShell script or integration has a specific Docker image listed in the YAML file. When the script or integration runs, if the specified Docker image is not available locally, it is downloaded from the Docker hub or the Cortex Container Registry. The script or integration then runs inside the Docker container. For more information on Docker, see the Docker documentation and Using Docker.

**NOTE:**

Docker images can be downloaded together with their relevant content packs, for offline installation.

#### 4.3.2.1 | Install Docker

#### Abstract

Install Docker on engines and troubleshoot the installation.

Docker is required for engines to run Python/Powershell scripts and integrations in a controlled environment.

If you use the Shell installer to install an engine, Docker is automatically installed. If using DEB and RPM installations, you need to install Docker or Podman before installing an engine. The engine uses Docker to run Python scripts, PowerShell scripts, and integrations in a controlled environment. By packaging libraries and dependencies together, the environment remains the same and scripts and integrations are not affected by different server configurations.

Cortex XSOAR supports the latest Docker Engine release from Docker and the following corresponding supported Linux distributions:

- 5.3.15 and later
- 5.4.2 and later
- 5.5 and later

These Linux distributions include their own Docker Engine package. In addition, older versions of Docker Engine released within the last 12 months are supported unless there is a known compatibility issue with a specific Docker Engine version. In case of a compatibility issue, Cortex XSOAR will publish an advisory notifying customers to upgrade their Docker Engine version.

You can use a version that is not supported. However, when encountering an issue that requires Customer Support involvement, you may be asked to upgrade to a supported version before assistance can be provided.

#### Docker installation by operating system

If you need to install Docker before installing an engine, use the following procedures:

- Red Hat
- Ubuntu
- Amazon Linux
- Oracle Linux

#### NOTE:

For Red Hat's Docker distribution you need Mirantis Container Runtime (formerly Docker Engine - Enterprise) to run specific Docker-dependent integrations and scripts. For more information, see [Install Docker distribution for Red Hat on an engine server](#).

#### Verify Docker user and permissions

##### Verify Docker user

If you installed an engine before installing Docker, verify the demisto operating system user is part of the docker operating system group.

1. Run `id demisto`. For example:

```
id demisto
uid=997(demisto) gid=997(demisto) groups=997(demisto),998(docker)
```

If needed, add the demisto user to the operating system group:

```
sudo groupadd docker
sudo usermod -aG docker demisto
```

Remove these keys from the engine configuration file.

```
python.executable
python.executable.no.docker
```

#### Verify user permissions

To verify that the operating system user (demisto) has necessary permissions and can run Docker containers, run the following command from the OS command line.

```
sudo -u demisto docker run --rm -it demisto/python:1.3-alpine python --version
```

If everything is configured properly you will receive the following output. Python 2.7.14.

#### 4.3.2.1.1 | [Install Docker distribution for Red Hat on an engine server](#)

##### Abstract

Install Docker distribution for Red Hat.

Red Hat maintains its own package of Docker, which is the version used in OpenShift Container Platform environments, and is available in the RHEL Extras repository.

#### NOTE:

If running RHEL v8 or higher, the engine installs Podman packages and configures the operating system to enable Podman in rootless mode.

For more information about the different packages available to install on Red Hat, see the Red Hat Knowledge Base Article (requires a Red Hat subscription to access).

1. Install Red Hat's Docker package.
2. Run the following commands.

```
systemctl enable docker.service
```

```
systemctl restart docker.service
```

3. Change ownership of the Docker daemon socket so members of the **dockerroot** user group have access.

a. Edit or create the file `/etc/docker/daemon.json`.

b. Enable OS group **dockerroot** access to Docker by adding the following entry to the `/etc/docker/daemon.json`: "group": "dockerroot" file.  
For example:

```
{ "group": "dockerroot" }
```

c. Restart the Docker service by running the following command.

```
systemctl restart docker.service
```

d. Install an engine.

e. After the engine is installed, run the following command to add the **demisto** os user to the **dockerroot** os group (Red Hat uses dockerroot group instead of docker).

```
usermod -aG dockerroot demisto
```

f. Restart the engine.

4. Set the required SELinux permissions.

The Cortex XSOAR engine uses the `/var/lib/demisto/temp` directory (with subdirs) to copy files and receive files from running Docker containers. By default, when SELinux is in **enforcing** mode directories under `/var/lib/` it cannot be accessed by Docker containers.

a. To allow containers access to the `/var/lib/demisto/temp` directory, you need to set the correct SELinux policy type, by typing the following command.

```
chcon -Rt svirt_sandbox_file_t /var/lib/demisto/temp
```

b. (Optional) Verify that the directory has the **container\_file\_t** SELinux type attached by running the following command.

```
ls -d -Z /var/lib/demisto/temp
```

c. Configure label confinement to allow Python and PowerShell containers to access other script folders.

In the `d1.conf` file, set the following parameters:

Key	Value
For Python containers	<code>python.pass.extra.keys</code>
	<code>--security-opt=label=level:s0:c100,c200</code>
For PowerShell containers	<code>powershell.pass.extra.keys</code>
	<code>--security-opt=label=level:s0:c100,c200</code>

d. Open any incident and in the incident War Room CLI, run the `/reset_containers` command.

#### 4.3.2.1.2 | Docker image security

##### Abstract

Information about Cortex XSOAR Docker image security practices.

The project that contains the source Dockerfiles used to build the images and the accompanying files is fully open source and available for review. Cortex XSOAR uses the secure Docker Hub registry for its Docker images. However, in an Engine environment, you can also use the PANW registry . You can view the Docker trust information for each image at the image info branch.

## Docker Trust

Signatures for demisto/python3:3.9.1.14969		
SIGNED TAG	DIGEST	SIGNERS
3.9.1.14969	b9d340c3334befc4277f4af8ceae0cbe6d8478c5fb0921d8bf98a73f525330a2	(Repo Admin)
Administrative keys for demisto/python3:3.9.1.14969		
Repository Key:	dd922c0a78cf908782d68ed2f99cae9350740fc2da582a1cb5c03e530ba2dd31	
Root Key:	46ab19438bffff81c925ab27466b52b175de43bad8e8a19cc878e9288246bff8d	

We automatically update our open source Docker images and their accompanying dependencies (OS and Python). Examples of automatic updates can be viewed on GitHub.

We maintain Docker image information which includes information on Python packages, OS packages and image metadata for all our Docker images. Data image information is updated nightly.

All of our images are continuously scanned using Prisma Cloud for known and newly published vulnerabilities, in two scenarios:

- Every new image, and every new version of an image, are scanned before publishing to our public registries, as part of our CI/CD process.
- All existing images are continuously scanned to check whether new vulnerabilities were published and now exist in those images.

We evaluate all critical/high findings and actively work to prevent and mitigate security vulnerabilities.

Cortex XSOAR ensures container images are fully patched and do not contain unnecessary packages. Patches and dependencies are applied automatically via our open source docker files build project.

### Response Prioritization

We remediate any critical and high level vulnerabilities, irrespective of who found them. Issues may be discovered by external researchers, found during internal testing, encountered by customers or reported by other organizations and vendors.

Any vulnerability with a possible exploitation against our images would be responded to with utmost urgency. If we conclude that there is a risk for our customers we will issue an advisory with recommended actions and mitigations. Advisories are published at: <https://security.paloaltonetworks.com/>.

In each version release (every 3 months,) we publish a new version of our content, that will use the latest and secure versions of our images.

### Troubleshooting

- Purge old and unused images periodically.
- If you scanned the Docker images locally, and found some critical CVE's - Make sure you use the latest version of the pack, as it should have the latest version of the image. In addition, purge the old and unused image with vulnerabilities.

### 4.3.2.1.3 | Docker FAQs

#### Abstract

Frequently asked questions (FAQ) about Docker in Cortex XSOAR.

- **Does Cortex XSOAR use COPY or ADD for building images?**

Cortex XSOAR uses COPY for building images. The COPY instruction copies files from the local host machine to the container file system. Cortex XSOAR does not use the ADD instruction, which could potentially retrieve files from remote URLs and perform operations such as unpacking, introducing potential security vulnerabilities.

- **Should the --restart flag be used?**

The --restart flag should not be used. Cortex XSOAR manages the lifecycle of Docker images and restarts images as needed.

- **Can we restrict containers from acquiring additional privileges by setting the no-new-privileges option?**

Cortex XSOAR does not support the no-new-privileges option. Some integrations and scripts may need to change privileges when running as a non-root user (such as Ping).

- **Can we apply a daemon-wide custom seccomp profile?**

The default seccomp profile from Docker is strongly recommended. The default seccomp profile provides protection as well as wide application compatibility. While you can apply a custom seccomp profile, Cortex XSOAR cannot guarantee that it won't block system calls used by an integration or script. If you apply a custom seccomp profile, you need to verify and test the profile with any integrations or scripts you plan to use.

- **Can we use TLS authentication for docker daemon configuration?**

TLS authentication is not used, because Cortex XSOAR does not use Docker remote connections. All communication is done via the local Docker IPC socket.

- **How do we set the logging level to info?**

Set the log level in the Docker daemon configuration file.

- **Can we restrict Linux kernel capabilities within containers?**

The default Docker settings (recommended) include 14 kernel capabilities and exclude 23 kernel capabilities. Refer to Docker's full list of runtime privileges and Linux capabilities.

You can further exclude capabilities via advanced configuration, but will first need to verify that you are not using a script that requires the capability. For example, Ping requires `NET_RAW` capability.

- **Is the Docker health check option implemented at runtime?**

The Cortex XSOAR tenant monitors the health of the containers and restarts/terminates containers as needed. The Docker health check option is not needed.

- **Can we enable live restore?**

Live restore is not used. Cortex XSOAR uses ephemeral Docker containers. Every running container is stateless by design.

- **Can we restrict network traffic between containers?**

Cortex XSOAR does not disable inter-container communication by default, as there are use cases where this might be needed. For example, a script communicating with a long running integration which listens on a port, may require inter-container communication. If inter-container communication is not required, it can be disabled by modifying the Docker daemon configuration.

- **Can we enable user namespace remapping?**

Cortex XSOAR does not support user namespace remapping.

- **How do we configure auditing for Docker files and directories?**

Auditing is an operating system configuration, and can be enabled in the operating system settings. Cortex XSOAR does not change the audit settings of the operating system.

- **Does Cortex XSOAR map privileged ports?**

Cortex XSOAR does not map privileged ports (TCP/IP port numbers below 1024).

- **Does Cortex XSOAR allow privileged execution?**

Cortex XSOAR does not allow privileged execution of Docker commands.

- **Does Cortex XSOAR run SSH within containers?**

Cortex XSOAR does not run SSH within containers.

- **Does Cortex XSOAR change the ownership of the socket?**

Cortex XSOAR does not change the ownership of the socket.

- **Can we disable the userland proxy?**

If the kernel supports hairpin NAT, you can disable docker userland proxy settings by modifying the Docker daemon configuration.

- **Does Cortex XSOAR support the AppArmor profile?**

Cortex XSOAR supports the default AppArmor profile (only relevant for Ubuntu with AppArmor enabled).

- **Does Cortex XSOAR support the SELinux profile?**

Cortex XSOAR supports the default SELinux profile (only relevant for RedHat with SELinux enabled).

- **How does Cortex XSOAR handle secrets management?**

For Docker swarm services, a secret is a blob of data, such as password, SSH private keys, SSL certificates, or other piece of data that should not be transmitted over a network or stored unencrypted in a Docker file or in your application's source code. Cortex XSOAR manages integration credentials internally. It also supports using an external credentials service such as CyberArk.

The following provides troubleshooting solutions for Docker networking and performance issues.

#### Troubleshoot Docker networking issues

In Cortex XSOAR, integrations and scripts run either on the tenant, or on an engine.

If you have Docker networking issues when using an engine, you need to modify the `d1.conf` file.

1. On the machine where the Engine is installed, open the `d1.conf` file.
2. Add the following to the `d1.conf` file:

```
{
    "LogLevel": "info",
    "LogFile": "/var/log/demisto/d1.log",
    "EngineURLs": [
        "wss://1234.demisto.live/d1ws"
    ],
    "BindAddress": ":443",
    "EngineID": "XYZ",
    "ServerPublic": "ABC"
    "ArtifactsFolder": "",
    "TempFolder": "",
    "python.pass.extra.keys": "--network=host"
}
```

3. Save the file.

4. Restart the engine using `systemctl restart d1` or `service d1 restart`.

#### Troubleshoot Docker performance issues

This information is intended to help resolve the following Docker performance issues.

- Containers are getting stuck.
- The Docker process consumes a lot of resources.
- Time synchronization issues between the container and the operating system.

#### Cause

The installed Docker package and its dependencies are not up to date.

#### Workaround

1. Update the package manager cache.

Linux Distribution	Command
Debian	<code>apt-get update</code>

2. (Optional) Check for a newer version of the Docker package.

Linux Distribution	Command
Debian	<code>apt-cache policy docker</code>

3. Update the Docker package.

Linux Distribution	Command
Debian	<code>apt-get update docker</code>

## 4.3.2.1.5 | Configure Docker pull rate limit

**Abstract**

Configure the Docker pull rate limit on public images. Create a Docker user account and receive a higher pull limit.

Docker enforces a pull rate limit on public images. The limit is based on an IP address or as a logged-in Docker hub user. The default limit (100 pulls per 6 hours) is usually high enough for Cortex XSOAR's use of Docker images, but the rate limit may be reached if using a single IP address for a large organization (behind a NAT). If the rate limit is reached, the following error message is issued:

```
Error response from daemon: toomanyrequests: You have reached your pull rate limit. You may increase the limit by authenticating and upgrading: https://www.docker.com/increase-rate-limit.
```

To increase the limit:

1. Sign up a free user in the Docker hub.

The pull limit is higher for a registered user (200 pulls per 6 hours).

2. Authenticate the user on the engine machine by running the following command.

```
sudo -u demisto docker login
```

3. (Optional) Instead of manually logging in to Docker to pull images, you can edit the Docker config file to use credentials from the file or from a credential store.

## 4.3.2.1.6 | Change the Docker installation folder

**Abstract**

Instructions for changing the default Docker folder.

The `/var/lib/docker/` folder is the default Docker folder for Ubuntu, Fedora, and Debian in a standard engine installation.

To change the Docker folder:

1. Stop the Docker daemon.

```
sudo service docker stop
```

2. Create a file called `daemon.json` under the `/etc/docker` directory with the following content:

```
{  
    "data-root": "<path to your Docker folder>"  
}
```

3. Copy the current data directory to the new one.

```
sudo rsync -aP /var/lib/docker/ <path to your Docker folder>
```

4. Rename the old docker directory.

```
sudo mv /var/lib/docker /var/lib/docker.bkp
```

5. After confirming that the change was successful, you can remove the backup file.

```
sudo rm -rf /var/lib/docker.bkp
```

6. Start the Docker daemon.

```
sudo service docker start
```

## 4.3.2.2 | Configure Docker integrations to trust custom certificates

**Abstract**

Configure CA signed and custom certificates for Docker. Trust custom certificates for python integrations in Cortex XSOAR.

Python, Javascript, and native integrations running in Docker use an engine's built-in set of CA-signed certificates to validate TLS communication. If you need to change the certificate bundle of the operating system you are working on, for Javascript and native integrations you need to add custom trusted certificates to the engine built-in set, and for Python Docker integrations you need to create a certificate file that includes the custom certificates and add it to the engine. This is relevant for example if you work with a proxy that performs SSL traffic inspection or use a service that has a self-signed certificate.

#### Configure Javascript and Native integrations to trust custom certificates

1. Add the certificate to the machine's trusted root CA bundle. The location of the CA bundle depends on the operating system version and the operating configuration.

Examples of bundle paths:

- "/etc/ssl/certs/ca-certificates.crt", // Debian/Ubuntu/Gentoo etc.
- "/etc/pki/tls/certs/ca-bundle.crt", // Fedora/RHEL 6
- "/etc/ssl/ca-bundle.pem", // OpenSUSE
- "/etc/pki/tls/cacert.pem", // OpenELEC
- "/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem", //RHEL 7
- "/etc/ssl/cert.pem", // Alpine Linux

Examples of certificate bundle directories:

- "/etc/ssl/certs", // SLES10/SLES11, <https://golang.org/issue/12139>
- "/etc/pki/tls/certs", // Fedora/RHEL

2. Restart the engine.

#### Configure Python Docker integrations to trust custom certificates

This procedure assumes that the Cortex XSOAR `lib dir` is configured to the default location `/var/lib/demisto`.

##### NOTE:

- Only PEM format for certificates is supported.
- `/var/lib/demisto` requires root access. This is relevant for Docker and Podman.

1. Configure the custom certificates for the engine.

- a. Create a certificate PEM file that includes all of the required custom certificates.

- To examine the certificate chain used by a specific endpoint, run the following command on the engine machine (requires openssl client):

```
openssl s_client -servername <host_name> -host <host_name> -port 443 -showcerts < /dev/null
```

For example, `openssl s_client -servername api.github.com -host api.github.com -port 443 -showcerts < /dev/null`

This prints certificate information including the PEM representation of the certificates. After examining the output, if you see **Verification error: unable to get issuer certificate**, one or more certificates in the certificate chain is not available and you need to obtain these certificates from your IT administrator.

- To save the certificates to a `certs.pem` file run the following command:

```
openssl s_client -servername api.github.com -host api.github.com -port 443 -showcerts < /dev/null 2>/dev/null | sed -n '/-----BEGIN CERT/,/-----END CERT/p' > certs.pem
```

- To verify that the `certs.pem` has all needed certificates as part of the certificate chain, run `openssl verify -CAfile certs.pem site.pem`, where `site.pem` contains the certificate of a specific site you want to trust. To get the cert of a site, run `openssl s_client -servername <site_host> -host <site_host> -port 443` and copy the base content including `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.

- After saving the `certs.pem` file, add its content to `/var/lib/demisto/python-ssl-certs.pem`, by running the following command:

```
cat certs.pem >> /var/lib/demisto/python-ssl-certs.pem
```

- b. (RedHat only) Set the required SELinux permissions.

- By default, when SELinux is in enforcing mode, directories under `/var/lib/` cannot be accessed by Docker containers. To allow container access to the `/var/lib/demisto/python-ssl-certs.pem` file, you need to set the correct SELinux policy type, by typing the following command:

```
chcon -t svirt_sandbox_file_t /var/lib/demisto/python-ssl-certs.pem
```

- (Optional) Verify that the file has the `container_file_t` SELinux type attached by running the following command:

```
ls -d -Z /var/lib/demisto/python-ssl-certs.pem
```

c. (Optional) If you require the standard set of certificates trusted by browsers, you can append the CA certificates provided by your operating system. For example, on Ubuntu, these certificates are located at the following path: `/etc/ssl/certs/ca-certificates.crt`. Alternatively, you can download the PEM certificates file provided by the Certifi Project and add your custom certificates to the file that contains the standard set of certificates. For more details, see the `cacert.pem` file.

This example adds the `proxy-ca.pem` file (custom certificate) to the `cacert.pem` file (standard certificates): `cat proxy-ca.pem >> cacert.pem`

d. Copy the certificates PEM file to the following path.

```
/var/lib/demisto/python-ssl-certs.pem
```

(Multi-tenant) In a multi-tenant deployment, the certificate is copied to the following path on the host machine:

```
/var/lib/demisto/tenants/acc_TENANT/python-ssl-certs.pem
```

2. Add the certificate file to your engines.

a. Configure each engine to use the `/var/lib/demisto/python-ssl-certs.pem` file.

i. Verify you have the following directory on the engine host.

```
/var/lib/demisto
```

ii. Set the `demisto` user as the directory owner with **0700** permissions.

iii. Copy the `python-ssl-certs.pem` file to the `/var/lib/demisto` directory.

iv. Add the following configuration to either the engine configuration file (in the UI) or to the `d1.conf` file.

```
"python.docker.use_custom_certs": true
```

b. Restart the engine.

After saving the `python.docker.use_custom_certs` configuration on your engine, Docker images that are launched by the engine will contain the certificates file mounted in the following path:

```
/etc/custom-python-ssl/certs.pem
```

Additionally, the following environment variables will be set with the value of the certificates file path, which enables standard Python HTTP libraries to automatically trust the certificates (without code modifications):

- `REQUESTS_CA_BUNDLE`
- `SSL_CERT_FILE`

The Python SSL library checks the `SSL_CERT_FILE` environment variable only when using OpenSSL. If you use a Docker image that uses LibreSSL, the `SSL_CERT_FILE` environment variable will be ignored. For more details, see LibreSSL support.

#### NOTE:

If you are developing your own integration (BYOI) and using non-standard HTTP libraries, you might need to include specific code that will trust the passed certificates file when the environment variable `SSL_CERT_FILE` is set. In this case, always use the value in the environment variable as the path for the certificates file, and do not hard code the mounted path specified above. For example:

```
certs_file = os.environ.get('SSL_CERT_FILE')
if certs_file:
    # perform custom logic to trust certificates...
```

3. Check the integration runs correctly on your engine.

For more information about troubleshooting, see TLS/SSL troubleshooting.

#### 4.3.2.3 | Docker hardening guide

##### Abstract

Use the Docker Hardening Guide to configure the Cortex XSOAR settings when running Docker containers.

The following describes the engine settings we recommend for securely running Docker containers on Ubuntu, using iptables to restrict IP access.

When editing the configuration file, you can limit container resources, open file descriptors, limit available CPU, and more. For example, add the following keys to the configuration file:

```
{"docker.run.internal.asuser": true, "limit.docker.cpu": true, "limit.docker.memory": true, "python.pass.extra.keys": "--pids-limit=256##--ulimit=nofile=1024:8192"}
```

##### TIP:

We recommend reviewing *Docker network hardening* below, before changing any parameters in the configuration file.

To securely run Docker containers, we recommend to use the latest Docker version.

You can *Check Docker Hardening Configurations* to verify that the Docker container has been hardened according to the settings we recommend.

#### NOTE:

The settings below can also be applied to Podman, with the exception of limiting available memory, limiting available CPU, and limiting PIDS.

#### Docker network hardening

Docker creates its own networking stack that enables containers to communicate with other networking endpoints. By default, Docker uses a networking configuration that allows unrestricted communication for containers so containers can communicate with all IP addresses. You can restrict the networking sources the containers communicate with. The following describes using Ubuntu with iptables commands to restrict IP access.

- **Block network access to the host machine**

Integrations and scripts running within containers do not usually require access to the host network. For added security, you can block network access from containers to services running on the engine machine.

For example, to limit all source IPs from containers that use the IP ranges 172.16.0.0/12, run `sudo iptables -I INPUT -s 172.16.0.0/12 -d 10.18.18.246 -j DROP`. This also ensures that new Docker networks which use addresses in the IP address range of 172.16.0.0/12 are blocked from access to the host private IP. The default IP range used by Docker is 172.16.0.0/12. If you configured a different range in Docker's `daemon.json` config file, use the configured range. Alternatively, you can limit specific interfaces by using the interface name, such as `docker0`, as a source.

1. Add the following iptables rule for each private IP on the tenant machine:

```
sudo iptables -I INPUT -s <IP address range> -d <host private ip address> -j DROP
```

2. (Optional) To view a list of all private IP addresses on the host machine, run `sudo ifconfig -a`

- **Assign a Docker network for a Docker image**

If your engine is installed on a cloud provider such as AWS or GCP, it is a best practice to block containers from accessing the cloud provider's instance metadata service. The metadata service is accessed via IP address **169.254.169.254**. For more information about the metadata service and the data exposed, see the AWS and GCP documentation

There are cases where you might need to provide access to the metadata service. For example, access is required when using an AWS integration that authenticates via the available role from the instance metadata service. You can create a separate Docker network, without the blocked iptable rule, to be used by the AWS integration's Docker container. For most AWS integrations the relevant Docker image is: `demisto/boto3py3`

1. Create a new Docker network by running the following command:

```
sudo docker network create -d bridge -o com.docker.network.bridge.name=docker-metadata aws-metadata
```

2. Edit the engine configuration file either by editing the `d1.conf` file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

3. Add the following key.

```
"python.pass.extra.keys.demisto/boto3py3": "--network=aws-metadata"
```

4. Save the changes.

5. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

6. Verify the configuration of your new Docker network:

```
sudo docker network inspect aws-metadata
```

- **Block internal network access**

In some cases, you might need to block specific integrations from accessing internal network resources and allow the integrations to access only external IP addresses. We recommend this setting for the Rasterize integration when used to Rasterize untrusted URLs or HTML content, such as those obtained via external emails. With internal network access blocked, a rendered page in the Rasterize integration cannot perform a SSRF or DNS rebind attack to access internal network resources.

1. Create a new Docker network by running the following command:

```
sudo docker network create -d bridge -o com.docker.network.bridge.name=docker-external external
```

2. Block network access to the host machine for the new Docker network:

```
iptables -I INPUT -i docker-external -d <host private ip> -j DROP
```

3. Block network access to cloud provider instance metadata:

```
sudo iptables -I DOCKER-USER -i docker-external -d 169.254.169.254/32 -j DROP
```

4. Block internal network access:

```
sudo iptables -I DOCKER-USER -i docker-external -d 10.0.0.0/8 -j DROP
```

```
sudo iptables -I DOCKER-USER -i docker-external -d 172.16.0.0/12 -j DROP
```

```
sudo iptables -I DOCKER-USER -i docker-external -d 192.168.0.0/16 -j DROP
```

5. Edit the engine configuration file either by editing the d1.conf file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

6. Add the following key to run integrations that use the demisto/chromium Docker image with the Docker network external.

```
"python.pass.extra.keys.demisto/chromium": "--network=external"
```

7. Save the changes.

8. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

9. Verify the configuration of your new Docker network:

```
sudo docker network inspect external
```

- **Persist iptables rules**

By default, iptables rules are not persistent after a reboot. To ensure your changes are persistent, save the iptables rules by following the recommended configuration for your Linux operating system:

- Ubuntu
- Red Hat and related operating system flavors

Configure Docker images

You can apply more specific fine tuned settings to Docker images, according to the Docker image name or the Docker image name including the image tag. To apply settings to a Docker image name, add the advanced configuration key to the engine configuration file. If you apply Docker image specific settings, they will be used instead of the general python.pass.extra.keys setting. This overrides the general memory and CPU settings, as needed.

1. Edit the engine configuration file either by editing the d1.conf file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

2. Add the following key to apply settings to a Docker image name.

```
"python.pass.extra.keys.<image_name>"
```

For example, "python.pass.extra.keys.demisto/d1".

- To apply settings to a Docker image name including the image tag, use "python.pass.extra.keys.<image\_name>": "<image\_tag>".

For example, "python.pass.extra.keys.demisto/d1": "1.4".

- To set the Docker images demisto/d1 (all tags) to use a higher max memory value of 2g and to remain with the recommended PIDs and ulimit, add the following to the configuration file:"python.pass.extra.keys.demisto/d1": "--memory=2g##--ulimit=no- file=1024:8192##--pids-limit=256"

3. Save the changes.
4. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

Run Docker with non-root internal users

For additional security isolation, we recommend to run Docker containers as non-root internal users. This follows the principle of least privilege.

1. Edit the engine configuration file either by editing the `d1.conf` file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see [Configure engines](#).
2. Add the following key:

```
"docker.run.internal.asuser": true
```

3. For containers that do not support non-root internal users, add the following key:

```
"docker.run.internal.asuser.ignore" : "A comma separated list of container names. The engine matches the container names according to the prefixes of the key values"
```

For example, `"docker.run.internal.asuser.ignore"="demisto/python3:","demisto/python:"`

The engine matches the key values for the following containers:

```
demisto/python:1.3-alpine
demisto/python:2.7.16.373
demisto/python3:3.7.3.928
demisto/python3:3.7.4.977
```

The `:` character should be used to limit the match to the full name of the container. For example, using the `:` character does not find `demisto/python-ubuntu:2.7.16.373`.

4. Save the changes.
5. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

Configure the memory limit support without swap limit capabilities

When a container exceeds the specified amount of memory, the container starts to swap. Not all Linux distributions have the swap limit support enabled by default.

- Red Hat distributions usually have swap limit support enabled by default.
- Ubuntu distributions usually have swap limit support disabled by default.

To protect the host from a container using too many system resources (either because of a software bug or a DoS attack), limit the resources available for each container. In the engine configuration file, some of these settings are set using the advanced parameter: `python.pass.extra.keys`. This key receives as a parameter full `docker run` options, separated with the `##` string.

How to check if your system supports swap limit capabilities

1. On the engine machine, run the following command:

```
sudo docker run --rm -it --memory=1g demisto/python:1.3-alpine true
```

2. If `swap limit capabilities` is enabled, configure the memory limitation . (To test the memory, see Step 5. Test the memory limit in [Configure the memory limitation](#).)

3. If you see the following message in the output (the message may vary between Docker versions):

```
WARNING: Your kernel does not support swap limit capabilities or the cgroup is not mounted. Memory limited without swap.
```

You have 2 options:

- Configure `swap limit capabilities` by following the Docker documentation.
- See [How to configure the memory limit support without swap limit capabilities](#) below.

If you see the `WARNING: No swap limit support` you can configure memory support without swap limit capabilities.

How to configure the memory limit support without swap limit capabilities

1. Edit the engine configuration file either by editing the d1.conf file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

2. Add the following key to disable swap memory enforcement:

```
"python.pass.extra.keys": "--memory=1g##--memory-swap=-1"
```

If you have the `python.pass.extra.keys` already set up with a value, add the value after the ## separator.

3. Save the changes.

4. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

Configure the memory limitation

We recommend to limit available memory for each container to 1 GB.

If `swap limit capabilities` is enabled (see [How to check if your system supports swap limit capabilities](#) above), in Cortex XSOAR configure the memory limitation using the following advanced parameters.

1. Edit the engine configuration file either by editing the d1.conf file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

2. Add the following keys.

```
"limit.docker.memory": true, "docker.memory.limit": "1g"
```

3. Save the changes.

4. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

5. Test the memory limit.

1. Go to Scripts and click New Script.

2. In the Script Name file, type `TestMemory`.

3. Add the following script:

```
from multiprocessing import Process
import os

def big_string(size):
    sys.stdin = os.fdopen(0, "r")
    s = 'a' * 1024
    while len(s) < size:
        s = s * 2
    print('completed creating string of length: {}'.format(len(s)))

size = 1 * 1024 * 1024 * 1024
p = Process(target=big_string, args=(size, ))
p.start()
p.join()
if p.exitcode != 0:
    return_error("Return code from sub process indicates failure: {}".format(p.exitcode))
else:
    print("Success allocating memory of size: {}".format(size))
```

4. In the SCRIPT SETTINGS section, select the script to run on the Single engine and select the engine where you want to run the script.

5. Save the script.

6. To test the memory limit, type `!TestMemory`. The command returns an error when it fails to allocate 1 GB of memory.

Configure the CPU, PIDs, and open file descriptors limit

Set the advanced parameters to configure the CPU limit, PIDs limit and the open file descriptor limit.

1. Edit the engine configuration file either by editing the d1.conf file, or If you installed via Shell, you can edit the configuration in the UI as well as editing the file directly. For details, see Configure engines.

2. Add the following keys:

Parameter	Key
Available CPU limit	"limit.docker.cpu": true, "docker.cpu.limit": "<CPU Limit>" We recommend to limit each container to 1 CPU. (For example, 1.0. Default is 1.0).
PIDs limit	"python.pass.extra.keys": "--pids-limit=256"
Open file descriptors limit	"python.pass.extra.keys": "--ulimit=nofile=1024:8192"

3. Save the changes.
4. Restart the demisto service on the engine machine.

```
sudo systemctl start d1
```

```
(Ubuntu) sudo service d1 restart
```

Check Docker hardening configurations

Check your Docker hardening configurations on an engine by running the !DockerHardeningCheck command in the CLI. The results show the following:

- Non-root User
- Memory
- File descriptors
- CPUs
- PIDs

Before running the command, ensure that your engine is up and running.

1. Update the DockerHardeningCheck script to run on the engine.

**NOTE:**

By default, the DockerHardeningCheck script runs on the Cortex XSOAR tenant.

- a. Go to Incident Response → Automation → Scripts → DockerHardeningCheck → Settings.
- b. In the Run on field select Single engine and from the list, select the engine you want to run the script.
- c. Save the script.

2. Verify the Docker container has been hardened according to recommended settings, in the CLI, run the !DockerHardeningCheck command.

#### 4.3.3 | Podman

Abstract

Run Podman containers instead of Docker for RHEL v8.

Podman is a daemonless container engine for developing, managing, and running OCI Containers on the Linux System. Containers can either be run as root or in rootless mode.

If you use the Shell installer to install an engine, Cortex XSOAR automatically detects the container management type based on the operating system. For example, if your operating system is running RHEL v8 and higher, Cortex XSOAR installs Podman packages and configures the operating system to enable Podman in rootless mode.

**NOTE:**

When upgrading an engine, the engine keeps the previously used container management type (regardless of distribution version).

If using PowerShell integrations, you may need to configure the default SELinux policy as Podman can affect processes which mmap to /dev/zero.

Docker hardening guidelines

Docker hardening guidelines can be applied to Podman, with the exception of Limit Available Memory, Limit Available CPU, and Limit PIDS.

#### 4.3.3.1 | Change container storage directory

By default, Podman uses the `$HOME/.local/share/containers/storage` directory. To use a different directory for container storage, edit the Podman config file located at `/home/demisto/.config/containers/storage.conf`. If the Podman config file does not exist, you need to create it and change the ownership.

The new storage directory needs to be owned by the `demisto` user, otherwise they will be denied access to it.

Do not use NAS storage for the `$HOME` directory. The directory needs to be a local directory for Podman to work.

##### TIP:

We recommend reserving 150 GB for container storage, either in the `/home` partition or a different storage directory that you have set using the `rootless_storage_path` key.

1. If the Podman config file does not exist:

- a. Create the Podman config file.

```
cp /etc/containers/storage.conf /home/demisto/.config/containers
```

- b. Change the ownership of the Podman config file.

```
chown demisto:demisto /home/demisto/.config/containers/storage.conf
```

2. To set a different directory for container storage, change the key: `rootless_storage_path` in the `storage.conf` file. For example, `rootless_storage_path=/var/lib/containers/$USER/storage`

3. To assign the `demisto` user ownership of the new storage directory, on the Linux command line, run `chown -R demisto:demisto <NEW-LOCATION>`.

#### 4.3.3.2 | Install Podman

##### Abstract

Install Podman on engines for RHEL v8 or later.

When installing a new engine on RHEL 8 or later, the shell installer configures Podman automatically. There are some cases, however, where you might need to install Podman manually:

- When using an installation method other than the shell installer (e.g. an RPM package) on RHEL 8 or later.
- When the shell installer did not successfully install Podman.
- When you want to migrate from Docker to Podman, for an existing Cortex XSOAR engine.

##### NOTE:

- This procedure is intended for RHEL 8 or later. It may not work for other operating system types.
- Do not use NAS storage for the `$HOME` directory. The directory needs to be a local directory for Podman to work.

1. For RHEL 8, install Podman by typing the following commands:

- `sudo yum -y install slirp4netns fuse-overlayfs`
- `sudo yum -y module install container-tools`

For RHEL 9 or later, install Podman by typing the following command:

- `sudo yum -y install slirp4netns fuse-overlayfs podman`

2. Run the following commands:

- `sudo touch /etc/subuid /etc/subgid`
- `sudo mkdir -p /home/demisto`
- `sudo chown demisto:demisto /home/demisto`

3. Configure the `unqualified-search-registries` used by Podman.

Podman by default uses the `fedoraproject.org`, `redhat.com`, and `docker.io` unqualified search registries. Since Cortex XSOAR images use only the `docker.io` registry, you can speed up download times for container images by setting `unqualified-search-registries` to just `docker.io`.

- a. Create or edit the `/home/demisto/.config/containers/registries.conf` config file.

- b. In the file, set `unqualified-search-registries = ["docker.io"]`

**NOTE:**

If you edit the file with the `root` user, make sure to set the `demisto` user as file owner by running `chown demisto:demisto /home/demisto/.config/containers/registries.conf`

4. Change the `subuids` and `subgids` by running the following command:

```
sudo usermod --add-subuids 200000-265535 --add-subgids 200000-265535 demisto
```

5. Migrate existing containers to Podman:

```
sudo sh -c "cd /; runuser -u demisto -- podman system migrate"
```

6. Set the `net.ipv4.ping_group_range`, by typing the following commands:

- `sudo sh -c "echo 'net.ipv4.ping_group_range=0 2000000' > /etc/sysctl.d/demisto-ping.conf"`
- `sudo sysctl -w "net.ipv4.ping_group_range=0 2000000"`

7. As root user, edit the following `config` file:

```
/usr/local/demisto/d1.conf
```

8. Change the "`container.engine.type`": "docker" to "podman".

If this line does not exist, add the following line to the file:

```
"container.engine.type": "podman"
```

```
"Server": {
    "HttpsPort": "443",
    "ProxyMode": true
},
"container": {
    "engine": {
        "type": "podman"
    }
},
"db": {
    "index": {
        "entry": {
            "disable": true
        }
    }
}
```

9. If the engine is running, restart the service.

```
sudo systemctl restart d1
```

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

**4.3.3.3 | Migrate From Docker to Podman****Abstract**

Switch from Docker to Podman when installing an engine for RHEL 8 or later.

Although Podman is set up automatically in an engine installation, it is possible to migrate from Docker to Podman in an existing engine. Follow the Podman installation instructions to migrate.

**4.3.3.4 | Troubleshoot Podman****Abstract**

Troubleshoot process leak or installation issues for Podman.

**dbus-daemon process leak**

Podman version 3.4.1 and lower has a known issue that dbus-daemon processes may leak when running in an environment containing the dbus-x11 OS package. The issue occurs when the dbus-x11 OS package is installed, for example when installing an X11 desktop environment like GNOME desktop on the host machine. If you experience this issue, you see a large number of dbus-daemon processes owned by the `demisto` OS user. To check if you are affected by the issue, run the following command:

```
ps -fe | grep demisto | grep dbus-daemon
```

To fix this issue:

1. Remove the dbus-x11 OS package and dependent packages by running the following command:

```
sudo yum remove dbus-x11
```

2. After removal you can kill the leaked dbus-daemon processes by running the following OS command:

```
pgrep -u demisto dbus-daemon | xargs sudo kill
```

Invalid argument error

When Podman fails to run with an “Invalid argument” error, such as:

```
ERROR[0000] running `/usr/bin/newuidmap 15936 0 1029 1 1 165536 65536 65537 200000 65536` : newuidmap: write to uid_map failed: Invalid argument
Error: cannot set up namespace using "/usr/bin/newuidmap": exit status 1
```

This can be caused by duplicate lines for Cortex XSOAR in /etc/subuid and /etc/subgid.

To fix this issue:

1. Check if the /etc/subuid file contains multiple lines that start with the Cortex XSOAR username (usually demisto). For example:

```
alice:10000:65536
demisto:165536:65536
demisto:20000:65536
splunk:331072:65536
```

2. If this is the case, edit the file as root, and remove the extra line(s) for Cortex XSOAR. The line you should keep is the one that ends with 200000:65536. Continuing with the above example, here is the end result:

```
alice:10000:65536
demisto:20000:65536
splunk:331072:65536
```

3. Repeat the above steps for the /etc/subgid file.

Verify Podman installation

When encountering errors in Cortex XSOAR that are Podman related, such as:

- failed to run "docker ps". stderr: [], err: [Timeout. Process killed (1400)]
- Timeout while waiting for pong response [error 'Read timed out (15s)']
- Error: error joining network namespace of container 06b8aec6eabe2e735128e3a72cb06c8ae2d97ade60a56ab555034442ea4e2a84: error retrieving network namespace at /tmp/podman-run-989/netns/cni-86dca01c-bd84-1aaf-85fb-72b659a8e42a: unknown FS magic on "/tmp/podman-run-993/netns/cni-86dca01c-bd84-1aaf-85fb-72b659a8e42a": 58465342

Verify that Podman is running properly and consider whether to delete Podman data directories.

1. Verify that Podman is running properly with the **demisto** OS user, by performing the following steps:

- Change the OS user to **demisto** by running the following command:

```
sudo su - -s /bin/bash demisto
```

- Check that your system complies with the minimum requirements, and view general system information such as host architecture, CPU, OS, registries, container storage path, etc., by running the following command:

```
podman info
```

- Check all active running containers, container names and IDs, by running the following command:

```
podman ps
```

- Check that Podman is able to run a container, by running the following command:

```
podman run --rm -t demisto/python3:3.10.4.29342 echo "podman is working"
```

If any of the Podman commands are not working, try running with the **--log-level=debug** to receive additional details as to why it is failing. For example:

```
podman -log-level=debug ps
```

```
podman --log-level=debug ps podman --log-level=debug run --rm -t demisto/python3:3.10.4.29342 echo "podman is working"
```

2. Reset the Podman Data Directories.

If the Podman commands in step 1 are failing, you should clean the Podman working directories. Sometimes Podman's data directories get corrupted (for example, as a result of insufficient disk space).

**NOTE:**

This step removes all Podman images including any custom images you may have created.

- Stop the engine by running the following command:

```
sudo systemctl stop d1
```

- Ensure that all Podman containers of the **demisto** user are stopped, by running the following command:

```
ps -fe | grep demisto | grep 'podman run'
```

If required, kill the running containers.

- Delete the following directories (assuming the **demisto** OS user's home directory is at: /home/demisto)

- sudo rm -rf /home/demisto/.cache/containers/
- sudo rm -rf /home/demisto/.local/share/containers/
- sudo rm -rf /tmp/podman-run-\$(id -u demisto)
- sudo rm -rf /tmp/containers-user-\$(id -u demisto)
- sudo rm -rf /tmp/tmp/run-\$(id -u demisto)

**NOTE:**

`$(id -u demisto)` is used to get the **demisto** user ID, which is part of the directory name. For example, `/tmp/podman-run-993`

Not all the directories above may be present.

- Start the engine by running the following command:

```
sudo systemctl start d1
```

- Verify that Podman is working properly with the **demisto** OS user by following step 1.

#### Unused containers taking up resources

In some cases, if the Podman process crashes or is killed abruptly it can leave containers on disk. You might see errors such as `error allocating lock for new container: allocation failed; exceeded num_lock` when the maximum number of locks used to manage containers is exhausted due to the unused containers that remain.

1. Change to the demisto operating system user `sudo su - -s /bin/bash demisto`.
2. Run `podman ps -a -f status=exited` to check for unused containers.
3. Clean up the unused containers `podman container cleanup --rm -a`.

**NOTE:**

When you run `podman container cleanup --rm -a`, you might see a message such as `running or paused containers cannot be moved without force`. The message can be safely ignored, as it only pertains to current running containers, which are not removed.

4. After cleanup, verify there are no remaining unused containers `podman ps -a -f status=exited`.

#### Keyring quota exceeded error

```
Script failed to run: Docker code runner got container error: [Docker code script is in inconsistent state, ... error: [exit status 126] stderr: [Error: OCI runtime error: crun: create keyring ...: Disk quota exceeded]
```

By default, Podman creates a keyring that is used by each container. The limit per user on the machine might be low and Podman can reach the limit when running more containers than the keyring limit. To check the keyring usage, run the `sudo cat /proc/key-users` operating system command.

The command returns the usage for each UID (to retrieve the demisto user UID, run `id demisto`). The fourth column shows the number of keys used out of the total number available. For more information about keys, see Kernel Key Retention Service.

You can either increase the limit of max keyrings (increasing to 1000 is safe and reasonable) per user as specified by your Linux vendor documentation or you can disable keyring creation by Podman. We recommend disabling keyring creation, unless keyrings are used by Podman in other applications on the machine. To disable keyring creation by Podman, modify the `containers.conf` file and add the option `keyring = false` under the "[containers]" section. For more information, see the Containers Engine Configuration File.

#### Report a support case for installation issues

If the procedure set out in the Verify Podman installation section above does not solve the Podman issue and you require assistance from Support, do the following:

1. Include the following files as part of the support case:

- `/etc/containers/storage.conf`
- `/home/demisto/.config/containers/storage.conf`  
If the file does not exist, indicate that there is no such file.
- `/home/demisto/.config/containers/registries.conf`  
If the file does not exist, indicate that there is no such file.

2. Include the output of the following commands as the `demisto` user.

**NOTE:**

To change to the `demisto` OS user, run the following command:

```
sudo su - -s /bin/bash demisto
```

- `podman info`
- `podman images`
- `podman --log-level=debug ps`
- `podman --log-level=debug run --rm -t demisto/python3:3.10.4.29342 echo "podman is working"`

Permission issues with directories under `/run` path

When installing a Cortex XSOAR engine on a RHEL system (version 8 or later), or when running an integration on such an engine, you get a permission error for a path under `/run` (for example `/run/user/0` or `/run/libpod`).

1. In RHEL 9 only: Make sure the `container-tools` meta-package is installed, by running:

```
yum -y install container-tools
```

2. Run the following commands:

```
cp /etc/containers/storage.conf /home/demisto/.config/containers/storage.conf
chown demisto:demisto /home/demisto/.config/containers/storage.conf
chmod 600 /home/demisto/.config/containers/storage.conf
```

3. Edit `/home/demisto/.config/containers/storage.conf`.

- Under [storage], change `runroot` to some temporary directory that is accessible by user `demisto`.  
For example: `runroot = "/tmp/podman-run-xsoar"`
- Also under [storage], change `graphroot` (which is where container images are stored) to any location that is owned and accessible by user `demisto`. We recommend using this standard path:

```
graphroot = "/home/demisto/.local/share/containers/storage"
```

- Under [storage.options.overlay], uncomment the following line (remove the # from the start):

```
mount_program = "/usr/bin/fuse-overlayfs"
```

4. Save the file and run the following.

**NOTE:**

You must switch to user `demisto` before running the "system migrate" (running it as root will have no effect).

```
su - demisto
```

```
podman system migrate
```

5. Also as user `demisto`, run the following to ensure the path changes were applied:

```
podman info | grep Root
```

You should see the correct `runRoot` and `graphRoot` settings.

6. Still as user `demisto`, verify the issue is resolved by running:

```
podman run hello-world
```

7. If the issue persists, purge Podman's database by running the following:

**NOTE:**

The "system migrate" must be done by user demisto.

```
rm -rf /home/demisto/.local/share/containers/*
```

```
podman system migrate
```

## 4.4 | Manage engines

### Abstract

Manage engines and load-balancing groups.

You can manage your engines and load-balancing groups by going to Settings & Info → Settings → Integrations → Engines.

You can view engine names, hosts, status, connection, and other engine information.

#### NOTE:

In the Name column, if the service name starts with a d1 prefix, it is a multiple engine.

You can do the following:

Option	Description
Load-Balancing Group	<p>Separate load-balancing groups have several uses. For example:</p> <ul style="list-style-type: none"> <li>• Use separate load-balancing groups for different integrations and instances. Create Load-Balancing groups for certain tasks, which can help segregate the infrastructure of critical integrations.</li> <li>• Managed Security Service Providers may want to split internal engines and SaaS product engines.</li> <li>• If you have multiple AWS accounts that are not connected and do not want a single point of failure for AWS integrations that use STS.</li> </ul> <p>You can do the following:</p> <ul style="list-style-type: none"> <li>• Add/remove engines to a load-balancing group</li> </ul> <p>You can only add the engine to the load-balancing group after you have connected the engine.</p> <p>If you want to remove the last engine from a specific load-balancing group and one or more integration instances use that engine, you will get an error. Before moving the engine, in the integration instance settings, you need to update the Run on field to a different engine or no engine.</p> <ul style="list-style-type: none"> <li>• Create load-balancing groups</li> </ul> <p>When selecting Load-Balancing Group → Add to new group, you can create multiple load-balancing groups and decide which engines are part of each group.</p> <p>Users can move an engine from one group to another. A group will be deleted when the last engine is removed from it.</p> <p>Each engine can only belong to one group.</p>
Upgrade Engine	Relevant for Shell installation only. If you didn't install an engine using the Shell installation you will need to remove the engine and do a fresh installation. For more information, see
Get Logs	Logs are located in <code>/var/log/demisto</code> . For multiple engines, logs are located in <code>/var/log/demisto/&lt;name of the engine&gt;</code> . For example, <code>var/log/demisto.d1_e1</code> .
Edit Configuration	Relevant for Shell installation only. Enables you to edit the <code>d1.conf</code> file without having to access the file on your remote machine. For more information, see Configure engines.
Download Configuration	Download the <code>d1.conf</code> file to view the attribute values. Useful when migrating from Cortex XSOAR 6 to Cortex XSOAR 8.

Option	Description
Delete Engine	Deletes an engine from Cortex XSOAR. To remove the engine from your remote machine, see Remove an engine.

## 4.5 | Upgrade an engine

### Abstract

Upgrade an engine on Cortex XSOAR or directly on the remote machine.

Whenever there is a Cortex XSOAR major version change or a change in tenant-engine protocol version, your engines require an upgrade. On the Engines page, the Status column shows those engines that require upgrades. You can upgrade an engine by doing the following:

- If you installed the engine using the Shell installer, you can upgrade the engine on the Engines page.
- If you didn't install the engine using the Shell installer, you need to remove the engine and do a fresh install.

### Upgrade an engine (Shell installations)

You can upgrade the engine on the Engines page if you have installed the engine using the Shell installer.

#### NOTE:

The Engine needs to be connected.

1. On the Engines page, select the checkbox for the engine that requires an upgrade.
2. Click Upgrade Engine.

When the upgrade finishes, the version appears in the Cortex XSOAR Version column. The upgrade procedure can take several minutes.

### Upgrade an engine (non-shell installations)

If you didn't use the Shell installer, you need to remove the engine and do a fresh install.

1. On the Engines page, locate the engine that requires an update.
2. In the Download link, click relevant Download files.
3. On the remote machine, do the following:
  - Remove the existing engine. For more information, see Remove an engine.
  - Install the engine you downloaded in step 2. For more information, see Install an engine.

When the upgrade finishes, the version appears in the Cortex XSOAR Version column. The upgrade procedure can take several minutes.

For troubleshooting, see Troubleshoot engine upgrades.

#### NOTE:

By default, auto-upgrade extracts the files to the `/tmp` directory. In some cases, you might need to use a different directory. For example, a common use case is if your `/tmp` directory is mounted as a non-executable directory. To use a different directory, edit the `XSOAR_ENGINE_AUTO_UPGRADE_TMP_DIR` env variable. The env variable can be specified as a global variable or can be edited in the crontab of the root user that runs the engine upgrade script. To edit the crontab of root, run `sudo crontab -e`. For example:

```
# d1 engine
XSOAR_ENGINE_AUTO_UPGRADE_TMP_DIR=/root/tmp
PATH=/sbin:/bin:/usr/sbin:/usr/bin
* * * * * /usr/local/demistoupgrade_engine.sh >> /var/log/demisto/demisto_install.log
```

## 4.6 | Remove an engine

### Abstract

Remove an engine by running the relevant command, depending on your operating system.

You can remove an engine when it is no longer needed.

- Run one of the following commands according to your operating system:

Installation	Command
RPM	Get the full package: <code>rpm -qa   grep -i ^d1_*</code> Remove the package: <code>rpm -evv d1_ &lt;package name&gt;</code>
DEB	Get the full package: <code>dpkg-query -W -f='\${Package}' d1_*</code> Remove the package: <code>dpkg --purge &lt;package name&gt;</code>
SH	Remove an Engine: <code>sudo &lt;engine-file-path&gt; -- -purge</code>

## 4.7 | Configure engines

### Abstract

Configure Cortex XSOAR engines to change the number of workers, access communication tasks, notify users if engine disconnects, and remove server from group.

When installing an engine, a d1.conf file is installed on your machine. Some configurations can only be done by editing the d1.conf file. If you install via Shell, you can edit the configuration in the UI as well as editing the file directly.

A use case for modifying the engine configuration is if you want to generate engine logs for a specific log level.

### Edit the d1.conf file

- On the machine on which you installed the engine, navigate to the d1.conf file:

Installation Type	Location
RPM, DEB, Shell	<code>/usr/local/demisto</code> If using multiple engines, the location is <code>/usr/local/demisto/name of the engine</code> . For example, <code>/usr/local/demisto/d1_e1</code>
ZIP	Same folder as the binary.

- Modify the file as required. See Common properties when editing an engine configuration

You can also Configure the engine to use a web proxy.

### Modify the configuration in Cortex XSOAR (Shell installations only)

Ensure that the data is in JSON format. The properties that you specify override the values defined in the d1.conf file.

- From the engines table, select the engine for which you want to modify the configuration.
- Click Edit Configuration.
- In the JSON formatted configuration dialog box, modify the properties as required. For more information, see Common properties when editing an engine configuration.



#### Common properties when editing an engine configuration

The following table describes the common properties when editing an engine configuration using the d1.conf file (located by default at /usr/local/demisto/) or in the JSON formatted configuration dialog box in Cortex XSOAR.

Property	Type	Values	Edit
http_proxy	String	The IP address of the HTTP proxy through which the engine communicates.  For an example, see <a href="#">Configure the engine to use a web proxy</a> .	The engine d1.conf file.
https_proxy	String	The IP address of the HTTP/s proxy through which the engine communicates.  For an example, see <a href="#">Configure the engine to use a web proxy</a> .	The engine d1.conf file.
LogLevel	String	<ul style="list-style-type: none"> <li>• debug</li> <li>• info</li> <li>• warning</li> </ul>	The engine d1.conf file or in the JSON formatted configuration dialog box.
BindAddress	String	The port on which the engine listens for agent connection requests and communication task responses.	The engine d1.conf file.
EngineURLs	String array	An array of tenant addresses to which the engine tries to connect. If you change the tenant URL, you need to update this parameter.	The engine d1.conf file.
LogFile	String	Path to the d1.log file. If you change the name or location of the d1.log file, you need to update this parameter.	The engine d1.conf file.
engine.allow.data.collection	String	Disables the option to send communication task forms through the engine. <ul style="list-style-type: none"> <li>• false</li> </ul>	The engine d1.conf file.

#### 4.7.1 | Configure the engine to use a web proxy

Abstract

Configure a Cortex XSOAR engine to use a web proxy by editing the d1.conf file.

Proxy settings can be configured in an engine by adding them as an engine configuration.

#### **NOTE:**

You need to configure Docker to use a proxy. When using a BlueCoat proxy, ensure you encode the values correctly.

1. On the machine on which you installed the engine, navigate to the d1.conf file and add the following keys.

Key	Value	Description
http_proxy	http://<user:password@proxy-server:port#> For example http://user:password@proxy-server:3128	Environment uses http proxy. Special characters must be escaped.
https_proxy	https://<user:password@proxy-server:port#> For example, https://user:password@proxy-server:3128	Environment uses https proxy. Special characters must be escaped.

2. If the environment variables are not set, or you wish to use a different settings than those specified in the environment variables, set the configuration with your specific proxy details in the d1.conf file. For example:

```
{"http_proxy": "http://proxy.host.local:8080",
"https_proxy": "https://proxy.host.local:8443"}
```

3. Save the file.

#### 4.7.2 | Configure the engine to call the server without using a proxy

##### Abstract

Configure an engine to call the server without using a proxy.

In some cases, due to specific environment architecture, you may need to configure the engine to use a proxy when working with integrations, but not use a proxy when calling the Cortex XSOAR tenant.

1. On the computer where you have installed the engine, go to the directory for d1.conf file.

For RPM, DEB, Shell go to /usr/local/demisto.

2. Add the following configuration:

Key	Value
engine.to.server.proxy	false (default is true)

#### 4.7.2.1 | Use NGINX as a reverse proxy

##### Abstract

Use NGINX as a reverse proxy to the Cortex XSOAR engines.

NGINX can act as a reverse proxy that sits between internal applications and external clients, forwarding client requests to the appropriate application. Using NGINX as a reverse proxy in front of the engine enables you to provide network segmentation where the proxy can be put on a public subnet (DMZ) while the engine can be on a private subnet, only accepting traffic from the proxy. Additionally, NGINX provides a number of advanced load balancing and acceleration features that you can utilize.

If you want to use an engine (d1) through the reverse proxy, you need to modify EngineURLs in the d1.conf file to point to the host and port the NGINX server is listening on.

##### Install NGINX

You can install NGINX on the Red Hat/Amazon (yum) and Ubuntu Linux distributions. For full instructions and available distributions, see NGINX documentation.

1. Run one of the following commands according to your Linux system:

- RedHat/Amazon: `sudo yum install nginx`
- Ubuntu: `sudo apt-get install nginx`

2. (Optional) Verify the NGINX installation by running the following command:

```
sudo nginx -v
```

Generate a certificate for NGINX

You should not use self-signed certificates for production systems. It is recommended to use a properly signed certificate for production systems. These instructions are intended only for non-production setups.

1. To use OpenSSL to generate a self-signed certificate, on the engine machine run the following command:

```
sudo openssl req -x509 -nodes 3650 -newkey rsa:2048 -keyout /etc/nginx/cert.key -out /etc/nginx/cert.crt
```

2. When prompted, complete the on-screen instructions to complete the required fields.

Configure NGINX

1. Open the following NGINX configuration file with your preferred editor:

```
/etc/nginx/conf.d/demisto.conf
```

2. Use the following configuration template:

Replace **DEMISTO\_ENGINE** with the appropriate hostname.

```
# Replace DEMISTO_ENGINE with the appropriate hostname. If needed, change port 443 to the port on which the engine is listening.
```

```
upstream demisto {
    server DEMISTO_ENGINE:443;
}

# Uncomment to redirect http to https (optional)
# server {
#     listen 80;
#     return 301 https://$host$request_uri;
# }

server {
    # Change the port if you want NGINX to listen on a different port
    listen 443;

    ssl_certificate      /etc/nginx/cert.crt;
    ssl_certificate_key  /etc/nginx/cert.key;

    ssl on;
    ssl_session_cache builtin:1000 shared:SSL:10m;
    ssl_protocols TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    access_log  /var/log/nginx/demisto.access.log;

    location / {

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_pass      https://demisto;
        proxy_read_timeout 90;
    }

    location ~ ^/(websocket|d1ws|d2ws) {
        proxy_pass https://demisto;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header Origin "";
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

3. Restart the NGINX server, by typing the following command:

```
sudo service nginx restart
```

4. Verify you can access the engine by browsing to the NGINX server host.

#### 4.7.2.2 | Configure an engine to use custom certificates

##### Abstract

Replace the self-signed certificate for an engine with a valid CA certificate for communication tasks.

For communication tasks that go through an engine, you can replace the default self-signed certificate for the engine with your own certificate.

1. Find the two files created by the engine. The default location is `/usr/local/demisto`.

```
d1.key.pem
```

```
d1.cert.pem
```

2. Replace the contents of these files with your own certificates.

3. Change file owner to demisto:

```
chown -R demisto:demisto d1.key.pem
```

```
chown -R demisto:demisto d1.cert.pem
```

4. Set the file permissions:

```
chmod 600 d1.key.pem
```

```
chmod 644 d1.cert.pem
```

## 4.8 | Use an engine in an integration

##### Abstract

Use an engine or load-balancing group of engines to fetch alerts and run commands for an integration.

When you create an integration instance, you can select whether to fetch alerts and run commands executed for the integration using the engine or a load-balancing group of engines. After you add the engine or load-balancing group to an integration instance, you can run commands using the engine or load-balancing group by specifying the `using` argument in the alert War Room.

Before configuring an integration to run using multiple engines in a load-balancing group, we recommend that you test the integration using a single engine in the load-balancing group.

##### Command Example

```
!url url="www.cnn.com" using=urlscan.io_instance_1
```

## 4.9 | Run a script using an engine

##### Abstract

Run a script on an engine or load-balancing group to distribute the workload and improve performance.

You can run a script on an engine or load-balancing group to distribute the workload and improve performance.

1. On the Scripts page, select the script and click Settings.
2. From the BASIC section, in the Run on field, select either a single engine or a load-balancing group.  
The option to select an engine or load-balancing group only appears if at least one engine or load-balancing group is connected.
3. From the list, select the name of the engine or load-balancing group.
4. Click Save.

## 4.10 | Troubleshoot engines

##### Abstract

Troubleshoot engines by accessing logs and viewing errors.

When troubleshooting engines, access the logs from Settings & Info → Settings → Integrations → Engines and select the engine from which you want to download the logs.

**NOTE:**

Ensure that pop-ups are not blocked by your browser.

**Debug engines**

The d1.log field appears whenever an engine is running. The d1.log field contains information necessary for your customer success team to debug any engine related issue. The field displays any error, as well as noting whether the engine is connected.

```

1119 Sep 13 15:25 d1.cert.pem
685 Nov 29 1979 d1.conf.pem
1679 Sep 13 15:25 d1.key.pem
477 Sep 13 15:25 d1.log
60623824 Sep 12 18:35 d1.darwin_amd64
61154501 Sep 12 18:35 d1.linux_amd64
61245448 Sep 12 18:35 d1.windows_amd64.exe

```

Troubleshoot engine installation

**NOTE:**

If the installer fails to start due to a permissions issue, even if running as root, add one of the following two arguments when running the installer:

- `--target <path>` - Extracts the installer files into the specified custom path.
- `--keep` - Extracts the installer files into the current working directory (without cleaning at the end).

If using installer options such as `--tools=false`, the option should come after the `--target` or `--keep` arguments. For example:

```
sudo ./d1-installer.sh --target /some/temp/dir --tools=false
```

After installing the engine, check that the engine is connected to the Cortex XSOAR tenant and that it is running.

1. Go to Settings & Info → Settings → Integrations → Engines and verify that the engine is connected.

2. If the engine is not connected, run the following command on the engine server to check if the engine service is running.

```
sudo systemctl status d1
```

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl status d1_<Engine_name>
```

3. Access the d1 log on the engine server.

```
sudo tail -f /var/log/demisto/d1.log
```

- If the engine service is not running, and there's nothing relevant in the log, run `journalctl` on the engine server to understand why the installation failed.
- If the engine service is running, review the errors to see if the engine is failing to connect or if there are other issues (ignore all errors related to `\d2ws`, because this is not the same as `d1ws`.) Most often, the server address is incorrect and you will see an error like this:

```
error Cannot connect to [wss://<mainServerIP/HostName>/d1ws]: wss://<mainServerIP/HostName>/d1ws: dial tcp: lookup localhost: no such host. . Waiting 3 seconds. Will try until...
```

In this case, navigate to `/usr/local/demisto/d1.conf` and change the `EngineURLs` parameter to an address the engine can reach. Check the addresses at the beginning of the `upgrade_engine.sh` file and update them to be the same as in the conf file. The addresses should be a comma-separated list.

**NOTE:**

You can ignore the following error: `Cannot create folder '/var/lib/demisto'`

The configurations that might affect the `upgrade_engine.sh` script are the following variables located at the beginning of the script:

- `SERVER_URLS`
- `TRUST_ANY_CERT`

If you make a change to the `baseURLs` configuration, you must apply the change in `/usr/local/demisto/d1.conf` AND in `/usr/local/demisto/upgrade_engine.sh` under the `SERVER_URLS` var.

If you make a change in the `engine.connection.trust_any_certificate` configuration, you must apply the change in `/usr/local/demisto/upgrade_engine.sh` as follows:

- If the `engine.connection.trust_any_certificate` configuration was set to true (trust any certificate), set the `TRUST_ANY_CERT` variable to `-k`.
- If the `engine.connection.trust_any_certificate` configuration was set to false, the `TRUST_ANY_CERT` variable should be blank ("").

4. To check the connectivity from the engine to the Cortex XSOAR tenant, see *Troubleshoot engine connectivity* below.

5. If the installation issue remains, open a support case with logs from the engine.

a. On the engine server, in /usr/local/demisto/d1.conf, set "LogLevel": "debug".

b. Restart the d1 service and let it run for a few minutes.

```
sudo systemctl restart d1
```

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

c. Capture a journalctl:

```
journalctl --since "1 day ago" > engineTroubleshootingJournalctl.log
```

d. On the engine server, tar up the log, conf, journalctl, and install log on the engine.

```
tar -cvzf engineLogs.tar.gz /var/log/demisto /usr/local/demisto/d1.conf /tmp/demisto_install.log  
engineTroubleshootingJournalctl.log
```

### Troubleshoot engine upgrades

During an upgrade, the upgrade file is sent to the engine server. A cron job running on the engine server checks if that file exists. The most common upgrade error is that the job is not running so the new installer does not run.

**NOTE:**

If the installer fails to start due to a permissions issue, even if running as root, add one of the following two arguments when running the installer:

- `--target <path>` - Extracts the installer files into the specified custom path.
- `--keep` - Extracts the installer files into the current working directory (without cleaning at the end).

If using installer options such as `-- -tools=false`, the option should come after the `--target` or `--keep` arguments. For example:

```
sudo ./d1-installer.sh --target /some/temp/dir -- -tools=false
```

1. SSH to the machine.

2. Check the d1 service status on the engine server. It is possible that it stopped or doesn't exist.

```
sudo systemctl status d1
```

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl status d1_<Engine _name>
```

3. Access the installer log on the engine server and review the error.

```
sudo vi /tmp/demisto_install.log
```

4. Rerun the installer on the engine using one of the following options. You can open a second window and run `watch df -h`. If the problem seems to be disk space, you should resolve the disk space issue and then rerun the installer.

a. Option 1

i. Download the installer from the user interface and copy it to the engine.

ii. Add the following commands:

```
sudo chmod +x installer.sh
```

```
sudo ./installer.sh -- -y
```

b. Option 2

i. Verify that /usr/local/demisto/d1\_upgrade.sh exists.

```
sudo chmod +x /usr/local/demisto/d1_upgrade.sh
```

```
sudo /usr/local/demisto/d1_upgrade.sh
```

ii. If `d1_upgrade.sh` does not exist, check if `/usr/local/demisto/archived_d1_upgrade.sh` exists and that it was created at the time of the attempted upgrade.

If the file exists and was created at the time of the attempted upgrade, run the following on the engine server:

```
sudo chmod +x /usr/local/demisto/d1_upgrade_archive.sh
sudo /usr/local/demisto/d1_upgrade_archive.sh
```

#### Troubleshoot engine connectivity

The following provides instructions for troubleshooting connectivity issues from the engine to the endpoint.

1. Follow the instructions in network troubleshooting.
2. Ensure that the engine can reach the endpoint by running the following command on the server engine.

```
sudo curl -kvv <endpointURL>
```

3. If the engine could not reach the endpoint, try the IP with curl instruction adding the http(s)://, or try using ping.

If this works, add the IP to the /etc/hosts file with the hostname and try to reach the endpoint again by running the following command on the engine server

```
sudo curl -kvv <endpointURL>
```

If this still fails, then this is an issue of connectivity between the engine and endpoint and you need to resolve this with your networking team.

4. After connectivity has been confirmed via curl:

- Try connecting within Docker without passing host networking.

```
docker run -it --rm demisto/netutils:1.0.0.6138 curl -kvv <endpointURL>
```

If this succeeds but the integration still fails, it could be an integration credentials issue. In that case, open a support case.

- If without passing host networking fails, run the following:

```
docker run -it --rm --network=host demisto/netutils:1.0.0.6138 curl -kvv <endpointURL>
```

If this succeeds, add "python.pass.extra.keys": "--network=host" to /usr/local/demisto/d1.conf and retest the integration.

If you see a Docker or Selinux issue, see Troubleshoot Docker networking issues .

5. If the installation issue remains, open a support case with logs from the engine.

- a. On the engine server, in /usr/local/demisto/d1.conf, set "LogLevel": "debug".

- b. Restart the d1 service and let it run for a few minutes.

```
sudo systemctl restart d1
```

#### NOTE:

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

- c. Capture a journalctl:

```
journalctl --since "1 day ago" > engineTroubleshootingJournalctl.log
```

- d. On the engine server, tar up the logs, conf, journalctl, and install log on the engine.

```
tar -cvzf engineLogs.tar.gz /var/log/demisto /usr/local/demisto/d1.conf /tmp/demisto_install.log
engineTroubleshootingJournalctl.log
```

#### Engine 443 error

This error might occur when a connection is established between an engine and the Cortex XSOAR tenant, because, by default, Linux does not allow processes to listen on low-level ports.

#### Error Message

```
listen tcp :443: bind: permission denied
```

#### Solution

- In the d1.conf file, change the port number to a higher one, for example, 8443.
- Run this command: `sudo setcap CAP_NET_BIND_SERVICE=+eip /path/to/binary`. After running this command the server should be able to bind to low-numbered ports.

## Bad handshake error

This error can occur in the engine logs relating to a bad handshake on the engine trying to connect to a Cortex XSOAR tenant.

### Error Message

```
Cannot connect to [wss:/xxx]: [wss://xxx|wss://xxx/]: websocket: bad handshake
```

### Solution

Verify that time is synchronized on the engine to a reliable NTP source. When timing is off on the engine, this can cause a failure during the SSL/TLS handshake process. When time is resynced, connectivity from the engine to the parent server should be restored.

## 4.11 | Troubleshoot integrations running on engines

The following are common errors that occur when integrations are running on an engine.

Troubleshoot engine import error or invalid syntax error

When running an integration on an engine, the most common errors are:

- **Broken Pipe**
- **"ImportError: No module named..."**
- **Invalid syntax**
- **Script failed to run: exec: "python": executable file not found in \$PATH (2603)**

These errors could indicate that the engine is not using Docker.

1. Use SSH to access the engine server.

2. Make sure Docker is healthy.

a. Ensure that Docker is installed and is running.

```
sudo systemctl status docker
```

If the Docker status is not good, restart your Docker.

```
sudo systemctl restart docker
```

b. Ensure Docker can run a container.

```
sudo docker run hello-world
```

If this fails, reinstall your Docker.

3. Access the d1.conf file on the engine server.

```
sudo vi /usr/local/demisto/d1.conf
```

4. Add the "python.engine.docker": true configuration to the d1.conf file and remove any other configurations related to python and Docker, such as "python.executable.no.docker".

5. Restart the system on the engine server.

```
sudo systemctl restart d1
```

#### NOTE:

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

6. Retest the integration from the user interface. This may take a few minutes because it may need to pull the relevant Docker image.

Troubleshoot permission denied

A common error message you may see when running integrations on engines is something like: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.35/images/json?t.

1. Determine if you are using a Docker group or Dockerroot group by running one of the following on the server engine:

- `ls -la /var/run/docker.sock`

The output from this command will show what user/group is running docker.sock. For example:

```
srw-rw----. 1 root docker 0 Apr 12 20:32 /var/run/docker.sock
```

shows that it's a Docker group and not Dockerroot.

- `cat /etc/group | grep docker`

This command shows if you are running Docker or Dockerroot.

**NOTE:**

Docker CE installations typically run Docker, while Docker EE installations typically run Dockerroot.

2. To fix a Docker user, run the following commands on the server engine:

- `sudo groupadd docker`
- `sudo usermod -aG docker demisto`
- `sudo systemctl restart docker`
- `sudo systemctl restart d1`

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

3. To fix a dockerroot user, run the following commands on the server engine:

- `sudo groupadd dockerroot`
- Set the dockerroot group in `/etc/docker/daemon.json`. For example: `{ "group": "dockerroot" }`.
- `sudo usermod -aG dockerroot demisto`
- `sudo chcon -Rt svirt_sandbox_file_t /var/lib/demisto/temp`
- `sudo systemctl restart docker`
- `sudo systemctl restart d1`

**NOTE:**

If the Allow running multiple engines on the same machine option is selected, run the command:

```
sudo systemctl restart d1_<Engine _name>
```

## 5 | Remote Repository Management

### Abstract

Configure and manage a remote repository in your dev/prod setup in Cortex XSOAR On-prem

Cortex XSOAR seamlessly integrates with private repositories, allowing you to develop and thoroughly test content in a secure environment in your development machine before pushing it to your production machine.

### 5.1 | Content management in Cortex XSOAR

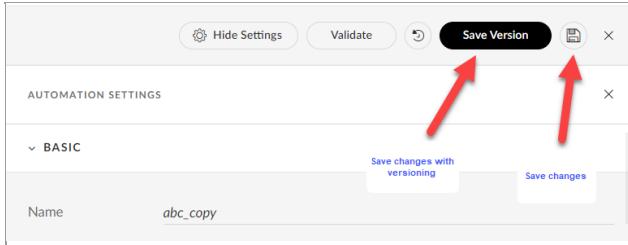
#### Abstract

Overview of how remote repositories work and how to configure a remote repository in Cortex XSOAR.

You can develop and manage content in Cortex XSOAR manually within the production tenant, using a CI/CD pipeline, or between development and production tenants using a remote repository.

#### Manual content management

Cortex XSOAR is a self-contained system. The Cortex XSOAR tenant serves as the content repository, content is developed using an IDE and stored locally.



If you only use a standalone tenant (with no development tenant), you can develop and manage content manually. You can save content versions and manage revisions locally for scripts, playbooks, integrations, etc. using the Save Version button. For all other content types, changes are automatically saved locally. You can also manage content by importing/exporting it in Cortex XSOAR.

## CI/CD for Cortex XSOAR

CI/CD pipelines are implemented using the XSOAR CI/CD content pack, which enables complete autonomy for developing, staging, and deploying custom content. This feature is intended for more advanced users who have an understanding of CI/CD concepts, with multiple developers working on different branches on their local machines.

Instead of building and maintaining code in a Cortex XSOAR development environment, you can build content from your private repository, and utilize third-party tools such as CircleCI and Jenkins. You can also use version control, perform code reviews, do linting and validations, use automatic testing, and run tests on development machines.

Content from a development instance is pushed to a Git repository. A CI/CD process runs to generate the required pack artifacts which are then uploaded to an artifact repository. These artifacts are deployed into Cortex XSOAR instances by running the Configuration Setup playbook.

For the complete CI/CD process flow, see XSOAR CI/CD.

### Content management using a remote repository

In Cortex XSOAR you can use a content management system with a private remote repository to develop and test content.

The development tenant pushes content to a remote repository and the production tenant or additional development tenants pull content from the remote repository.

If after setting up the remote repository feature you later decide to revert a tenant to standalone, go to Settings & Info → Settings → Advanced → Content Repository and toggle the Content repository slider to off. If you disable the remote repository feature, content on the tenant is not deleted. If you enable the remote repository feature again and the remote repository contains content, you need to choose which content to keep, either the content on the tenant or the content on the remote repository. We recommend backing up any content that you want to keep before enabling again.

#### The development tenant

The development tenant provides a safe environment to develop and test the functionality of custom content before using it in a production environment.

##### **NOTE:**

Development tenants are not intended for performance checks.

After you develop your content, if you want it to be available as part of a content update for the production tenant or additional development tenants, you must push content from a development tenant.

#### The production tenant

The production tenant is the operational environment for investigating real data. It pulls content as updates that you can install after the development tenant pushes it to the remote repository. For more information, see Install content on a production tenant.

#### Push and pull content between tenants

In a system with a single production tenant and several development tenants, only one development tenant can push content. The production tenant and any other development tenants pull from the one development tenant that is configured to push content. For example, you can have an additional development tenant for testing that pulls content from the development tenant configured to create and edit content.

All system content, content updates, and custom (user-defined) content are managed (downloaded, installed, edited, created, and updated) only in the development tenant that pushes content. For example, system content updates from Marketplace are only delivered to the development tenant that is configured to push. You cannot create or edit content in a production tenant or additional development tenant, they are configured only to pull content (except for dashboards and lists).

When pushing content from the development tenant, the content is synchronized and pulled into the production or other development tenants as content updates. For more information, see Push content from a development tenant.

You can decide which updates you want to push from the development tenant to pull tenants through the remote repository.

## 5.2 | Set up a private remote repository

### Abstract

Set up the private content repository feature.

When you set up a remote repository, you can add any private content repository that is Git-based, including GitHub, GitLab, and Bitbucket. Also, On-prem repositories are supported.

Although you can set up multiple development tenants, in a cluster of tenants that includes one production tenant and one or more development tenants, only one development tenant can push content. The production tenant and any other development tenants pull from the one development tenant that is configured to push content. After the remote repository is enabled in the production tenant, by default, the first development tenant that has been installed is set to push content to the remote repository. When you create additional development tenants, they are set to pull content from the remote repository.

If the content repository option is disabled for the production or development tenant, the tenant becomes standalone and does not push or pull content.

Once the development tenant is set up, you can only change content repository settings within the tenant.

### Use case scenarios for a private remote repository

The following are typical scenarios for setting up a private remote repository for the production and one or more development tenants.

- New development tenant and new or existing production tenant

The production tenant is first activated as a standalone (by default), and the private remote repository is then enabled in the production tenant. Once enabled, the first development tenant becomes the push tenant, the production tenant becomes a pull tenant, and any additional tenants need to set to pull tenants.

- Existing development and production tenants

The production and development tenants were managed in parallel with different sets of content.

### Before you begin

- Verify that you have network connectivity from Cortex XSOAR to the private remote repository. All communication goes through Cortex XSOAR, so it must have access to the remote repository. If direct access from Cortex XSOAR is not enabled you can use engines with access to the repository.
- If you are changing your remote repository settings, back up existing content to your local computer by navigating to Settings & Info → Settings → System → Server Settings → Custom Content and click Export all custom content.
- You must have Instance Administrator or Account Admin permission.
- Download and install the development image file. For more information, see Step 3. Set up a remote repository.

### How to set up a private remote repository

Perform the following procedures in the order listed below to set up a private remote repository.

#### NOTE:

When the first tenant (development or production) is enabled for the remote repository, the content from that tenant automatically populates the repository. When you first enable additional tenants (development or production) to the same remote repository, you will see the Specified repository is not empty window and have the option to use the content in the remote repository or replace the content with content from the new tenant.

These instructions describe enabling the production tenant first, so the remote repository will initially contain production tenant content. You can enable a development tenant first if you want the remote repository to initially contain the content from the development tenant.

#### Task 1. Enable the private remote repository on the production tenant

1. On the production tenant, go to Settings & Info → Settings → Advanced → Content Repository and toggle the Content repository slider to enable the content repository.

When set to On, the sync direction is Pull.

The Repository type is Private.

2. Define the Git settings using HTTPS or SSH.

- If your private Git remote repository uses personal access tokens instead of usernames and passwords, enter the access token in the password field and leave the username field blank.
- For repository vendors that use tokens, the token type is entered in the username field and the token is entered in the password field. Verify details with your vendor.
- If using SSH, only RSA private keys are supported. If your SSH connection uses a port other than port 22 (the default SSH port), you must include the SSH string and port number in the Repository URL field. In the following example, we use port 20017:

```
ssh://git@content.demisto.com:20017/~/my-project.git
```

3. Select the active branch on which you will be working.

4. In the Advanced section, the engine is set by default. You can change the engine by selecting from the list of available engines.

5. Save the settings.

#### Task 2. Enable the private remote repository on the development tenants

Once enabled, the first development tenant automatically becomes the push tenant.

1. On the development tenant, go to Settings & Info → Settings → Advanced → Content Repository and toggle the Content repository slider to enable the content repository.

When set to On, the sync direction for the development tenant is Push. Set the sync direction for any additional development tenants to Pull.

The Repository type is Private.

2. Define the GitHub settings using HTTPS or SSH.

- If your private Git remote repository uses personal access tokens instead of usernames and passwords, enter the access token in the password field and leave the username field blank.
- For repository vendors that use tokens, the token type is entered in the username field and the token is entered in the password field. Verify details with your vendor.
- If using SSH, only RSA private keys are supported. If your SSH connection uses a port other than port 22 (the default SSH port), you must include the SSH string and port number in the Repository URL field. In the following example, we use port 20017:

```
ssh://git@content.demisto.com:20017/~/my-project.git
```

3. Select the active branch on which you will be working.

4. (Optional) In the Advanced section, you can add any engines you want to connect.

5. Save the settings.

6. For any additional tenants that are enabled for the remote repository, select which content to keep and which to overwrite.

After the first tenant is enabled for the remote repository, its content automatically populates the remote repository (which in this example initially contains the production tenant content after it is enabled).

The Specified repository is not empty window opens. Options are:

- Existing content on your tenant: Keeps the existing content on your tenant and replaces the content on the specified repository. Cortex XSOAR checks if any other tenants are using the remote repository. If yes, this option is disabled. In this example, the remote repository was already enabled in the production tenant, so the remote repository holds production content. If you want to keep the content on the development tenant:
  1. Disable the remote repository in any additional enabled tenants. In this case, for the first development tenant, only the production tenant must be disabled.
  2. Select Existing content on your tenant for this tenant.
  3. Complete synchronization.
  4. Re-enable the remote repository in any additional tenants and select Existing content on the specified repository in each additional tenant.
- Existing content on the specified repository: Deletes the existing content on your tenant and replaces it with content from the specified repository.

7. Click Continue.

After completion, all tenants are now synced. You can start creating and testing content on the development tenant that you can push to production and additional development tenants when ready.

## 5.3 | Push content from a development tenant

### Abstract

Push content to a remote repository and control access for pushing content.

Once you develop your content, for it to be available as part of a content update for the production tenant, you must push the changes from the development tenant.

#### CAUTION:

You should not manually export content from the development tenant to import to the production tenant. Use only the procedures outlined in the documentation to ensure that your content is properly updated in the production tenant.

On each page you can decide whether to include or exclude items, which prevents them from being pushed to production, on a temporary or permanent basis. You can only exclude individual content items, not content packs.

The following types of content can be synchronized between development and production tenants:

- Scripts
- Playbooks
- Integrations: Integration instances are not pushed to the production tenant. Only customized integration YML files are pushed to the production tenant.
- Classifiers and mappers
- Content packs: When pushing a content pack to the production tenant, we recommend pushing all of the content for the content pack to work properly.
- Incidents: Including incident types, fields, and layouts
- Indicators: Including indicator types, fields, and layouts
- Evidence fields
- Pre-processing rules: If you reorder your pre-processing rules you must push all of the pre-processing changes to the production tenant.
- Lists
- Reports: When pushing a report to the production tenant, the time range set in the report on the development tenant does not sync with the production tenant.
- Dashboards
- Widgets

How to push content from a development tenant

1. In the development tenant, go to Settings & Info → Settings.

2. Under the Local Changes section, go to the relevant page according to the items you want to push:

- Items

Content that is not related specifically to a content pack. For example, customized scripts or playbooks. When creating custom content, the content is automatically added here. If you have already pushed a content pack and later edit one of its content items, the edited items appear in the Items page, not the Content Packs page.

- Content Packs

All of the content that is specific to the content packs you installed from Marketplace.

- Content Pack Items

If you do not want to install the whole content pack, you can install specific items in the content pack.

3. Select the items you want to push to production, and click Push.

4. If the items have dependencies, review the contents and click Push

Sometimes you may not want to push all content, content pack dependencies, etc. For example, when a user makes a change in a playbook that includes a script dependency to which another user is adding a feature, and the change does not require the new feature (version) of the script, you can push the playbook without the new script.

5. In the dialog box, add an optional message and click Push.

6. On the production tenant, Install content on the production tenant.

## 5.4 | Install content on a production tenant

### Abstract

Install new content that has been pushed from the development tenant to the production tenant.

After you push content from the development push tenant, on the right-hand side of any page in the production tenant you have the option to install the content. In case of conflicts, you have a choice whether to keep local content or delete and replace.

1. On the right-hand side of any page, click Install New Content.

2. In the dialog box, click Install Content.

You can also check for new content that has been pushed.

3. If conflicts appear, click Resolve conflicts.

4. In the Action column, select one of the following:

- Skip: Keeps the local content in your production environment.
- Replace: Deletes the local content and installs the content from the content repository.

5. Click OK to install the content.

## 5.5 | Remote repository troubleshooting

### Abstract

Scenarios that occur when managing content with a remote repository in Cortex XSOAR.

The following scenarios can occur when managing the remote repository.

#### Pointing to a non-empty branch when enabling a tenant

If you configure a tenant to use a remote repository, you have two options:

- Overwrite all content in the tenant with content from the repository.
- Overwrite all content in the remote repository with content from the tenant.

To overwrite the remote repository with content from the tenant, you must use an empty branch. If the branch is not empty, you will get an error message prompting you to select an empty branch. Alternatively, you can select the first option and overwrite all content in the tenant with the content from the remote repository.

#### Switching between remote repository types

If you switch between built-in and private remote repository types, you get a warning that switching between repository types may result in the loss of all version history.

To keep your content history, select Existing content on your tenant to overwrite all content in the remote repository with content from your tenant.

## 6 | Users and Roles Management

### Abstract

Configure and manage roles, users, and user groups, and set up authentication in Cortex XSOAR On-prem.

Learn how to configure and manage users, roles, and user groups. Assign roles and set up authentication for users.

## 6.1 | Users and roles in Cortex XSOAR

### Abstract

Set up and configure roles and user groups in Cortex XSOAR. Configure authentication, and manage and create users.

Cortex XSOAR uses role-based access control (RBAC) to manage roles with specific permissions for controlling user access. RBAC helps manage access to Cortex XSOAR components, so that users, based on their roles, are granted the minimal access required to accomplish their tasks.

## Roles

Roles enable you to define permissions for specific components, such as incident data, playbooks, scripts, and jobs. For example, you can create a role that allows users to edit the properties of incidents, but not delete incidents. You can create new roles or customize out-of-the-box roles.

If you assign one or more roles to an incident, only users with those roles can view and interact with the incident. For example, you might have an incident with sensitive data that should only be accessible to Tier-1 analysts and managers.

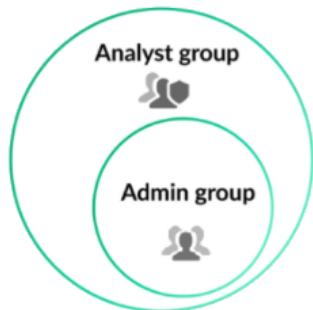
Roles can also be used to define permissions for integration commands. On the Integration Permissions page, you can assign roles to specific integration instances (all commands for that instance) or specific integration instance commands. For example, you could assign the Generic Export Indicators Service integration instance the Account Admin role, or you could restrict certain commands in the Core Rest API to a specific role. For more information, see Integration Permissions.

## User groups

While roles can be assigned directly to users, we recommend instead creating user groups. Each user group has a single role associated with it, but each user group can contain multiple users and user groups can be nested within each other, enabling you to further refine your RBAC requirements. Users can belong to multiple user groups.

## Nested roles

Cortex XSOAR 8 uses group nesting, where the group with higher permissions includes the permissions of the group with lower permissions, but as a subset of the group with lower permissions. For example, the Admin user group is included as a subset of the Analyst user group, as shown in the following graphic. The Admin role includes the permissions of the Analyst role, the same as in Cortex XSOAR 6.



For example, Content Developer and Analyst user groups include Employee user group permissions, and are nested in the Employee user group.

## Authentication

You can create users locally or by using SAML Single Sign-On (SSO) in the tenant. After you create users, they authenticate by either:

- Using a username and password
- Using SSO

## Manage users

You can manage users including resetting passwords, sending invitations, and removing users.

By default, users do not have roles assigned and do not automatically have access to tenant data until you assign them a role or add them as members of a user group that has an assigned role.

## 6.2 | Roles management

### Abstract

Configure roles in the Cortex XSOAR tenant.

You can assign the following permissions to various components in Cortex XSOAR:

Permission	Description
None	No access to the specified component.

Permission	Description
View	View, but not edit the specified component.
View/Edit	View and edit the specified component.

#### Out-of-the-box roles

Cortex XSOAR includes the following out-of-the-box roles:

Role	Type	Description
Account Admin	Predefined	<p>The user who supplied their credentials when installing Cortex XSOAR is assigned the Account Admin role. This user has view/edit permissions for all components and access to all pages in the Cortex XSOAR tenant (the same view/edit permissions as the Instance Administrator). You cannot create additional Account Admin roles in Cortex XSOAR.</p> <p>You cannot edit this role. You can copy the role by saving it as a new role and then change permissions.</p>
Instance Administrator	Predefined	<p>View/edit permissions for all components and access to all pages in the Cortex XSOAR tenant. The Instance Administrator can also assign the Instance Administrator role to other users on the tenant. If the application has predefined or custom roles, the Instance Administrator can assign those roles to other users.</p> <p>You cannot edit this role. You can copy the role by saving it as a new role and then change permissions.</p>
Analyst	Custom	A mix of view and view/edit permissions for all components and access to all pages in the Cortex XSOAR tenant.
Read-Only	Custom	Read permissions for all components and pages in the Cortex XSOAR tenant.

#### NOTE:

By default, users do not have roles assigned. If no direct or user group role has been assigned, users have no permission to view or edit data in Cortex XSOAR.

#### Next steps

Before you start creating or customizing roles, do the following:

- Review the Role-based permissions topic.
- Decide whether you want to assign roles to users directly or through membership in user groups (recommended) in the Cortex XSOAR tenant.

#### 6.2.1 | Role-based permissions

##### Abstract

Describes the role-based permissions available in Cortex XSOAR .

When creating or editing a role, you can set permission levels (RBAC) for specific components (such as playbooks, scripts, jobs, etc.), set page access, define preset role queries, and set up shift management.

In the Cortex XSOAR tenant, you can set permission levels for each role by going to Settings → Settings & Info → Access Management → Roles and editing or creating a new role.

#### NOTE:

You can only create, edit, copy, or delete a role if you have administrator (Instance/Account Admin) permissions. You cannot change the predefined (Instance Administrator or Account Admin) role permissions.

Each role contains the following tabs:

##### The Components tab

The Components tab includes the following areas where you can define permissions.

## Data

**NOTE:**

You need to select View/Edit to see the permissions for the components.

Component	Description
Data	Sets the permission level generally for data related to investigations, dashboards, and reports. If you select none, the user role cannot view and edit incidents, indicators, dashboards, and reports.
Execute potential harmful actions	Allows executing integration commands that are marked as Potentially Harmful in the integration code/settings. Users can run these commands from the CLI. Playbook tasks that use these commands would not be affected, as they are run by the DBot user as part of playbook execution.
Edit incident properties	Allows editing an incident's fields from the layout or via the Actions menu.
Change the incident status	Allows editing an incident's status, which includes closing an incident, or investigating an incident which is in the Pending status.
Delete incidents	Allows deleting incidents. We recommend only granting this permission to the default Admin or select Administrators.
Manage incident workplan	Allows interacting with the playbook for the incident.
Edit indicators	Allows editing indicators either from the Threat Intel pane or when viewing the indicator via its full layout or quick view tab.
Retain incidents	Allows marking an incident for permanent retention or disabling retention for an incident. Retained incidents cannot be deleted.
Incidents Table Actions	Limits table actions in the Incidents page, such as delete, command line actions, edit, close, and mark as duplicate.

## Exclusion list

Component	Description
EXCLUSION LIST	Limits permissions when editing, creating, or deleting an indicator in an exclusion list.

## Playbooks

Component	Description
Playbooks	<p>Limits permissions for creating, editing, and deleting playbooks.</p> <p><b>NOTE:</b></p> <p>You can also add, change, and remove roles from a playbook by clicking Settings on the Playbooks page.</p>

## Scripts

Component		Description
Scripts	<p>Limits permissions for managing scripts. If the role has read/write permissions, you can enable user roles to create scripts that run as a Super User.</p> <p>On the Scripts page, you can define which roles are permitted to run a script, and according to which role the script executes.</p>	

## Jobs

Component		Description
Jobs	<p>Limits permissions for managing jobs. Roles that have read permissions to content items, retain partial read access. If you do not want to retain partial read access, set the permission to none.</p>	

## Marketplace

Component		Description
Marketplace	<p>You can set the following permissions for Marketplace.</p> <ul style="list-style-type: none"> <li>• None: The user role is not able to view Marketplace.</li> <li>• View: The user role can view, but not take any action in Marketplace.</li> <li>• View/Edit: The user role can install, upgrade, downgrade, and delete content packs in Marketplace.</li> </ul>	

## Configurations

Section	Component	Description
General Setting	Auditing	Whether a user role can access the Management Audit Logs page.
General Setting	Alert Notifications	Whether a user role can forward Management Audit Logs to an email distribution list or a syslog server.
Integrations	Public API	Whether a user role can access the API Keys page. View/Edit enables the user role to manage API keys, including creating, editing, and deleting.  <b>NOTE:</b> If you select None, the user role can still use the API, but they cannot view API keys in the UI.
Integrations	Integrations	Whether a user role can view, add, edit, or delete integration instances, pre-process rules, and classify and map incidents and indicators.  Roles that have view permissions for content items, retain partial read access. If you do not want to retain partial read access, set the permission to none.

Section	Component	Description
Integrations	Integrations Permissions	<p>Enables you to set the permissions on the Integration Permissions page. Integration permissions enable you to assign different permission levels for the same command in each instance.</p> <ul style="list-style-type: none"> <li>None: The user role cannot view the page.</li> <li>View: The user can view the page.</li> <li>View/Edit: The user can view and edit permissions.</li> </ul>
Integrations	Credentials	Whether a user role can add, edit, or delete integration credentials.
Object Setup	Fields and Types	Whether a user can add, edit, or delete fields and types for indicators, incidents, and Threat Intel Reports.
Object Setup	Layouts	Whether a user can add, edit, or delete layouts for indicators, incidents, and Threat Intel Reports.
Advanced	Administration	Limits permissions for administration tasks, such as server configurations, audit trails, and changing logos.

#### Page Access

Select the pages the user role should have access to.

**NOTE:**

If you select None in the Data section, even though you allow page access, the user role cannot access those pages. For example, if you allow page access to Dashboards, but DATA is set to none, the user role cannot access the Dashboards page.

#### The Advanced tab

Define access to default dashboards, pre-set role queries, and shifts. For more information, see Manage roles in the Cortex XSOAR tenant.

Component	Description
DEFAULT DASHBOARDS	Select the default dashboards for each role. If a user has not modified their dashboard, these dashboards are added automatically; otherwise, users can add these dashboards to their existing dashboards.
PRE-SET ROLE QUERIES	Select the preset query for each of the available components.
SHIFTS	Weekly shifts start on Sunday and are specified in the UTC zone.

#### 6.2.2 | Manage roles in the Cortex XSOAR tenant

##### Abstract

Manage roles in Cortex XSOAR tenant.

On the Roles page, you can view all roles in Cortex XSOAR, whether they are custom roles, who created the role, when it was created, and additional information about the roles. When right-clicking on a role, you can edit the role and permissions.

Cortex XSOAR includes the following role types:

- Predefined roles:** Includes Account Admin and Instance Administrator roles. Permissions cannot be changed. You can create a duplicate of these roles but you cannot remove them.
- Custom roles:** Includes out-of-the-box roles and custom roles.

When right-clicking a role, you can perform several actions, such as editing a role, saving it as a new role, and removing a role (deleting a role that is not assigned to a user).

#### Create a role

The roles you create provide more granular access control. You can add as many new roles as you need and combine them with user groups. When you create or edit a role, you can perform activities such as adding permissions and permission levels, defining shift periods, and setting default dashboards.

To create, edit, or delete a role, you must have administrator permissions.

#### TIP:

For analysts, we recommend the following settings:

- Remove the ability to delete incidents in production environments (DATA → Data → Delete incidents).
- Remove the ability to install, delete, and contribute to Marketplace which should be reserved for engineers and administrators. We recommend setting Marketplace permissions for analysts to None or View.
- Remove access to API keys. Under CONFIGURATIONS, set the Public API access to None or View. If you select None, the user role can still use the API, but they cannot view API keys in the UI.

1. In the Cortex XSOAR tenant, select Settings & Info → Settings → Access Management → Roles → New Role.

#### TIP:

We recommend making a copy of out-of-the-box roles and editing the copies, rather than creating new roles, to avoid missing any important permissions.

2. Add the Role name and a meaningful Description.

3. In the Components tab, add the permissions as required. For more information, see Role-based permissions.

4. In the Advanced tab, do the following:

- Define dashboards
- Define preset role queries
- Set up shift management

5. Save the role.

6. You can create user groups and add roles to them (recommended), assign roles directly to users after they have been added, or both.

#### Define dashboards

In a production environment, an administrator defines the default dashboard for each user and selects the default dashboards that the user sees when logging into the tenant, depending on a user's role. If a user has not modified their dashboard, these dashboards are added automatically, otherwise users can add these dashboards to their existing dashboards. These default dashboards can be removed but not deleted, and can be added again if required.

If you select Only allow these dashboards, the current role will only be able to access the designated default dashboards. The role will not be able to import, edit, create, or duplicate any other dashboards. It will not be possible to share any additional dashboards with this role.

If a user had a role that accessed certain dashboards and the Admin changes the role to access only specific default dashboards, then the user will lose access to the previous dashboards.

- The admin unselects Only allow these dashboards for the role.
- The user exports the relevant dashboards and shares them with the Admin.
- The Admin then adds the relevant dashboards to the default dashboards list for the role and reselects Only allow these dashboards.

#### Define preset role queries

A default query associated with a user's role is useful for new users who are unsure which query to use when accessing the incident, indicators, and jobs pages. When accessing the relevant page, the role's preset query is the default query for a new user. Existing users can keep their default query, but the default query is available for selection.

When you define or edit a role, in the Advanced tab, you can view or edit a list of queries for incidents, indicators, and jobs, which are based on your saved queries for these components.

1. On the component page, such as the Incidents page, create the query.
2. Save the query (next to the query field).
3. Go to Settings & Info → Settings → Roles.
4. Select the role you want to update.

5. In the Advanced tab, select the relevant query.

The list of queries is populated with your own saved queries.

6. Save the role.

The preset query runs when a user with that role accesses that component page. If you update the preset query for a role, the query is added to the users' queries, but not as the preset query. If you delete one of your queries after you configure a role, the role's list of queries is unaffected.

Users can view the preset query based on their role when clicking the ellipsis on each component page. The preset role query has (Pre-set) appended to the name of the query. Although users can change their default query, they cannot delete the preset role query. If a user has permissions for multiple roles, the user sees multiple queries. The preset role queries appear at the top of the saved queries list.

If a user's role changes, the user's preset role query is automatically updated.

#### Set up shift management

Shift management helps you define multiple shifts within Cortex XSOAR. You can create user groups, so each shift can be assigned to a user group role, and you can assign one or more analysts across different shifts.

With shift management, you can:

- Enable incidents to be routed automatically to analysts based on shifts, ensuring full staff coverage for incoming incidents.
- Define multiple shifts, which can be added to a role, and in turn assigned to a user group.
- Automatically reassign incidents when shifts change.

#### NOTE:

To view suggestions for on-call users to assign to an incident, run the `getOwnerSuggestions` command with the `shiftOnly=true` argument.

When assigning an incident, you can manually assign it to analysts who are on-call or you can use the `AssignAnalystToIncident` script with argument `onCall=true` to automatically assign it to users who are on call and active.

How to define and assign shifts

1. Create or edit a role.
2. In the Advanced tab, Shifts field, click Add Shift and add the required period.

Weekly shifts start on Sunday and are specified in the UTC timezone format.

For example, create a role called `First Shift` and add a shift starting on Sunday and ending Monday.

3. Save the role.
4. Create a user group and assign the shift role to the user group.

For more information about how to create a user group, see User group management.

5. Assign one or more users to the user group.

#### TIP:

(Optional) We recommend installing the Shift Management content pack. This content pack includes widgets to view Roles Per Shift, Users On-Call, and more in a dashboard, as well as playbooks and scripts for assigning incidents to on-call users.

## 6.3 | User group management

### Abstract

Create user groups, and assign roles and users to further refine your requirements,

Users are assigned roles and permissions either by being assigned a role directly or by being assigned membership in one or more user groups. A user group can only be assigned to a single role, but users can be added to multiple groups if they require multiple roles. You can also nest groups to achieve the same effect. Users who have multiple roles through either method will receive the highest level of access based on the combination of their roles.

For example:

- Joe has an Analyst role and is a member of the Tier-1 Analyst user group, which is assigned the Triage role. Joe has the permissions of the Analyst role and the Triage role. Joe is assigned 2 roles, and has the highest permission based on the combination of both roles.
- John is a member of two user groups - Tier-1 Analyst and Tier-2 Analyst. One group is configured to use the Triage role and the other group is configured to use the Incident Response role. John is assigned both roles and has the highest permissions based on the combination of all roles.
- Jack is a member of the Tier-2 user group which has an Incident response role. This user group is included in a Tier-3 user group (Threat Hunter role), added as a nested group. Jack is assigned both roles and has the highest permissions based on the combination of all roles.

On the User Groups page, you can create a new user group for several different system users or groups. You can see information including the details of all user groups, the roles, nested groups, IdP groups (SAML), and when the group was created/updated.

You can also right-click in the table to edit, save as a new group, remove (delete) a group, and copy text to the clipboard.

#### How to create a user group

1. Go to Settings & Info → Settings → Access Management → User Groups.
2. To create a new user group for several different system users or groups, click New Group, and add the following parameters:

Parameter	Description
Name	Name of the user group.
Description	Description of the user group.
Role	Select the group role associated with this user group. You can only have a single role designated per group.
Users	<p>Select the users you want to belong to this user group.</p> <p><b>NOTE:</b></p> <p>If users have been created locally, but you want them to access the tenant through SSO only, skip this field and add only SAML group mapping after SSO is set up, otherwise, users can access the tenant through their username and password and through SSO.</p> <p>If you have not yet created any users, skip this field and add them later. See Set up authentication.</p>
Nested Groups	<p>Lists any nested groups associated with this user group. If you have an existing group you can add a nested group.</p> <p>User groups can include multiple users and nested groups, which inherit the permissions of parent user groups. The user group will have the highest level of permission.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>• Group A has Tier-1 Analyst permissions</li> <li>• Group B has Tier-2 Analyst permissions</li> </ul> <p>If you add Group A as a nested group in Group B, Group A inherits Group B's permissions (Tier-1 and Tier-2 permissions).</p>
SAML Group Mapping	<p>Maps the SAML group membership to this user group. For example, you have defined a Cortex XSOAR Admins group. You need to name this group exactly how it appears in Okta.</p> <p>You can add multiple groups by separating them by a comma.</p> <p><b>NOTE:</b></p> <p>When using Azure AD for SSO, the SAML group mapping needs to be provided using the group object ID (GUID) and not the group name.</p> <p>If you have not set up SSO in your tenant, skip this field and add it later. After you have added it, follow the procedure relevant to your IdP. For example, see Task 6. Map SAML Group Memberships to Cortex XSOAR User Groups.</p>

3. Create a new user group.

## 6.4 | Set up authentication

### Abstract

Decide whether you want to add users locally or through SSO in Cortex XSOAR On-prem.

You can create users locally or by using SSO in the tenant. Users authenticate by doing one of the following:

- Authenticate locally

After you create users, they authenticate using their username and password. For more information, see Create users in Cortex XSOAR.

- SAML single sign-on

Users can be authenticated using your IdP provider such as Okta, Ping, or Azure AD. You can use any IdP that supports SAML 2.0.

After you have created users, add them to user groups or assign roles directly.

SSO has the following advantages:

- Enforces multi-factor authentication (MFA) and any conditional access policies on the user login at the IdP before granting a user access to Cortex XSOAR.
- Maps SAML group memberships to user groups and roles, allowing you to manage role-based access control.
- Removes access to Cortex XSOAR when a user is removed or disabled in the IdP.

### 6.4.1 | Create users in Cortex XSOAR

#### Abstract

Create users in Cortex XSOAR on-prem by inviting users to access Cortex XSOAR using their username and password.

To add users locally (not SSO), you can either send an invitation to users by adding their details manually or by uploading a CSV file with multiple users.

#### PREREQUISITE:

Before inviting users to Cortex XSOAR:

- Add an email integration instance, such as EWS v2, EWS O365, or Gmail.

When you invite users to Cortex XSOAR, an email is sent to their email address, using the integration instance. After inviting users you can also copy the invitation link and send it to the users.

- Review the predefined roles and consider whether you want to create roles and user groups before or after inviting users to Cortex XSOAR.

When you send invitations to users you can invite them without a role assigned. Although they can log into Cortex XSOAR, they cannot view/edit any data. This is useful if you want to add multiple users at one time and then define roles at a later stage, rather than users having access immediately after accepting an invitation .

- If you want to add multiple users with different roles, you should split them up according to their roles before inviting them. You can add them manually or in a CSV format according to their role. Alternatively, leave blank for no role.

After you invite a user, an invitation is valid for seven days from the time it was sent. If not accepted, the invitation expires unless the invite expiration is reset. Once users accept the invite they have access and permissions within Cortex XSOAR according to their assigned roles. You have the option to copy the invite link, resend, or cancel the invitation. The invite link cannot be used after the invite has expired.

How to create users

1. Select Settings & Info → Settings → Access Management → Users → Add User.

2. In the Send Invitation section, select one of the following:

- Manually enter users
  1. Add the email address and first and last names.
  2. Add the user.
  3. Repeat the above steps for any other users you want to add, if they have the same role, user group, or no role.
  4. (Optional) Select the Role and User Group, if relevant.  
You cannot select different roles and user groups for multiple users.
  5. Invite the users.
- Upload a file
  1. (Optional) Download the example file.
  2. Add the users' details to the file and upload it.

**NOTE:**

- The file must be in a CSV format.
- At least one row must exist including email address, first and last names.
- You cannot select different roles and user groups for each user. If you want different roles and user groups for each set of users upload separate files.

3. (Optional) Select the role and User Group.
4. Invite the users.
5. If you want to invite additional users, repeat these steps.

If you have set up a mail integration, users will receive a link to access Cortex XSOAR. When accessing the link, users need to complete the password and will be able to log in.

3. Unless already done so, add roles and user groups to users.

#### 6.4.2 | Authenticate users using SSO

##### Abstract

Set up authentication in the Cortex XSOAR tenant using SSO.

Cortex XSOAR enables you to securely authenticate system users across enterprise-wide applications and websites with one set of credentials using single sign-on (SSO) with SAML 2.0. System users can authenticate using your organization's Identity Provider (IdP), such as Okta or PingOne. You can integrate with any IdP that is supported by SAML 2.0.

Configuring SSO with SAML 2.0 is dependent on your organization's IdP. Some of the parameter values need to be supplied from your organization's IdP and some need to be added to your organization's IdP. You should have sufficient knowledge about IdPs, how to access your organization's IdP, which values to add to Cortex XSOAR, and which values to add to your IdP fields.

**NOTE:**

- To set up SSO authentication in the tenant, you must be assigned an Instance Administrator or Account Admin role.
- SAML 2.0 users must log in to Cortex XSOAR using the FQDN (full URL) of the tenant. To allow login directly from the IdP to Cortex XSOAR, you must set the relay state on the IdP to the FQDN of the tenant.
- If you have multiple tenants, you must set up the SSO configuration separately for each tenant, both in the IdP and in Cortex XSOAR.
- Create groups in your IdP that correspond to the roles in Cortex XSOAR and assign users to those groups in your IdP. Users can belong to multiple groups and receive permissions associated with multiple roles. Add the appropriate SAML group mapping from your IdP to each Cortex XSOAR role.

If you are configuring Okta or Azure, follow the procedure in Okta or Azure AD. You can also adapt these instructions for use with any similar SAML 2.0 IdP.

1. In Cortex XSOAR go to Settings & Info → Settings → Access Management → Authentication Settings.
2. In the Login Options tab, toggle SSO Disabled to on.

You can see the SSO settings, so you can configure them according to your organization's IdP.

3. If you want to add another SSO connection to enable managing user groups with different roles and different IdPs, click Add SSO Connection.

Different SSO parameters for an SSO are displayed to configure according to your organization's additional IdP.

**NOTE:**

- The first SSO cannot be deleted, it can only be deactivated by toggling SSO Enabled to off.

- The Domain parameter is predefined for the first SSO.

If you add additional SSO providers, you must provide the email Domain in the SSO Integration settings for all providers except the first. Cortex XSOAR uses this domain to determine which identity provider the user should be sent to for authentication.

- When mapping IdP user groups to Cortex XSOAR user groups, you must include the group attribute for each IdP you want to use. For example, if you are using Microsoft Azure and Okta, your Cortex XSOAR user group SAML Group Mapping field must include the IdP groups for each provider. Each group name is separated by a comma.

#### 4. Set the following parameters using your organization's IdP.

- General parameters**
- IdP Attribute Mapping**
- Advanced Settings** (optional)

#### 5. Save your changes.

Whenever an SSO user logs in to Cortex XSOAR, the following login options are available.

- Sign-in with SSO

If you have enabled more than one SSO provider, an optional email field appears. If the user does not enter an email address or if the email address does not match an existing domain, the user is automatically directed to the default IdP provider (the first in the list of SSO providers in the Authentication Settings). If the user enters an email address and it matches a domain listed in the Domain field in the SSO Integration settings for one of your IdPs, Sign-In with SSO sends the user to the IdP associated with that email domain.

- Sign-in with your local credentials

Users login with their local username and password.

#### General parameters

Parameter	Description
IdP SSO or Metadata URL	Select the option that meets your organization's requirements.  Indicates your SSO URL, which is a fixed, read-only value based on your tenant's URL using the format <a href="https://&lt;name of Cortex-XSOAR&gt;.crtx.paloaltonetworks.com/idp/saml">https://&lt;name of Cortex-XSOAR&gt;.crtx.paloaltonetworks.com/idp/saml</a> . For example, <a href="https://tenant1.crtx.paloaltonetworks.com/idp/saml">https://tenant1.crtx.paloaltonetworks.com/idp/saml</a>  You need this value when configuring your IdP.
IdP SSO URL	Specify your organization's SSO URL, which is copied from your organization's IdP.
Metadata URL	
Audience URI (SP Entity ID)	Indicates your Service Provider Entity ID, also known as the ACS URL. It is a fixed, read-only value using the format, <a href="https://&lt;name of Cortex-XSOAR&gt;.paloaltonetworks.com">https://&lt;name of Cortex-XSOAR&gt;.paloaltonetworks.com</a> . For example <a href="https://tenant1.crtx.paloaltonetworks.com">https://tenant1.crtx.paloaltonetworks.com</a> .  You need this value when configuring your organization's IdP.
Default Role	(Optional) Select the default role that you want any user to automatically receive when they are granted access to Cortex XSOAR through SSO. This is an inherited role and is not the same as a direct role assigned to the user.
IdP Issuer ID	Specify your organization's IdP Issuer ID, which is copied from your organization's IdP.

Parameter	Description
X.509 Certificate	Specify your X.509 digital certificate, which is copied from your organization's IdP.
Domain	Relevant only for multiple SSOs. For one SSO, this is a fixed, read-only value. Associate this IdP with a specific email domain (user@<domain>). When logging in, users are redirected to the IdP associated with their email domain or to the default IdP if no association exists.

#### IdP attribute mapping

These IdP attribute mappings are dependent on your organization's IdP.

Parameter	Description
Email	Specify the email mapping according to your organization's IdP.
Group Membership	Specify the group membership mapping according to your organization's IdP. <b>NOTE:</b> Cortex XSOAR requires the IdP to send the group membership as part of the SAML token. Some IdPs send values in a format that include a comma, which is not compatible with Cortex XSOAR. In that case, you must configure your IdP to send a single value without a comma for each group membership. For example, if your IdP sends the Group DN (a comma-separated list), by default, you must configure IdP to send the Group CN (Common Name) instead.
First Name	Specify the first name mapping according to your organization's IdP.
Last Name	Specify the last name mapping according to your organization's IdP.

#### Advanced settings

The following advanced settings are optional to configure and some are specific for a particular IdP.

Parameter	Description
Relay State	(Optional) Specify the URL for a specific page that you want users to be directed to after they've been authenticated by your organization's IdP and log in to Cortex XSOAR.
IdP Single logout URL	(Optional) Specify your IdP single logout URL provided by your organization's IdP to ensure that when a user initiates a logout from Cortex XSOAR, the identity provider logs the user out of all applications in the current identity provider login session.
SP Logout URL	(Optional) Indicates the Service Provider logout URL that you need to provide when configuring a single logout from your organization's IdP to ensure that when a user initiates a logout from Cortex XSOAR, the identity provider logs the user out of all applications in the current identity provider login session. This field is read-only and uses the following format <code>https://&lt;name of Cortex-XSOAR&gt;.crtx.paloaltonetworks.com/idp/logout</code> , such as <code>https://tenant1.crtx.paloaltonetworks.com/idp/logout</code> .

Parameter	Description
Service Provider Public Certificate	(Optional) Specify your organization's IdP service provider public certificate.
Service Provider Private Key (Pem Format)	(Optional) Specify your organization's IdP service provider private key in Pem Format.
Remove SAML RequestedAuthnContext	(Optional) Requires users to log in to Cortex XSOAR using additional authentication methods, such as biometric authentication.  Selecting this removes the error generated when the authentication method used for previous authentication is different from the one currently being requested. See here for more details about the RequestedAuthnContext authentication mismatch error.
Force Authentication	(Optional) Requires users to reauthenticate to access the Cortex XSOAR tenant if requested by the IdP, even if they already authenticated to access other applications.

#### Troubleshoot SSO issues

The following list describes the common errors and issues when using SAML 2.0 authentication.

- Errors in your IdP could mean the Service Provider Entity ID and/or Service Identifier are not properly configured in the IdP or in the Cortex XSOAR settings.
- SAML attributes from the IdP are not properly mapped in Cortex XSOAR. The attributes are case sensitive and must exactly match in your IdP and in the Cortex XSOAR IdP Attributes Mapping.
- Group memberships from the IdP have not been properly mapped to Cortex XSOAR user groups. Verify the values your identity provider is sending, to properly map the groups in Cortex XSOAR.
- The identity provider is not configured to sign both the SAML response and the assertion on the login token. Your IdP must be configured to sign both to ensure a secure login.
- If you require further troubleshooting, we recommend using your browser's built-in developer tools or additional browser plugins to capture the login request and SAML token.

#### 6.4.3 | Set up Okta as the Identity Provider Using SAML 2.0

This topic provides specific instructions for using Okta to authenticate your Cortex XSOAR users. As Okta is third-party software, specific procedures, and screenshots may change without notice. We encourage you to also review the Okta documentation for app integrations.

To configure SAML SSO in Cortex XSOAR, you must be a user who can access the Cortex XSOAR tenant and have either the Account Admin or Instance Administrator role assigned.

The following video is a step-by-step guide to configure SSO in Cortex XSOAR (specific Okta instructions begin at minute 3:30).



##### Task 1. Configure Okta Groups

Within Okta, assign users to groups that match the user groups they will belong to in Cortex XSOAR. Users can be assigned to multiple Okta groups and receive permissions associated with multiple user groups in Cortex XSOAR. Use an identifying word or phrase, such as Cortex XSOAR, within the group names. For example, Cortex XSOAR Analysts. This allows you to send only relevant group information to Cortex XSOAR, based on a filter you will set in the group attribute statement.

Create a list of the Okta groups and their corresponding Cortex XSOAR user groups (or the Cortex XSOAR user groups you intend to create) and save this list for later use when configuring user groups in Cortex XSOAR.

#### Task 2. Copy Single SSO and Audience URI Values from Cortex XSOAR

1. In Cortex XSOAR, go to Settings & Info → Settings → Access Management → Authentication Settings.
2. In the Login Options tab, toggle SSO Disabled to on.
3. Expand the SSO Integration settings.
4. Copy and save the values for Single Sign-On URL and Audience URI (SP Entity ID).

Both values are needed to configure your IdP settings.

You cannot save the enabled SSO Integration at this time, as it requires values from your IdP.

#### Task 3. Configure Cortex XSOAR Application in Okta

1. In Okta, create a Cortex XSOAR application and Edit the SAML Settings.
2. Paste the Single sign-on URL and the Audience URI (SP Entity ID) that you copied from the Cortex XSOAR SSO settings. The Audience URI should also be pasted in the Default RelayState field, which allows users to log in to Cortex XSOAR directly from the Okta dashboard.
3. Click Show Advanced Settings, verify that Okta is configured to sign both the response and the assertion signature for the SAML token, and then click Hide Advanced Settings.
4. Cortex XSOAR requires the IdP to send four attributes in the SAML token for the authenticating user.
  - Email address
  - Group membership
  - First Name
  - Last Name

Configure Okta to send group memberships of the users using the memberOf attribute. Use the word or phrase you selected when configuring Okta groups (such as Cortex XSOAR) to create a filter for the relevant groups.

5. Copy the exact names of the attribute statements from Okta and save them, as they are required to configure the Cortex XSOAR SSO integration. In the example above, the names are FirstName, LastName, Email, and memberOf. The attribute names are case-sensitive.

#### Task 4. Copy IdP SSO URL, Identity Provider Issuer, and X.509 Certificate Values

1. In Okta, from your Cortex XSOAR application page, click View SAML setup instructions. If you do not see this button, verify you are on the Sign On tab of the application.
2. Copy and save the values for Identity Provider Single Sign-On URL, Identity Provider Issuer, and the X.509 Certificate. These values are needed to configure your Cortex XSOAR SSO Integration.

#### Task 5. Configure the Cortex XSOAR SSO Integration

1. In Cortex XSOAR go to Settings & Info → Settings → Access Management → Authentication Settings.
2. In the Login Options tab, toggle SSO Disabled to on.
3. Expand the SSO Integration settings.
4. Use the following table to complete the SSO Integration settings, based on the values you saved from Okta.

Okta	Cortex XSOAR Field
Identity Provider Single Sign-On URL	IdP SSO URL
Identity Provider Issuer	IdP Issuer ID
X.509 Certificate	X.509 Certificate

5. In the IdP Attributes Mapping section, enter the attribute names from Okta. The names are case-sensitive and must match exactly.

6. Save your settings.

#### Task 6. Map SAML Group Memberships to Cortex XSOAR User Groups

1. Select Settings & Info → Settings → Access Management → User Groups.
2. Right-click a user group and select Edit Group.
3. In the SAML Group Mapping field add the Okta group(s) that should be associated with this user group. Multiple groups should be separated with a comma. The Okta group name must match the exact value sent in the token.
4. Save your settings.
5. Repeat for each user group.

#### Task 7. Test SSO Login

1. Go to the Cortex XSOAR tenant URL and Sign-In with SSO.
2. After authentication to Okta, you are redirected again to the Cortex XSOAR tenant.
3. When logged in, validate that you have been assigned the proper roles.

To view your role and any role assigned to a user group you are a member of, click your name in the bottom left-hand corner, and click About.

### 6.4.4 | Set up Azure AD as the Identity Provider Using SAML 2.0

This topic provides specific instructions for using Azure AD to authenticate your Cortex XSOAR users. As Azure AD is third-party software, specific procedures, and screenshots may change without notice. We encourage you to also review the Azure AD documentation.

To configure SAML SSO in Cortex XSOAR, you must be a user who can access the Cortex XSOAR tenant and have either the Account Admin or Instance Administrator role assigned.

The following video is a step-by-step guide configuring SSO in Cortex XSOAR (specific Azure AD instructions begin at minute 12:42).



#### Task 1. Configure Azure AD Security Groups

Within Azure AD, assign users to security groups that match the user groups they will belong to in Cortex XSOAR. Users can be assigned to multiple Azure AD groups and receive permissions associated with multiple user groups in Cortex XSOAR. Use an identifying word or phrase, such as Cortex XSOAR, within the group names. For example, Cortex XSOAR Analysts. This allows you to send only relevant group information to Cortex XSOAR, based on a filter you will set in the group attribute statement.

#### Task 2. Copy Single SSO and Audience URI Values from Cortex XSOAR

1. In Cortex XSOAR go to Settings & Info → Settings → Access Management → Authentication Settings.
2. In the Login Options tab, toggle SSO Disabled to on.

By default, SSO is disabled in Cortex XSOAR.

3. Expand the SSO Integration settings.
4. Copy and save the values for Single Sign-On URL and Audience URI (SP Entity ID).

Both values are needed to configure your IdP settings.

You cannot save the enabled SSO Integration at this time, as it requires values from your IdP.

#### Task 3. Configure Cortex XSOAR Application in Azure AD

1. From within Azure AD, create a Cortex XSOAR application and Edit the Basic SAML Configuration.

## Set up Single Sign-On with SAML

An SSO implementation based on federation protocols improves security, reliability, and end user experiences and is easier to implement. Choose SAML single sign-on whenever possible for existing applications that do not use OpenID Connect or OAuth. [Learn more.](#)

Read the [configuration guide](#) for help integrating Cortex XSOAR 8 Production.

**1 Basic SAML Configuration**

Identifier (Entity ID)	<code>https://xxxxxxxxxx.paloaltonetworks.com</code>	<a href="#">Edit</a>
Reply URL (Assertion Consumer Service URL)	<code>https://xxxxxxxxxx.paloaltonetworks.com/idp/saml</code>	
Sign on URL	<code>https://xxxxxxxxxx.paloaltonetworks.com/idp/saml</code>	
Relay State (Optional)	<code>https://xxxxxxxxxx.paloaltonetworks.com</code>	
Logout Url (Optional)	<code>Optional</code>	

**2 Attributes & Claims**

givenname	<code>user.givenname</code>	<a href="#">Edit</a>
surname	<code>user.surname</code>	
emailaddress	<code>user.mail</code>	
name	<code>user.userprincipalname</code>	
memberOf	<code>user.groups</code>	
Unique User Identifier	<code>user.userprincipalname</code>	

- Paste the Single sign-on URL and the Audience URI (SP Entity ID) that you copied from the Cortex XSOAR SSO settings. The Single sign-on URL from Cortex XSOAR should be pasted in the Reply URL and the Sign on URL fields. The Audience URI (SP Entity ID) value from Cortex XSOAR should be pasted in the Identifier (Entity ID) and Relay State fields. This allows users to log in to Cortex XSOAR directly from Azure AD.

**Basic SAML Configuration**

[Save](#) | [Got feedback?](#)

**Identifier (Entity ID)** \* ⓘ

The unique ID that identifies your application to Microsoft Entra ID. This value must be unique across all applications in your Microsoft Entra tenant. The default identifier will be the audience of the SAML response for IDP-initiated SSO.

<code>https://xxxxxxxxxx.com</code>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>
-------------------------------------	-------------------------------------	----------------------

[Add identifier](#)

**Patterns:** `https://samltoolkit.azurewebsites.net`

**Reply URL (Assertion Consumer Service URL)** \* ⓘ

The reply URL is where the application expects to receive the authentication token. This is also referred to as the "Assertion Consumer Service" (ACS) in SAML.

Index	Default
<code>https://xxxxxxxxxx</code>	<input checked="" type="checkbox"/>

[Add reply URL](#)

**Patterns:** `https://samltoolkit.azurewebsites.net/SAML/Consume`

**Sign on URL** \*

Sign on URL is used if you would like to perform service provider-initiated single sign-on. This value is the sign-in page URL for your application. This field is unnecessary if you want to perform identity provider-initiated single sign-on.

<code>https://xxxxxxxxxx.com/</code>	<a href="#">Edit</a>
--------------------------------------	----------------------

**Patterns:** `https://samltoolkit.azurewebsites.net/`

**Relay State (Optional)** ⓘ

The Relay State instructs the application where to redirect users after authentication is completed, and the value is typically a URL or URL path that takes users to a specific location within the application.

<code>https://xxxxxxxxxx.com/</code>
--------------------------------------

3. In the SAML Certificates section, click Edit and verify that Azure is configured to sign both the response and the assertion.

Status	Expiration Date	Thumbprint
Active	8/30/2026, 4:03:21 PM	0120401318F5DC6F084CEFB1E70AD49FA97D276A

**Signing Option:** Sign SAML response and assertion

**Signing Algorithm:** SHA-256

4. To have Azure AD send group membership for the user in the SAML token, you must + Add a group claim in the Attributes & Claims section. Send the Security groups, using the source attribute Group ID. Use the word or phrase you selected when configuring Azure AD security groups (such as Cortex XSOAR) to create a filter. Customize the name of the group claim as memberOf.

**Source attribute:** Group ID

Emit group name for cloud-only groups

Advanced options

Filter groups

**Attribute to match:** Display name

**Match with:** Contains

**String:** Cortex XSOAR

Customize the name of the group claim

**Name (required):** memberOf

**Namespace (optional):**

Emit groups as role claims

Apply regex replace to groups claim content

5. In addition to group membership, verify that there are also claims for:

- Email address
- First Name
- Last Name

#### Task 4. Copy Login URL, Azure ID Identifier, and Attribute Claims

1. In Azure, from the Single sign-on page, in the Set up Cortex XSOAR Production section, copy the values for the Login URL and Azure AD Identifier. You need these values to configure the SSO Integration in Cortex XSOAR.

2. Edit Attributes & Claims and copy the values in the Claim name column. The claim name is case sensitive. You need these values to configure the SSO Integration in Cortex XSOAR.

**NOTE:**

The default attributes shown on the main single sign-on page in Azure AD are not the values you need. You must click Edit next to Attributes and Claims to view and copy the actual values.

Required claim		
Claim name	Type	Value
Unique User Identifier (Name ID)	SAML	user.userprincipalname [...]

Additional claims		
Claim name	Type	Value
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/role	SAML	user.mail
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	SAML	user.givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	SAML	user.userprincipalname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	SAML	user.surname
memberOf	SAML	user.groups

#### Task 5. Download the Certificate

From the SAML Certificates section in Azure AD, Download the Certificate (Base64). You need the contents of this file to configure the Cortex XSOAR SSO Integration.

#### Task 6. Copy the Source IDs for Azure AD Security Groups

The claim for the membership attribute that is sent to Cortex XSOAR uses the Object Id of the group. The Object Id is different from the Azure AD security group name. You can find the Object Id for each of your Azure AD security groups by navigating to Users and groups in Azure AD, clicking on the group name, and viewing the Object id. Create a list of the group names and corresponding Object Ids for every Azure AD security group you want to map to a Cortex XSOAR user group.

#### Task 7. Configure the Cortex XSOAR SSO Integration

1. In Cortex XSOAR go to Settings & Info → Settings → Access Management → Authentication Settings.

2. In the Login Options tab, toggle SSO Disabled to on.

By default, SSO is disabled in Cortex XSOAR.

3. Expand the SSO Integration settings.

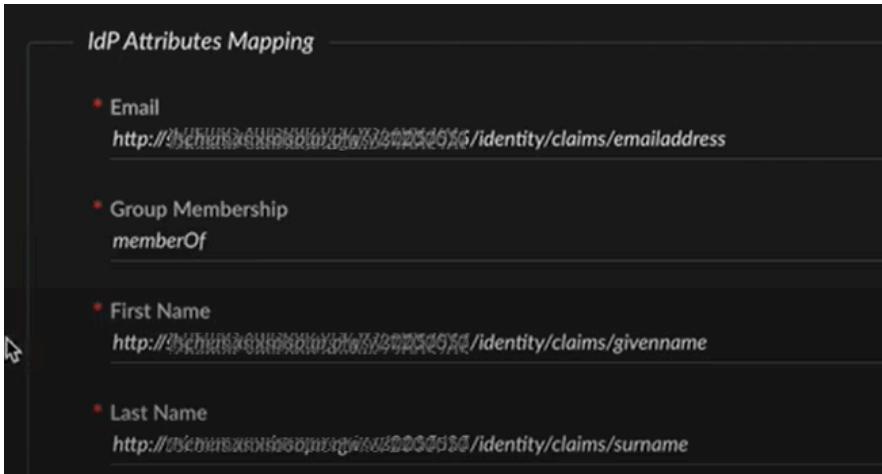
4. Use the following table to complete the SSO Integration settings, based on the values you saved from Azure AD.

Azure AD	Cortex XSOAR Field
Login URL	IdP SSO URL
Azure AD Identifier	IdP Issuer ID
Contents of the downloaded certificate file.	X.509 Certificate

5. In the IdP Attributes Mapping section, enter the attribute claim names from Azure AD. The names are case sensitive and must match exactly.

**NOTE:**

The attribute claim name must exactly match the value sent by your IdP. In some cases, this may be the full attribute name/namespace, depending on the configuration of our IdP



6. (Optional) Under Advanced Settings, select the checkboxes for ADFS and Compress encode URL (ADFS). In some circumstances, these fields may be required by your Azure AD configuration.

7. Save your settings.

#### Task 8. Map SAML Group Memberships to Cortex XSOAR User Groups

1. Select Settings & Info → Settings → Access Management → User Groups.

2. Right-click a user group and select Edit Group.

3. In the SAML Group Mapping field add the Azure AD group(s) Object Ids that should be associated with this user group. Multiple Object Ids should be separated with a comma. The Azure AD group Object Id must match the exact value sent in the token.

4. Save your settings.

5. Repeat for each user group.

#### Task 9. Test SSO Login

1. Go to the Cortex XSOAR tenant URL and Sign-In with SSO.

2. After authentication to Azure AD, you are redirected again to the Cortex XSOAR tenant.

3. When logged in, validate that you have been assigned the proper roles.

To view your role and any role assigned to a user group you are a member of, click your name in the bottom left-hand corner, and click About.

## 6.5 | User management

### Abstract

Invite users to the platform and set user roles and user groups in Cortex XSOAR On-prem.

To access Cortex XSOAR, users must either be added to Cortex XSOAR locally or via SSO. When logging into Cortex XSOAR users must have an assigned role. If no role is assigned either directly or via a user group, they cannot view/edit any data when logging in.

On the Users page, you can view user information, such as user type, role, and user groups.

### User information

Name	Description
User Type	Indicates whether the user was Local (added in Cortex XSOAR), SSO (single sign-on) using your organization's IdP, or both Local/SSO.  For information about enabling SSO in Cortex XSOAR, see <a href="#">Authenticate users using SSO</a> .
Direct Role	Name of the role assigned to the user (not inherited from elsewhere, such as a User Group).
Groups	Lists the user groups to which a user belongs.  Any group imported from Active Directory has the letters AD added beside the group name.  If a user is assigned to multiple user groups, which are mapped to different roles, or if the user is assigned to nested user groups, the user has the highest level of privileges based on the combination of roles.
Group Roles	Lists the different group roles based on the groups to which the user belongs. When you hover over the group role, the group associated with this role is displayed.
Last Login Time	Last date and time the user accessed Cortex XSOAR.
Status	Displays whether the user is Active or Inactive
Phone number	Displays the user's phone number. Including the user's phone number enables playbooks and scripts to trigger direct analyst communication by phone.

### Create users

To add users locally (not SSO), you can either send an invitation to users by adding their details manually or by uploading a CSV file with multiple users. See [Create users in Cortex XSOAR](#).

### Add/update user roles

You can update user roles for one or multiple users. You can add/update the following user roles:

- **Pre-Defined roles:** Instance Administrator and Account Admin.
- **Custom roles:** Includes out-of-the-box roles and roles.

**NOTE:**

To update the permissions attributable to each role, you need to change them in the Roles tab.

1. Go to Settings & Info → Settings → Access Management → Users, and do one of the following:
  - To edit one user, right-click the user's name and select Edit Users Permissions.
  - To edit multiple users, select multiple users, right-click, and select Edit Users Permissions.
2. In the Role field, select one of the pre-defined or custom roles.
3. Add User Groups if required.
4. Save the user role.

#### **NOTE:**

If no role is assigned either directly or via a user group, users do not have view or edit permissions in Cortex XSOAR.

The Show Accumulated Permissions field shows the roles and user groups assigned to the user. You can also select the specific roles assigned to the user, which enables you to compare available permissions based on the roles selected. This can help you understand how the role permissions for a particular user are built. For example, if you need to isolate a specific component, the permissions are provided by a particular role or user group.

#### **Remove a user role**

If a user has a role in the tenant (besides Account Admin), you can remove their user permission to access the tenant. If no direct or user group role has been assigned, the user has no permission to view or edit data in Cortex XSOAR.

1. In the Users tab, right-click the user's name and select Remove User Role.
2. Confirm that you want to Remove the user role.

#### **Unlock users**

If the user's account has been locked, for example, due to too many login attempts, you can unlock the user.

#### **NOTE:**

The user has up to 10 attempts to login before being locked. In any event, the user will be unlocked after 15 minutes.

1. Go to Settings & Info → Settings → Access Management → Users and select the user.
2. Right-click the user and then select Unlock.

The user's status changes to Active.

#### **Deactivate users**

Users should be deactivated to temporarily remove user access to Cortex XSOAR. All user information is maintained for deactivated users. Users should be permanently removed if they no longer need access to Cortex XSOAR.

#### **NOTE:**

When you remove a role, the role associated with the API keys is deleted. When a user is deactivated, API keys that the user created are not revoked.

- If more than one role was associated with the API key, a yellow warning symbol appears next to the API key in the API key table. When you hover over the symbol, a message indicates that some of the roles associated with the API key had been deleted.
- If all roles associated with the API key are removed, a red warning symbol appears next to the API key in the API key table. When you hover over that symbol, a message indicates that the key is no longer usable because it does not have a role associated with it. The API key is still visible in the API table but it cannot be assigned.

If the user is assigned to incidents or tasks or is the owner of a dashboard, these assignments do not automatically change when the user is removed or deactivated. We recommend changing incident and task assignments manually before removing or deactivating users.

Any reports the user has created remain available. Reports are not owned by specific users and can be edited or deleted by other users.

#### **NOTE:**

When you remove a role, the role associated with the API keys is deleted. When a user is deactivated, API keys that the user created are not revoked.

Before you begin:

- Reassign open incidents to another user.

Go to the Incidents page and search for `-status:closed owner:user_name` to find any incidents the user is assigned and reassign.

- Reassign tasks to another user.

Go to the Incidents page and search for `-status:closed investigation.users:user_name` and reassign.

When a user is assigned a task in an incident, the user is added to the incident. This search finds all incidents where the user is a participant.

## How to deactivate users

1. Go to Settings & Info → Settings → Access Management → Users and select the user.
2. Right-click the user and then select Deactivate User and then Deactivate to confirm.

## Delete users

In Cortex XSOAR, you can permanently remove a user, or temporarily disable a user. Users should be permanently removed if they no longer need access to the system.

### NOTE:

You cannot deactivate or delete a user that has an Account Admin role.

When you delete users, all their personal information is deleted, including email addresses, usernames, phone numbers, and first and last names.

Before you begin:

- Reassign open incidents to another user.

Go to the Incidents page and search for `-status:closed owner:user_name` to find any incidents the user is assigned and reassign.

- Reassign tasks to another user.

Go to the Incidents page and search for `-status:closed investigation.users:user_name` and reassign.

When a user is assigned a task in an incident, the user is added to the incident. This search finds all incidents where the user is a participant.

## How to delete users

1. Go to Settings & Info → Settings → Access Management → Users and select the user.
2. Right-click the user and then select Delete User and then Delete to confirm.

### NOTE:

You can also delete a Single Sign-on (SSO) user. This option is only available when you've enabled SSO in Cortex XSOAR.

## 6.6 | Configure a password policy

### Abstract

Configure and edit the Cortex XSOAR password policy

To define your password policy, go to Settings & Configurations → Settings → Access Management → Password Policy.

You can define your password policy using the following parameters:

- Minimum number of characters
- Minimum number of lowercase letters
- Minimum number of uppercase letters
- Minimum numbers of digits or symbols

In addition, you can require users to change their password every X number of days or months. By default, this setting is not enabled and passwords do not automatically expire. You can also prevent users from reusing previous passwords.

The lock settings enable you to lock a user out of Cortex XSOAR after a set number of failed login attempts within one minute. You can either have the user automatically unlocked after a set number of minutes or hours, or you can only allow the user to be unlocked by an administrator. To unlock a user, go to Settings & Configurations → Settings → Access Management → Users, right-click on the username, and select Unlock.

Users can change their passwords by clicking the username at the bottom of left hand main menu and selecting User Preferences → Details.

## 7 | Marketplace

### Abstract

In Marketplace, download your content packs to suit your use case in Cortex XSOAR.

Marketplace is a centralized content portal enabling you to manage content in Cortex XSOAR. Content is organized into Content Packs created by different contributors such as Palo Alto Networks, Partners, and MSSPs. Download a content pack to suit your use case.

## 7.1 | Cortex Marketplace

### Abstract

Search the Cortex Marketplace and find content. Search by use cases, integrations, and categories.

Cortex XSOAR Marketplace is the premier digital storefront for discovering, exchanging, and contributing security automation playbooks, built into Cortex XSOAR. Cortex XSOAR content packs are prebuilt bundles of integrations, playbooks, dashboards, fields, subscription services, and all the dependencies needed to support specific security orchestration use cases.

Marketplace enables you to:

- **Leverage content from the largest SOAR community:** Continuously extend Cortex XSOAR with proven use cases contributed by SecOps users and SOAR partners.
- **Discover top-rated, validated content:** Identify the content offerings recommended by your peers and validated by the world's leading cybersecurity company. Discover how to increase automation with the tools that you already have.
- **Solve your toughest security use cases:** Deploy turnkey security workflows that span integrations, playbooks, dashboard layouts, and reports with a single click.

Marketplace enables you to build a strong community with other security professionals by exchanging content. You can explore the latest trends from Cortex XSOAR and other contributors and test drive use cases all within your Cortex XSOAR platform.

Cortex XSOAR supports free content packs, which are either Cortex XSOAR, or partner-supported content packs. You can restrict a user role from managing content packs in Marketplace when defining/editing user roles.

In Marketplace, you can browse all content packs (including installed content), or view only installed content packs.

You can search for content packs by entering text in the search bar and selecting the relevant content pack from the search results.

You can sort content packs by latest update, best match, recommended, number of downloads, and filter according to the following criteria:

- **Use cases:** Filter according to high-level use cases, such as Phishing, Malware, Ransomware, Access.
- **Integrations:** Filter according to the integration included in the content pack.
- **Categories:** Filter according to content pack categories, such as Messaging, and Forensics & Malware Analysis
- **Published:** Filter according to whether published by Cortex XSOAR or by Cortex XSOAR technology partners.
- **General:**
  - **Certified:** Created and supported by a user and certified by Cortex XSOAR. Cortex XSOAR has tested the content to ensure that it meets standards and works correctly.
  - **Support:** Supported by either Cortex XSOAR or a partner-supported content pack.
  - **Uses my integrations:** Content packs that use integrations that you have added instances for (whether or not they are enabled).
- **Content Pack Includes:** Filter according to the content of the content pack, such as scripts, Integrations, and Playbooks.
- **Tags:** Filter according to tags, such as Alerts, Network, and Security.
- **Types:** Filter according to Collection or TIM.

When clicking a content pack you can view detailed information including content that it installs (such as scripts and playbooks, and indicator fields), dependencies (what content packs are required or optional) and version history (including whether you want to roll back to earlier versions).

## 7.2 | Content packs

### Abstract

Download content packs in Marketplace for your use case.

Cortex XSOAR content in Marketplace is organized in packs. Content packs are created by Palo Alto Networks, technology partners, consulting companies, MSSPs, customers, and individual contributors. Content packs may include a variety of different components, such as integrations, scripts, playbooks, and widgets, grouped together to address a specific use case. Content packs are free and can be used by all customers.

### Pre-installed content packs

Cortex XSOAR comes with a number of pre-installed content packs that cover many common uses cases. Pre-installed content packs include, but are not limited to:

- Common Scripts, Common Widgets, Common Playbooks, Common Types, Common Reports, Common Dashboards

These content packs provide important tools and building blocks you can use to customize your playbooks and workflows in Cortex XSOAR. The Common Scripts content pack, for example, includes scripts that convert file formats, fetch indicators from a file, export context data, send emails, and more.

- VirusTotal

Provides integration with the popular Virus Total service to analyze suspicious files, domains, IPs and URLs to detect malware and other security breaches.

- TIM - Indicator Auto-Processing

The TIM - Indicator Auto-Processing content pack includes playbooks that automate the processing of indicators for multiple use cases such as tagging, checking for existence in various lists , running enrichment for specific indicators and preparing indicators if necessary for a manual review. The content pack also includes incident types and incident layouts for manual review.

#### Recommended content packs

In addition, we recommend reviewing if you require the following popular content packs:

## Suggested Use Cases to start with



- Phishing

Create and respond to phishing incidents based on user reports.

- Cortex XDR by Palo Alto Networks

Automate Cortex XDR incident response. Includes custom Cortex XDR incident views and layouts to aid analyst investigations.

- ServiceNow

Manage ServiceNow tickets directly from the Cortex XSOAR and enrich them with Cortex XSOAR data.

- PAN-OS by Palo Alto Networks

Manage Palo Alto Networks Firewall and Panorama, from Cortex XSOAR.

- Integrations & Incidents Health Check

Review failed integrations, incidents, and playbooks.

- A mail sender integration, such as Microsoft Exchange Online.

- A collaboration integration, such as Microsoft Teams or Slack to send messages and notifications to your team.

Content packs such as the Malware Investigation and Response content pack and the Phishing content pack include a deployment wizard. When you install the content pack, you are prompted to use a wizard, which sets up your use case. The deployment wizard sets up the fetching integration, configures the playbook and parameters, and configures supporting integrations, in a user friendly, step-by-step interface.

## 7.3 | Manage content packs

### Abstract

Install, delete, update, and revert content packs.

You can install, delete, update, and revert content packs. Before you install a content pack you should review the content pack to see what it includes and what are the various dependencies. Following is the information you can view:

- **Details:** General information about the content pack such as installation, content, version, author, and status.
- **Content:** The content to be installed, such as scripts or integrations.
- **Dependencies:** Details of any required content packs and optional content packs that may need to be installed with your content pack.
- **Version History:** View the currently installed version, earlier versions, available updates, and revert if required.

### Dependencies

In Cortex XSOAR content packs, some objects are dependent on other objects. For example, an alert may be dependent on a playbook, an alert type, and an alert field. A script may be dependent on another script, or an integration. When you place a content pack in your cart, mandatory dependencies including required content packs are added automatically to ensure that the content pack installs correctly.

Optional content packs are used by the content pack you want to install but are not necessary for installation. When you place a content pack in your cart, you can choose which optional content pack to install. When you install optional content packs, mandatory dependencies in the optional content pack are automatically included.

#### NOTE:

Optional content packs that are already installed are treated like they are required content packs to preserve content integrity.

### Install a content pack

You can only install one content pack at a time. Cortex XSOAR automatically adds any content that is required to install the content pack. You can also add any optional content packs that use the content pack you want to install.

If you receive an error message when you try to install a content pack, you need to fix the error before installing. If a warning message is issued, you can still download the content pack, but you should fix the problem otherwise the content may not work correctly.

#### NOTE:

You can use the Deployment Wizard to significantly reduce the time to set up your use case, for example for **Malware Investigation and Response** or **Phishing**.

1. Go to Marketplace → Browse and locate the content pack you want to install.
2. Click the required content pack and review the contents.
3. Click Install to add the content pack to the Cart.
4. (Optional) If the content pack includes optional content, select the content packs you want to add.

The Cart displays the number of items you are installing including any required content packs. You can log in and out, but the content packs remain in the Cart until you click either Empty cart or Install.

5. Click Install.

6. After installation, click Refresh content.

### Delete a content pack

When you delete a content pack, all content is deleted including all detached and customized content.

#### CAUTION:

If another content pack is dependent on the content pack you want to delete, it may break the other content pack. You can reinstall the content pack, but you cannot restore detached and customized content.

#### NOTE:

After you delete a content pack, it is recorded in the audit log. The version appears in the installation/update entry.

1. Go to Marketplace → Installed Content Packs.
2. In the Content Packs Library section, search for the content pack and select the content pack you want to delete.
3. Click the trash can icon.
4. Review the warning message and click Delete.

## Update a content pack

Content packs are updated for bug fixes, enhancements, and more. Marketplace is updated every 2 hours and when there is an update available for a content pack, you will see a notification in the Installed Content Packs tab in Marketplace.

In the Version History tab of a content pack, you can see the currently installed version, earlier versions, and available updates. You can revert to a previous version of a content pack if required.

If you have made any customizations, these are automatically included in any update. All dependent content packs update automatically with the content pack.

### NOTE:

Third party product Integrations are developed and tested against a specific product version. For products which are on-prem or cloud based with specific API versions, the version developed and tested against will be included in the integration's documentation. Newer versions of the product are not always immediately tested and it is expected that products maintain API compatibility upon release of newer product versions. When upgrading to a newer product version, it is highly recommended to test the integration in a dev environment before deploying to production.

### CAUTION:

If you want to downgrade, any content that depends on the content pack including any customizations may be deleted if it does not exist in the target content pack version.

1. In the Show field of the Installed Content Packs tab, select Update available to display the content packs that are available to update.
2. Click the content pack you want to update.
3. In the Version History tab of the content pack, view the updates that are available.
4. Click Update. If there is more than one update available, click the version to update.

If you choose to install the latest version it includes the previous version. If you have made any customizations these are included in any update. If any dependencies require updating, these are automatically added.

5. Click Install.
6. After the content pack installs, click Refresh content.

## Revert a content pack

You can revert to an earlier version of an installed content pack. Items that are not included in the version are also deleted, such as detached playbooks, or scripts that use other scripts from the content pack. This may cause other content packs to stop working

1. In the Installed Content Packs tab, click the content pack you want to revert.
2. In the Version History tab, select the version to which you want to revert.
3. Click Revert to this version. The version will be added to your Cart.
4. In the Cart, click Downgrade.

## 7.4 | Set up your use case with the Deployment Wizard

### Abstract

The Deployment Wizard guides you step-by-step to quickly adopt your use case.

The Deployment Wizard can be used to set up your use case for the **Malware Investigation and Response** content pack and the **Phishing** content pack. In order to work with your content pack you need to set up your integrations. The Deployment Wizard guides you through:

- Configuring the integrations that will be used to fetch events (fetching integrations). These events will be mapped as incidents.
- Configuring the main playbook and its input parameters. For example, the Setup Malware playbook pane opens showing the recommended primary playbook for the incident type you selected when configuring the fetching integration. The playbook configuration includes all the input parameters to configure that will change the playbook behavior, for example, whether to use sandbox detonation or whether to perform isolation response. You can open the playbook by clicking the link on the bottom.
- Configuring any supporting integrations. such as an email integration

The default fetching integration for your content pack depends on which fetching integration(s) are installed. For example:

Content Pack	Default Fetching Integration In Order Of Priority
Malware Investigation and Response	1. Palo Alto Networks Cortex XDR - Investigation and Response 2. CrowdStrike Falcon 3. Microsoft Defender for Endpoint
Phishing	1. Gmail 2. EWS v2 (Make sure you also install the Microsoft Exchange On-Premise pack) 3. O365 Outlook Mail (Using Graph API) 4. Gmail Single User 5. O365 Outlook Mail Single User (Using Graph API)

## Prerequisites

To access the Deployment Wizard for the first time, you need to first install or update your Malware Investigation and Response content pack or your Phishing content pack in Marketplace. The Deployment Wizard tab appears in Marketplace after the content pack installation or update is completed.

### For example:

- For the Malware Investigation and Response content pack, you need at least one incident fetching content pack (mandatory). You can also optionally install sandbox, messaging, case management, and data enrichment and threat intelligence content packs.
- For the Phishing content pack, you need at least one email gateway content pack (mandatory). You can also optionally install sandbox, EDR systems, network devices, email security gateways, mail sender, and data enrichment and threat intelligence content packs.

### How to set up your use case with the Deployment Wizard

- In Marketplace, select the content pack for your use case (for example, Malware Investigation and Response or Phishing) and click Install or Update (if the pack is already installed).
- In the Select Content Packs window, select one or more content packs from the required categories. You can also install other supportive content packs from other categories if needed. These items will be automatically be added to the cart.
- Click Continue and then Install or Update.
- When the content pack finishes installing or updating, click Refresh content.

The Deployment Wizard tab appears.

#### NOTE:

After you start running your use case you can return to this tab and make changes to the configurations, such as your integration's credentials or playbook parameters.

- Click Let's Start in the small dialog box that appears next to the Deployment Wizard tab.

The tab opens showing the use case deployment flow.

- Step 1: Fetching Integration - Click the displayed fetching integration. If the integration is new, select New instance. If you want to use an existing instance, select it from Update existing instance. The integration will stay disabled until you complete all steps of the wizard.

#### NOTE:

You must define the incident type in order to set the playbook in the next step.

A list of What needs to be done guides you through the required fetching integration instance settings configurations. Scroll down to see the complete list.

After you save your settings, the wizard initiates a test connection. If the connection succeeds, the Fetching Integration step turns green and moves to the next step (Set Playbook).

- Step 2: Set Playbook - Select Configure Playbook & Parameters.

#### NOTE:

The wizard displays the recommended playbook. If for the fetching integration setup you chose an incident type that uses a different playbook from the recommended one, the incident type will be detached.

8. Click Done.

9. Step 3: Supporting Integrations - Configure any installed supporting integrations in the content pack.

If a supporting integration is already installed and connected, it appears with a green check. Otherwise, click the integration to configure it.

**NOTE:**

After you save the settings, the integration instance is automatically enabled.

10. Step 4: What's Next - Select Turn on Use Case to start the fetching process and running the playbooks and scripts.

## 7.5 | Marketplace FAQs

### Abstract

Frequently asked questions about Cortex XSOAR Marketplace Content

#### Should Marketplace content always be updated?

Marketplace updates are a source for bug fixes and provide new commands for integrations and scripts. It's a best practice to update content packs to the newest available version. If you encounter any issue with content updates, you can revert to a previous version with one click.

#### When can Marketplace content be updated?

You can update content while the system is in use. If a playbook, for example, is running on an incident while you update that playbook, the original version of the playbook will continue to run without issues. If the playbook includes an integration command that has been updated, and the update occurs before the playbook reaches this task, the new version of the integration command will be used.

#### When should content items be duplicated versus detached?

To edit a content item, the item must be detached or be custom content. When content items are detached, they do not receive updates from Marketplace. There are two options for editing content items:

- Detach content item (such as playbooks, and automations) and edit the content item. If you want to receive content updates in the future, you can reattach the content item, but the modifications you made while the item was detached will be overwritten with the content update.
- Duplicate the content item and edit the copy. When a content item is duplicated it becomes a custom content item, and therefore will not receive updates, but you can view updates to the original content item.

#### How does Marketplace content differ from custom content?

After Marketplace content is installed you can detach or duplicate the content and customize the content as needed. Custom content is, by definition, detached and does not receive updates.

#### How can content updates be rolled back? Are dependencies automatically rolled back as well?

You can view all versions of a content pack in Marketplace and revert to earlier versions there. When you revert a content pack, only the content pack is reverted, not the pack dependencies.

## 7.6 | Content pack update notifications

### Abstract

Enable update notifications for individual content packs.

You can receive daily notifications of content packs that have available updates. When you enable notifications, they are sent only to you via email, Slack, or another notification service, depending on your user settings. Notifications are enabled and disabled on a per content pack basis.

#### Enable content pack update notifications

You enable content pack update notifications only after the content pack is installed.

1. Go to Marketplace → Installed Content Packs,
2. Search for the content pack.
3. Click  (on the upper right).
4. To cancel, click the icon again.

**NOTE:**

You can also disable notifications for individual content packs by clicking Stop Notifications in the daily email.

### View content packs with notifications enabled

To view a list of all content packs with notifications, go to Marketplace → Installed Content Packs and in the Show field, select the Notifications active filter.

### Opt out of notifications

You can temporarily or permanently opt out of notifications for all content packs, without disabling the notification option for individual content packs. The content packs still appear in the Notifications active list.

1. Click your username (in the Cortex XSOAR side menu) and click User Preferences.
2. In the Notifications tab, expand the option for Other Notifications.
3. In the Marketplace Content Packs Updates field, select or clear the checkbox to enable or disable notifications.

#### 7.6.1 | Customize content pack notifications

##### Abstract

Customize the frequency and time of content pack update notifications and how much information is included.

You can decide whether to enable notification of new content updates and whether to notify users by email.

##### Notify users for content pack updates

By default, updates for content packs are not sent out to users. You can change the default and add users as required.

1. Select Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.
2. Add the following:

Key	Value
<code>content.notification.enabled</code>	Set to true to enable notification for new content updates.
<code>content.notification.users</code>	Notifies all users by email when there is a content update available (comma separated user names in Cortex XSOAR).

## 7.7 | Content pack contributions

##### Abstract

You can create content packs for submission to the Cortex XSOAR Marketplace.

Contributions are content packs that you create for Cortex XSOAR Marketplace, which are submitted to Cortex XSOAR for review and approval. After approval, these content packs are uploaded to Marketplace, and are shared and installed like any other content pack. When creating new content such as playbooks, scripts, incident types, and integrations, or when updating content, you can:

- Create and submit content directly from Cortex XSOAR. For example, from a playbook, click Contribute. You then have the option to submit the contribution for review or download the contribution and upload it, for example, to GitHub.
- Submit a content pack of one or more items through the Cortex XSOAR Marketplace UI. When you create or edit content in Cortex XSOAR, that content is added to the Add Content section in the Contributions tab in Marketplace. You can add content from this list to a content pack. From the Contributions tab in Cortex XSOAR Marketplace, you can create, edit, submit, and delete content that you have submitted through Marketplace.
- Create a GitHub pull request on the public XSOAR Content Repository.

Users with the Contribute to Marketplace permission can contribute content packs to Marketplace.

When adding content to the content pack, Cortex XSOAR scans the content and automatically adds dependencies, which ensures that the content pack installs and runs correctly on all environments.

Although Cortex XSOAR scans and tests the content to ensure it works correctly, you need to review the content to ensure that all dependencies are incorporated and work as they should in the event that not all dependencies are added automatically.

## Validation

Content validation enables users to improve the quality of the content they develop in Cortex XSOAR by running a script to check for errors before submission.

## Configuration

By default, content validation passes your content item(s) as inputs to the ValidateContent script included in the Base pack. The ValidateContent script uses the `demisto-sdk` utility to run `validate` and `lint` on the content item(s) and returns the results.

## Automatic

When contributing content, either from the Contributions page, the Contribution Pack Editor page or directly from a content item's menu, the content goes through content validation before submission. After clicking Contribute, you have the option to Save and submit your contribution or Save and download your contribution. In both cases, your contribution goes through validation before you submit or download the content.

If the content pack passes validation, the process continues. If you are downloading the content, a download will start automatically. If you are submitting the content, the content will submit automatically. If the content pack does not pass validation, the validation issues are listed and you have the option to export a raw JSON file with the error details. You can then make changes to your content items and resubmit for validation.

You also have the option to skip the validation step or to contribute a content pack that does not pass validation. For example, there might be an issue you are aware of that cannot yet be resolved. For a large content pack, where you have already validated the individual content items, you might want to skip the final validation as it can be a lengthy process for a large content pack.

Validation Results - Dedup - Generic v3	
Playbook	
Content Item	Error Message
Dedup - Generic v3	The following tasks ids have no previous tasks: ["18"]

**Buttons:**

- Export as raw json
- Contribute as is
- Cancel

You can also manually trigger content validation. The Validate button appears in the Contribution page, the Contribution Pack Editor page, as well as in both the Script and Integration Editors. With manual validation, you can check your content during the development process and make changes.

## Review process

The review process consists of the Cortex XSOAR team checking that your contribution meets code, documentation, naming, and other standards. You receive a form to complete asking for more information, such as certification, contact details, etc. The Cortex XSOAR team will be in touch with you during the review process.

During the review process you may be asked to make changes in the code, or for more data, metadata, dependencies, documentation, support and certification model, etc. You can anonymize your name if required.

When your contribution is approved it is uploaded to Marketplace where other Cortex XSOAR users can view, download, and rate it. We encourage you to learn more about the contribution process.

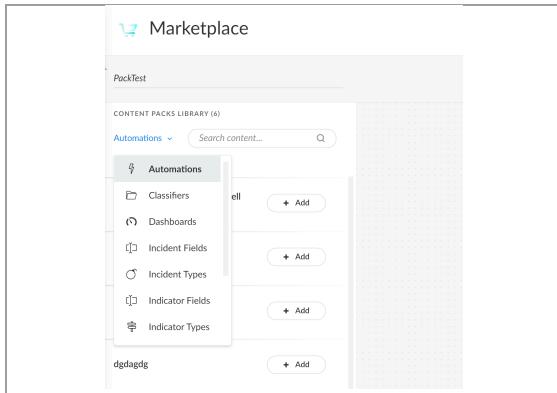
### 7.7.1 | Create a content pack

#### Abstract

Create a content pack and submit it to Cortex XSOAR for approval. Add your content pack to Marketplace.

Any user can add content that has been created to a content pack. The content pack is submitted and reviewed by Cortex XSOAR to ensure it complies with Cortex XSOAR standards. After approval, the content pack can be used in Marketplace.

1. Select Marketplace → Contributions → Contribute Content.
2. In the Pack Name field type a meaningful name for the content pack.
3. From the list select the type of content you want, locate the content you want to add and click Add.



For more information about how to build a content pack, see the Dev Hub documentation.

4. If you want to continue adding content at a later time or to use the Validate Pack option, click the save button.
5. (Optional) Click Validate Pack to check for errors. The pack must be saved before you validate.
6. If you have finished and want to send the content pack to a Cortex XSOAR developer for review, click Save and Contribute to either contribute or download your contribution.
7. (Optional) In the Contribute dialog box, add a description.
8. Select Save and submit your contribution, enter your email address and click Contribute.

**NOTE:**

Instead of submitting the contribution, you can also download the content pack and upload it, for example, to GitHub.

9. After you contribute the content pack, you will receive an email with a link to a form you must complete.
  - a. Fill in additional details, such as pack name, author, and description.
  - b. Log in to your GitHub account to participate in the review process of the pull request that is automatically opened for your content pack.
  - c. Sign the Palo Alto Networks Contributor License Agreement.
- If you are updating an existing content pack:
10. If you are updating an existing content pack:
  - a. Select Update Existing Pack from the Select Contribution Mode menu.
  - b. Select the pack you want to update from the Select Existing Pack menu.
  - c. Log in to your GitHub account to participate in the review process of the pull request that is automatically opened for the content pack.
  - d. Sign the Palo Alto Networks Contributor License Agreement.
  - e. Given a version number MAJOR.MINOR.REVISION, select Update Type.

Type	Description
Revision	When you make backwards compatible bug fixes.
Minor	When you add functionality in a backwards compatible manner.
Major	A major revision contains one or more of the following: <ul style="list-style-type: none"> <li>• Incompatible API changes</li> <li>• New features</li> <li>• Changes to existing features that could break backward compatibility</li> </ul>
Documentation	When only documentation files are updated.

11. After you submit the form, a GitHub branch is created in the xsoar-contrib content repository fork with the changes from your contribution.
12. You will receive an invitation to join the xsoar-contrib organization. Being a member of the organization enables the xsoar-bot to invite you to a GitHub team and grant you write permissions to the created branch. (Each contributor can only modify files in content packs that they contributed).
13. The pull request is created and a reviewer is assigned.

**NOTE:**

The documentation for new integrations/scripts/playbooks is automatically generated and contains basic information. You need to review the documentation file, README.md, and modify it according to Cortex XSOAR standards. The files to be reviewed are listed at the pull request comment.

14. You can now modify the files changed in the pull request as part of the review process.

### 7.7.2 | Resubmit a content pack

#### Abstract

Resubmit an existing content pack with new changes from the Cortex XSOAR UI.

If you have already submitted a content contribution, and you need to make changes to the submission, you can do so by resubmitting the content pack from Cortex XSOAR. The resubmission process is similar to the initial submission, but you are updating an existing pull request instead of creating a new one.

**NOTE:**

You cannot update JavaScript integrations or scripts in an existing content pack using this method.

1. Create or edit any content items that need to be included in your resubmission.
  2. Go to Marketplace → Contributions, select your pack, and Edit the pack.
  3. Add or remove content items from the pack as needed.
  4. Save and Contribute.
  5. After you submit the content pack, you will receive an email with a link to a form you must complete.
    - a. Select Update Existing Pack from the Select Contribution Mode menu.
- NOTE:**
- The contribution mode option only appears if content items that are part of your contribution are detected as originating from existing sources. For example, if you created a new automation in the UI by clicking Duplicate Automation.
- b. Select the pack you want to update from the Select Existing Pack menu.
  - c. Log in to your GitHub account to participate in the review process of the pull request that is automatically opened for the content pack.
  - d. Sign the Palo Alto Networks Contributor License Agreement.
  - e. Given a version number MAJOR.MINOR.REVISION, select Update Type.

Type	Description
Revision	When you make backwards compatible bug fixes.
Minor	When you add functionality in a backwards compatible manner.
Major	A major revision contains one or more of the following: A major revision contains one or more of the following: <ul style="list-style-type: none"> <li>• Incompatible API changes</li> <li>• New features</li> <li>• Changes to existing features that could break backward compatibility</li> </ul>
Documentation	When only documentation files are updated.

**NOTE:**

Changing the pack name or the email address of the contributor at this stage will result in creating a new pull request on GitHub, rather than updating an existing pull request.

In the form, you can include notes describing the update, or provide an updated demo video link which will be displayed in a comment on the pull request after the changes are successfully pushed to GitHub.

6. After the changes are pushed to your branch, you will receive an email notification.

In addition to the resubmission process described above, there are other ways to modify your existing content pack. You can modify the files directly in the GitHub pull request, or close the pull request and create a new contribution that includes your changes. We do not recommend closing the pull request and creating a new request.

## 8 | Integrations

### Abstract

Configure integrations, manage credentials, run commands, and troubleshoot integrations in Cortex XSOAR On-prem

integrations seamlessly connect your security and incident management tools right within Cortex XSOAR. Easily configure them to streamline your workflow: fetch incidents, execute commands, manage credentials, and seamlessly handle long-running integrations.

### 8.1 | Integration use cases

#### Abstract

Common integration use cases for Cortex XSOAR, including analytics and SIEM, authentication, case management, data enrichment, threat intelligence, forensic and malware,

The following categories are common use cases for Cortex XSOAR integrations. While this list is not meant to be exhaustive, it's a starting point to understand what use cases are supported by Cortex XSOAR and third-party integrations.

#### Analytics and SIEM

Top use cases:

- Fetch incidents with relevant filters.
- Create, close, and delete incidents/events/cases.
- Update incidents - update status, assignees, severity, SLA, and more.
- Get events related to an incident/case for enrichment/investigation purposes.
- Query SIEM (consider aggregating logs).

These integrations usually include the Fetch Incidents option for an instance. It can also include `list-incidents` or `get-incident` as integration commands, or important information for an event or incident.

Analytics & SIEM integration Example: ArcSight ESM

#### Authentication and Identity Management

Top use cases:

- Use credentials from the authentication vault to configure instances in Cortex XSOAR. (Save credentials in: Settings & Info → Settings → Integrations → Credentials.) Integrations that use credentials from the vault should have the Switch to credentials option.
- Lock/Delete Account – Use an integration to lock/unlock a third-party account.
- Reset Account - Perform a reset password command for a third-party account.
- Lock an external credentials vault - in case of an emergency (if the vault has been compromised), allow the option to lock/unlock the entire vault via an integration.
- Step-Up authentication - Enforce Multi-Factor Authentication for an account.
- Create, update, and delete users.
- Manage user groups.
- Block users, force change of passwords.
- Manage access to resources and applications.
- Create, update, and delete roles.

Authentication integration example: CyberArk AIM v2 (Partner Contribution)

## Case Management

Top use cases:

- Create, get, edit, close a ticket or issue, and add and view comments.
- Assign a ticket/issue to a specified user.
- List all tickets, and filter by name, date, and assignee.
- Get details about a managed object, update, create, delete.
- Add and manage users.

Case Management/Ticketing integration example: ServiceNow V2

## Data Management & Threat Intelligence

Top use cases:

- Enrich information about different IOC types: Upload object for scan and get the scan results. (If there's an option to upload private/public, the default should be set to private.) Search for former scan results about an object to get information about a sample without uploading it yourself. Enrich information and scoring for the object.
- Add indicators to the system and search for existing indicators.
- Add indicators to the exclusion list.
- Calculate DBot Score for indicators.
- Enrich asset – get vulnerability information for an asset (or a group of assets) in the organization.
- Generate/trigger a scan on specified assets.
- Get a scan report including vulnerability information for a specified scan and export it.
- Get details for a specified vulnerability.
- Scan assets for a specific vulnerability.

Data Enrichment & Threat Intelligence integration example: Unit 42 Objects Feed.

## Email

Top use cases:

- Get message – download the email itself, retrieve metadata, body.
- Download attachments for a given message.
- Manage senders – block/allow specified mail senders.
- Manage URLs – block/allow the sending of specified URLs.
- Encode/decode URLs in messages
- Release a held message when a gateway has placed a suspicious message on hold.

Email Gateway integration example: MimeCast v2

#### Endpoint

Top use cases:

- Fetch incidents & events
- Get event details (from a specified incident)
- Quarantine a file
- Isolate and contain endpoints
- Update indicators (for example, network and hashes) by policy (can be block, monitor) – deny list
- Add indicators to the exclusion list
- Search for indicators in the system (Seen indicators and related incidents/events)
- Download file based on hash, and path
- Trigger scans on specified hosts
- Update .DAT files for signatures and compare existing .DAT file to the newest one on the Cortex XSOAR tenant
- Get information for a specified host (OS, users, addresses, hostname)
- Get policy information and assign policies to endpoints

Endpoint integration example: Palo Alto Networks Cortex XDR - Investigation and Response

#### Forensics and Malware Analysis

Top use cases:

- Submit a file and get a report (detonation)
- Submit a URL and get a report (detonation)
- Search for past analysis (input being a hash/URL)
- Retrieve a PCAP file
- Retrieve screenshots taken during analysis

Forensic and Malware Analysis example: Cuckoo Sandbox

#### Network Security

Top use cases:

- Create block/accept policies (source, destination, port), for IP addresses and domains
- Add addresses and ports (services) to predefined groups, create groups, and more
- Support custom URL categories
- Fetch network logs for a specific address for a configurable time frame
- URL filtering categorization change request
- Built-in blocked rule command for fast-blocking
- If there is a Management Firewall allow the option to manage policy rules through it
- Get/fetch alerts
- Get PCAP file, packet
- Get network logs filtered by time range, IP addresses, ports, and more
- Create/manage/delete policies and rules
- Update signatures from an online source / upload + get last signature update information
- Install policy (if existing)

Network Security Firewall integration examples: Tufin (Partner Contribution), Protectwise

#### Vulnerability Management

Top use cases:

- Enrich asset – get vulnerability information for an asset (or a group of assets) in the organization.
- Generate/trigger a scan on specified assets
- Get a scan report including vulnerability information for a specified scan and export it
- Get details for a specified vulnerability
- Scan assets for a specific vulnerability

Vulnerability Management integration example: Tenable.sc

## 8.2 | Configure integrations

### Abstract

Configure an integration including creating your own integration

Integrations are mechanisms through which Cortex XSOAR connects and communicates with other products. These integrations can be executed through REST APIs, webhooks, and other techniques. Integrations enable you to orchestrate and automate SOC operations.

Integrations can be one-way or two-way. Two-way integrations allow both systems to interact directly, making it easier to manage security operations across multiple tools.

### Integrations installed from a content pack

Integrations are included in content packs which you download and install from Marketplace. After you download and install a content pack that includes an integration, you need to configure the integration by adding an instance. You can have multiple instances of an integration, for example, to connect to different environments. Additionally, if you are an MSSP and have multiple tenants, you could configure a separate instance for each tenant.

Cortex XSOAR comes out-of-the-box with several integrations to help you onboard, such as:

- Mail Sender

Sends email notifications to users. By default, this integration is configured to send emails. You can change the main sender by configuring a different mail sender, such as Gmail. For more information, see Configure notifications in Cortex XSOAR.

- Generic Export Indicators Service

Provides an endpoint with a list of indicators as a service for the system indicators. For more information about how to set up the integration, see Export indicators using the Generic Export Indicators Integration.

- Palo Alto Networks WildFire Reports

Generates a Palo Alto Networks WildFire PDF report. For more information, see Palo Alto Networks WildFire Reports.

- Rasterize

Converts URLs, PDF files, and emails to an image file or PDF file. For more information, see Rasterize.

## Create an integration

You can create an integration, by adding parameters, commands, arguments, and outputs as well as writing the necessary integration code. You should have a working Cortex XSOAR tenant and programming experience with Python.

To create an integration, on the Instances page, click BYOI.

The screenshot shows the 'Instances' page in the Cortex XSOAR IDE. At the top right, there is a black button with a white plus sign and the text 'BYOI'. This button is highlighted with a yellow rectangular box. Below the button is a search bar containing the placeholder 'Search integration...'. To the right of the search bar is a magnifying glass icon and a checkbox labeled 'Show Deprecated'. The main area of the page displays a table of integration instances, with the first row visible.

The Cortex XSOAR IDE and the HelloWorld integration template are loaded by default. For more information about how to create an integration including an example, see Create an Integration.

## Configure an integration

On the Instance integration page, after you have either downloaded the integration or created an integration, you can do the following:

Option	Description
Add instance	Configure an integration instance to connect and communicate with other products. For more information, see Add an integration instance.  After configuring the instance, you can also enable/disable the integration instance, copy the instance, and view the integration fetch history.
View Integration's source	View the integration settings and source code.
Edit integration's source	Edit the integration settings and source code. For more information about editing the integration's source code, see Create an Integration.  <b>NOTE:</b> If the integration was installed from a content pack you need to duplicate the integration before editing.
Duplicate integration	If you want to change the source code, and settings, or download the integration, you need to duplicate the integration.
Delete	Although you can't delete an integration installed from a content pack (unless a duplicate), you can delete an integration instance.

Option	Description
Download the integration	<p>Download the integration in YAML format. You can also upload an integration.</p> <p><b>NOTE:</b> If the integration was installed from a content pack you need to duplicate the integration before downloading.</p>
Version History	If the integration is a duplicate or you create your integration, you can see the changes in the integration.
Contribute to Marketplace	You can send the integration to Palo Alto Networks for review and for it to be added to Marketplace. For more information, see Content pack contributions.

You can view all the integration changes (the last 100 changes) by clicking the Version History button.

## 8.3 | Change the Docker image in an integration or script

### Abstract

Use Docker to run Python scripts and integrations in a controlled environment in Cortex XSOAR.

Docker enables you to run scripts and integrations from an image in a controlled environment that isolates and safeguards the server. It also simplifies environment setup by packaging dependencies and configurations within an image, ensuring consistent execution across different systems. By default, Cortex XSOAR pulls images from the Demisto Docker image registry in Github, which are used in scripts and integrations as needed. Cortex XSOAR integrations and scripts have the relevant Docker image already selected. For example, the Rasterize integration uses the `demisto/python.3.3.11.9.1079` Docker image.

You may want to select a different Docker image for your integration or script. In Cortex XSOAR, you can select a different Docker image from a dropdown that is pulled from the Demisto Docker image registry. In Github, the `dockerfiles-info` branch contains information about each image to help you find one that is relevant.

#### **NOTE:**

You can access publicly available Docker hub images from the Cortex XSOAR tenant even if there is no external connection to the Demisto registry hub, for example, if due to firewall constraints your engine cannot access the Demisto registry.

Alternatively, instead of pulling publicly available images in the Demisto registry, you can pull images from a private authenticated image registry. For more information, see Pull images from a private image registry.

You can pull Docker images either directly or through an engine. If using an engine to pull Docker images from a private authenticated registry, you first need to configure the authentication on the engine machine. For more information, see Connect your engine to an image registry.

### Change the Docker image for a script

1. Edit the script.
2. Under ADVANCED, in the Docker image name field, click X to clear the current selection and then select a Docker image name from the dropdown menu.

For more information about changing the Docker image for a script, see the Advanced tab in Create a script.

3. Save your changes.

### Change the Docker image for an integration

1. Go to Settings & Info → Settings → Integrations → Instances, find your integration, and click the pencil icon to edit the integration's source.

For an out-of-the-box content pack integration, you first need to duplicate the integration to edit it.

2. In the Integration Settings, expand the Script section.
  3. Click X to clear the current selection and then select a Docker image name from the dropdown menu.
- For more information about changing the Docker image, see the Advanced tab in Create a script.
4. Save your changes.

### 8.3.1 | Connect your engine to an image registry

#### Abstract

Connect via an engine to your own authenticated Docker image registry.

Using an engine to communicate with an image registry streamlines deployment by managing dependencies, ensuring version control, and facilitating scalability, load balancing, and secure access to private images.

To use an engine, you need to connect the engine to an authenticated Docker image registry and then set it up in the tenant.

#### NOTE:

This procedure uses the `--username` and `--password` command line options to pass the username and password directly. For environments where command history or logs are visible to others, consider more secure methods like Docker configuration files for handling authentication in production or CI/CD environments. For more details, see [docker login](#) or [podman-login](#).

1. Open a terminal on the machine where your engine is running.

2. Run `docker login` with username and password.

```
docker login --username=<your-username> --password=<your-password> <registry-url>
```

Replace `<your-username>`, `<your-password>`, and `<registry-url>` with your Docker registry credentials and the URL of your Docker image registry.

3. (Optional) Search for or pull a Docker image.

After logging in successfully, you can optionally validate access to images by searching for an image or pulling an image from the registry to your local machine using the `docker search` or `docker pull` command.

```
docker search <registry-url>/<image-name>:<tag>
docker pull <registry-url>/<image-name>:<tag>
```

Replace `<registry-url>`, `<image-name>`, and `<tag>` with your registry URL, the name of the Docker image, and the image tag, respectively.

4. In the tenant, set up the engine to pull images from a private image registry.

### 8.3.2 | Pull images from a private image registry

#### Abstract

Create your own authenticated Docker image repository for Cortex XSOAR. View all available images.

Pulling images from a private image registry enables securely accessing and deploying Cortex XSOAR content, for example, custom integrations containing scripts and code packaged into Docker images. You can then run the integrations and scripts in Cortex XSOAR.

#### Before you begin

- Before pulling a custom image ensure the image does not infringe any licenses.
- If using an engine, connect the engine to the private image registry using Docker or Podman. See [Connect your engine to an image registry](#).

#### NOTE:

(Multi-tenant) This feature is not supported on the Main Account.

1. Go to **Settings & Info** → **Settings** → **Advanced** → **Image Registry**.
2. Set **Use Additional Private Image Registry** to **On**.
3. Configure access to the private image registry and the images to pull.

- Select the Connection to either Direct or Using Engine.

For Using Engine:

- Select the Engine to use. Authentication is set on the engine machine itself, not in the Cortex XSOAR tenant. For an example, see Connect your engine to an image registry.

For Direct:

- Set the Username and Password/Access Token.

- Define the Registry URL, for example registry.organization.com

- Click Test the connection to make sure the connection to the registry works.

- Define the Import images in name:tag format, for example myorg/python/new:2.7.18.24398 or myorg/python:latest

You can add, edit, or remove images. If you don't specify a tag, the default tag latest will be added automatically, specifying the latest version of the image.

**NOTE:**

The demisto/ prefix cannot be used for custom registry images.

4. Click Save to persist the configuration and initiate synchronization.

#### Image synchronization

When you click Save or Update Docker Images, Cortex XSOAR performs synchronization, which involves:

- Pulling the images from the external registry.
- Copying and storing the images on the platform.
- Updating the engines with the new images.

**NOTE:**

The synchronization process make take time. The Image Registry page displays synchronization status (for example in progress, complete, failure).

If the engine fails to synchronize, it may be offline. When it goes back online, it will pull any new images when running scripts or integrations that use them.

## 8.4 | Manage credentials

Credentials simplify and compartmentalize administrative tasks, and enable you to save login information without exposing usernames, passwords, certificates, and SSH keys. You can reuse credentials across multiple systems, for example, when using the same administrator password across multiple endpoints.

After you set up a credential, you can configure integration instances to use it instead of entering the name and password manually.

How to add credentials to an integration instance

1. Create the credential.

- a. Select Settings & Info → Settings → Integrations → Credentials → New Credential.

- b. Add the following parameters:

Parameters	Description
Credential Name	The name of the credential. You select this name when adding the credential to the integration instance. For example, Cortex XDR API Key.
Username	The username for the credential.
Workgroup	The workgroup to associate this credential with. Relevant for third-party services, such as Active Directory, CyberArk, and HashiCorps.

Parameters	Description
Password	The password for the credential. For example, add the API Key when defining the API credential.
Certificate	Certificate or SSH to use for the credential.

c. Save the credential.

2. Add the credential to the integration instance.

a. Go to Integrations → Instance and select the integration instance you want to add the credential.

b. Click Add instance.

c. Locate the relevant section and click Switch to credentials.

If there is more than one credential, select the relevant credential.

d. Test and click Save & Exit the integration.

#### Configure an external credentials vault

Cortex XSOAR integrates with external credential vaults, which enables you to use them without hard coding or exposing the credentials. The credentials are not stored in Cortex XSOAR, but the integration fetches the credentials from the external vault when called. The credentials are passed to the relevant executed integrations as part of the integration parameters.

Sample credentials provider integrations:

- CyberArk AIM v2
- HashiCorp Vault

After the integration is configured to fetch credentials, you can also use them in scripts and playbooks. To use these credentials in an integration, click Switch to credentials in an integration instance, and select the necessary credential from the drop-down menu.

## 8.5 | Add an integration instance

### Abstract

Set up an integration instance and start ingesting incidents/indicators.

Configure an integration instance to connect and communicate with other products.

When you define an integration instance for your third-party security and incident management vendors events triggered by this integration instance can become incidents in Cortex XSOAR. When incidents are created, you can run playbooks on these incidents to enrich them with information from other products in your system. For indicators, you can run enrich those indicators depending on the integration instance and add to an incident if required.

Although you can view the integration documents when adding an instance, the Developer Hub has more detailed information about the integrations including commands, outputs, and recommended permissions. You can also see more information about content packs, playbooks, scripts, and Marketplace documentation.

### Before you begin

- From Marketplace, download and install the relevant content pack, which includes your integration.
- Consider whether you want to add credentials, which enable you to save login information without exposing usernames, passwords, certificates, and SSH keys. For more information, see Manage credentials.

1. Go to Settings & Info → Settings → Integrations → instances and search for the integration.

2. In the integration you want to add, click Add instance.

3. Add the parameters, as required.

4. If you want to fetch incidents, select the Fetches incidents.

For more information, see Fetch incidents from an integration instance.

5. (Optional) To check that the integration instance is working correctly, click Test.

## 6. Save & Exit.

Expand the integration to see more details such as the number of pulled incidents/indicators or error messages.

You can also enable/disable the integration instance, copy the instance, and view the integration fetch history.

If you encounter an error, see Troubleshoot integrations.

## 7. (Optional) To manage access to specific commands, see Configure integration permissions.

## 8. (Optional) If you want to set up notifications on an incident fetch error, see Receive notifications on an incident fetch error.

After initially ingesting incidents/indicators, you may need to customize incident/indicator types, fields, and layouts. If relevant to your integration, review and customize classifiers and mappers. Classification determines the type of incident/indicator ingested into Cortex XSOAR from a specific integration. You create a classifier and define that classifier in an integration, if applicable, mapping enables you to map the fields from your third-party integration to the fields in your layouts. For more information, see Classification and mapping.

Example 1.

### How to configure the Cortex XDR - Investigation and Response instance integration

In this example, set up the Palo Alto Networks Cortex XDR - Investigation and Response integration. If you have not done so, download the Cortex XDR content pack from Marketplace. Most integrations follow a similar configuration.

1. Go to Settings & Info → Settings → Integrations → instances and search for Palo Alto Networks Cortex XDR - Investigation and Response.

2. Click Add Instance.

You can see the mandatory fields (with an asterisk) and on the right side, the documentation that contains a link to the full documentation including available commands. See Palo Alto Networks Cortex XDR - Investigation and Response.

3. In the Incident Mirroring field, specify the incident direction.

- Incoming: Changes made to an incident in Cortex XDR are reflected in the fetched event in Cortex XSOAR.
- Outgoing: Changes made in Cortex XSOAR for XDR incidents are reflected in the Cortex XDR tenant.
- Both: Changes made in either platform are to be reflected in either Cortex XDR/XSOAR.

4. Add the Server URL, API Key ID, and the API key that you obtained from Cortex XDR.

5. Add the maximum number of incidents to fetch. By default, there is a maximum number of 10 incidents per minute.

6. Select whether you want only starred incidents from Cortex XDR and the number of days to fetch. By default, fetching is 3 days ago.

7. In the First fetch timestamp field, specify when the first fetch occurs. By default, fetching is 3 days ago.

8. Select the following:

- Sync incident owners between Cortex XDR and Cortex XSOAR.
- Whether to trust certificates not signed by a trusted security authority, such as self-signed certificates.
- Whether to use the system proxy settings.
- Whether to run on Prevent Only Mode to match the Cortex XDR tenant.
- The incident status to fetch.
- Incidents fetch interval. By default, the incidents are fetched every one minute.
- The engine to run on.
- When troubleshooting the instances troubleshooting adjust the default setting from off to a higher debugging level.

9. Specify how Cortex XSOAR collects, classifies, and maps data fetched by this instance. In the Collect Settings you can define the following:

Field	Description
Fetches incidents	Fetches incidents from Cortex XDR. We recommend only fetching incidents when everything is set up. When enabled, Cortex XSOAR searches for events that occurred within the time frame set for the integration, which is based on the specific integration. The default is 10 incidents per minute.
Classifier	Determines which type of incident type is created. For more information about classifiers, see Classification and mapping.
Incident type	If a classifier does not exist, specify an incident type. If a classifier is specified it takes precedence when assigning an incident type to the fetched incident. Incident types determine what playbooks are running on the fetched incident.
Mapper (Incoming)	Determines how incoming data is mapped to the Cortex XSOAR incident fields. In this integration, we are given a default incoming and outgoing mapper. For more information about mappers, see Classification and mapping.
Mapper (Outgoing)	Specifies how Cortex XSOAR incident data should be mapped to external integrations (Cortex XDR). This is important when using incident mirroring.

10. Click Test and Save & Exit.

### 8.5.1 | Fetch incidents from an integration instance

#### Abstract

Configure a third-party integration instance to fetch incidents into Cortex XSOAR incidents for investigation.

You can poll third-party integration instances for events and turn them into Cortex XSOAR incidents (fetching). Many integrations support fetching, but not all support this feature. You can view each integration in the Developer Hub.

When setting up an instance, you can configure the integration instance to fetch events. You can also set the interval for which to fetch new incidents, by configuring the Incidents Fetch Interval field. The fetch interval default is 1 minute. This enables you to control the interval in which an integration instance reaches out to third-party platforms to fetch incidents into Cortex XSOAR.

#### NOTE:

- In some integrations, the Incidents Fetch interval is called Feed Fetch Interval.
- If the integration instance does not have the Incidents Fetch Interval field, you need to add this field by editing the integration settings. If the integration is from a content pack, you need to create a copy of the integration. Any future updates to this integration will not be applied to the copy integration.
- If you turn off fetching for a while and then turn it on or disable the instance and enable it, the instance remembers the last run and pulls all events that occurred while it was off. If you don't want this to happen, verify that the instance is enabled and click Reset the "last run" timestamp when editing the instance. Also, note that "last run" is retained when an instance is renamed.

#### How to fetch incidents from an integration

1. Select the integration instance you want to fetch incidents by going to Settings & Info → Settings → Integrations → Instances finding the integration and clicking + Add instance.

2. Select Fetches incidents.

When enabled, Cortex XSOAR searches for events that occurred within the time frame set for the integration, which is based on the specific integration. The default is 10 minutes prior but can be changed in the integration script.

3. (Optional) In the Incidents Fetch Interval field, set the interval of hours and minutes to fetch incidents (default 1 minute).

4. (Optional) If the Incidents Fetch Interval field does not appear, add it to the integration.

Relevant for any incident fetching integration.

a. For integrations installed from a content pack, select the duplicate integration button.

If you already duplicated the integration, click the Edit integration's source button.

b. In the Basic section, select the Fetches incidents checkbox.

In the Parameters section, you can see that the `IncidentFetchInterval` parameter is added. Change the default value if necessary.

- c. Save the changes.

### 8.5.2 | Receive notifications on an incident fetch error

#### Abstract

Add a server configuration to receive notifications if an integration experiences an incident fetch error.

The administrator and Cortex XSOAR users on the recipient's list receive a notification when an integration experiences an incident fetch error. Administrators with multiple instances of mail senders can choose to receive one email notification instead of multiple email notifications. Cortex XSOAR users can select their notification method, such as email, from their user preferences.

#### **NOTE:**

The connectivity behavior that exists between third-party applications may trigger a fetch failure, which will send a notification to an administrator and users. The notification may no longer be relevant because the fetch might operate correctly just after the notification was sent.

1. In the integration instance, select the Fetch Incidents checkbox.
2. Select Settings & Info → Settings → System → Server Settings → Add Server Configuration.
3. Add the following keys and values:

Key	Value
module.health.notification.users	List of names in CSV format, for example, <code>user1,user2,user3</code> .
message.ignore.failedFetchIncidents	<code>false</code>

4. (Optional) Administrators that have multiple instances of a mail sender configured that want to receive only one email notification should select the Do not use by default option in the integration instances that should not be used to send emails.

### 8.5.3 | Configure integration permissions

#### Abstract

Integration permissions enable you to assign permissions to commands in integrations. Use role-based access control (RBAC) to assign commands.

You can use role-based access control (RBAC) to assign commands at the integration instance level. If you have multiple instances of the same integration, you can assign different roles (permission levels) for the same command in each instance.

Users who do not have permission to run a command, cannot do the following:

- Run the command from the CLI.
- Complete pending tasks in a Work Plan that uses the restricted command.
- Edit arguments for playbook tasks that use the restricted command.
- Select the command when editing a playbook.
- Leverage the restricted command when executing a reputation command, such as IP, Domain, and File.

#### **NOTE:**

To restrict access to integrations (not just commands), see Role-based permissions.

To view or edit integration permissions:

1. Go to Settings & Info → Settings → Integrations → Integration Permissions.

You can see a list of all enabled integrations.

2. Select the integration.

You can see the following:

- INSTANCE: Lists all instances for the integration.
- COMMANDS: Lists all commands for the integration.
- PERMITTED ROLES: Lists the roles you can assign to the command.

3. If you want to limit a command to a role, do the following:

You may want limit potentially harmful commands, such as in Cortex XDR you may want to limit the ability to isolate endpoints.

1. Click Edit.
  2. Go to the relevant command.
  3. In the PERMITTED ROLES, column, select the roles that you want to limit.
4. Save the integration permissions.

#### 8.5.4 | Troubleshoot integrations

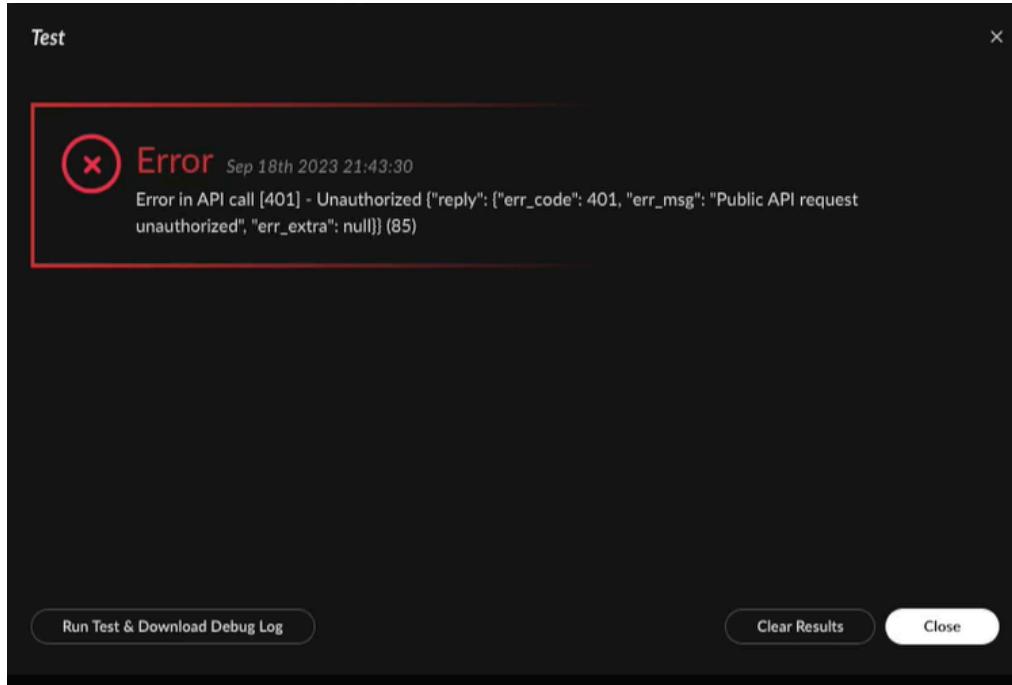
##### Abstract

Learn how to troubleshoot your integration in Cortex XSOAR.

When troubleshooting integrations, do the following:

- Use the Test button in the integration instance.
- Verify the integration settings. Check settings such as usernames, URLs, and passwords.
- Download the debug log file and review its contents.

In the following example, you receive a 401 unauthorized error code after testing the integration.



Click Run Test & Download Debug log, to download the debug file locally. You can verify what server the URL request is being forwarded to and any other reasons as to why you received this error code. The 401 unauthorized error code usually relates to invalid error credentials, expired tokens, or incorrect API settings.

- Enable verbose or debug-level logging on the integration.
- Review the integration logs (Settings & Info → Settings → Integrations → Integration Logs).

You can sort the logs by things such as source instance, command, and log level. You can also export the file

If you are unable to fix the integration, contact Customer Support for further assistance.

## 8.6 | Integration commands in the CLI

### Abstract

Run integration commands in the CLI.

The command line interface (CLI) enables you to run system commands, integration commands, scripts, and more from the CLI. The CLI auto-complete feature allows you to find relevant commands, scripts, and arguments.

Cortex XSOAR uses the following commands:

- System commands: These commands are not specific to an integration. System commands are entered in the CLI using a "/". For example, /clear\_playground.
- External commands: These commands are specific to an integration and perform actions relating to a specific integration, using "!". For example, !xdr-get-alerts.

Go to Settings & Info → Settings+Integrations → Instances, under each integration, you can view a list of commands.

### NOTE:

Integration commands are only available when the integration instance is enabled. Some commands depend on a successful connection between Cortex XSOAR and third-party integrations.

You can run the CLI commands on any page where the CLI appears or in an incident. If run on a page not in an incident, the results are returned to the Playground. The Playground is a non-production environment where you can safely develop and test automation scripts, APIs, commands, and more. It is an investigation area that is not connected to a live (active) investigation.

In the following example, set up the Palo Alto Networks Cortex XDR - Investigation and Response integration instance. To retrieve Cortex XDR incidents, for the last year, sort by time in ascending order and limit to 5 incidents type the following in the CLI:

```
!xdr-get-incidents limit = 5 since_creation_time="1 year" sort_by_creation_time=asc
```

In the Playground, you can see the list of incidents in a markdown table.

To see the incidents in a JSON format, select Side Panels → Context Data. Each incident contains information obtained from the Cortex XDR endpoint that can be used in subsequent commands. You can search for a field such as `incident_id`. To get more information about the `incident_id:1`, copy the data, by clicking the `incident_id` in the context data.

To retrieve additional data from `incident_id`:

```
!xdr-get-incident-extra-data incident_id ${value copied from context data}
```

For example `!xdr-get-incident-extra-data incident_id ${PaloAltoNetworksXDR.Incident.[0].incident.id}`

You can then see additional information.

Clear Context Data | Collapse | Expand

- > Account: [...] 1 item
- ✓ Endpoint:
  - Hostname: cpss-XDR-piyush
  - ID: 1b70072e0f094891a53dfbcfa4ea4b8
- ✓ File:
  - > 0: [...] 2 items
  - > 1: [...] 2 items
  - > 2: [...] 2 items
  - > 3: [...] 2 items
  - > 4: [...] 2 items
  - > 5: [...] 2 items
- > PaloAltoNetworksXDR: [...] 1 item
- ✓ Process:
  - ✓ 0:
    - Hostname: cpss-XDR-piyush
    - Name: System
    - PID: 4
    - Path: System
    - Start Time: 1686130957797
  - > 1: [...] 7 items
  - > 2: [...] 8 items
  - > 3: [...] 8 items
- > incident: [...] 61 items

**TIP:**

If you want to delete context in the Playground, type !DeleteContext all=yes. To clear the playground, at the top of the page, click Clear playground. To erase a playground and create a new one, run the /playground\_create command.

## 8.7 | Forward requests to long-running integrations

### Abstract

Configure and manage long-running integrations to export internal data from Cortex XSOAR.

Some long-running integrations provide internal data via API calls, to your third-party software, such as a firewall. You can set up Cortex XSOAR to allow third-party software to access long-running integrations installed either on the Cortex XSOAR tenant or on an engine.

Rather than adding credentials separately for long-running integration instances, you can set up universal credentials for all long-running integrations.

Long-running integrations provide internal data via API calls such as:

Integration	Description	See More
O365 Teams (Using Graph API)	Get authorized access to a user's Teams app in a personal or organization account.	O365 Teams (Using Graph API)
Generic Webhook	Creates incidents on event triggers. The trigger can be any query posted to the integration.	Generic Webhook
Generic Export Indicators Service	Use the Generic Export Indicators Service integration to provide an endpoint with a list of indicators as a service for the system indicators. You can set up the tenant to export internal data to an endpoint.  <b>NOTE:</b> This integration replaces the External Dynamic list integration, which is deprecated.	Generic Export Indicators
TAXII Server	Provides TAXII Services for system indicators (Outbound feed).	TAXII Server
TAXII2 Server	Provides TAXII2 Services for system indicators (outbound feed). You can choose to use TAXII v2.0 or TAXII v2.1.	TAXII2 Server

Integration	Description	See More
XSOAR-Web-Server	Supports handling configurable user responses (like Yes/No/Maybe) and data collection tasks that can be used to fetch key value pairs.	XSOAR-Web-Server
PingCastle	Listens for PingCastle XML reports.	PingCastle
Publish List	Publishes XSOAR lists for external consumption.	Publish List
Simple API Proxy	Provides a simple API proxy to restrict privileges or minimize the amount of credentials issued at the API.	Simple API Proxy
Syslog v2	Opens incidents automatically from Syslog clients.	Syslog v2
Web File Repository	Makes your environment ready for testing purpose for your playbooks or automations to download files from a web server.	Web File Repository

**NOTE:**

- When running on the tenant, you can only use long-running integrations provided by Cortex XSOAR, you cannot create custom ones. Custom long-running integrations are supported only on engines at this time.
- Configuring custom certificates or private API Keys in the long-running integration instance is supported only on engines, not on the Cortex XSOAR tenant.

**Define universal credentials for long-running integrations**

When defining credentials for long-running integrations, you can do one of the following:

- Set up universal credentials for all long-running integrations

You need the Account Admin or Instance Administrator's permission to define credentials.

**TIP:**

For long-running integrations running on an engine, we strongly recommend defining a username and password, but it is not required.

- Set up credentials for each separate integration

Users with sufficient permissions can set the username and password for specific integration instances, on the Integrations → Instances page.

**IMPORTANT:**

If you define credentials in long-running integrations, but there is a different username and password in an individual integration instance, the credentials for the integration instance override the long-running integration credentials.

- Go to Settings & Info → Settings → Integrations → Long Running Integrations.
- In the Configure Universal Credentials for Long Running Integrations (Optional) section, add a username and password.
- Save the configuration.

When configuring a long-running integration, you don't need to add a username and password.

**Test the long-running integration connection**

- **Integration Instance Running on a Tenant**

You can use CURL commands from any terminal to access and test the long-running integration at the URL:

```
https://ext-<cortex-xsoar-address>/xsoar/instance/execute/<instance-name>
```

For example, curl -v -u user:pass https://ext-mytenant.paloaltonetworks.com/xsoar/instance/execute/edl\_instance\_01\?q\=type:ip

**NOTE:**

The data URL must always be prefixed by ext-.

- **Integration Instance Running on an Engine**

You can use CURL commands from any terminal to access and test the long-running integration at the engine URL:

```
http://<engine-address>:<integration listen port>/
```

For example, curl -v -u user:pass http://<engine\_address>:<listen\_port>/?n=50

#### Curl request parameters

When sending a curl request to the URL, you can use the following parameters.

Argument	Description	Example
n	The maximum number of entries in the output. If no value is provided, will use the value specified in the List Size parameter in the integration instance settings.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?n=50</code>
s	The starting entry index from which to export the indicators.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?s=10&amp;n=50</code>
v	The output format. Supports PAN-OS (text), CSV, JSON, mwg and proxysg (alias: bluecoat).	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=json</code>
q	The query used to retrieve indicators from the system.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?q="type:ip and sourceBrand:my_source"</code>
t	Only with mwg format. The type indicated on the top of the exported list. Supports: string, applcontrol, dimension, category, ip, mediatype, number and regex.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=mwg&amp;t=ip</code>
sp	If set, will strip ports off URLs, otherwise will ignore URLs with ports.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=text&amp;sp</code>
di	Only with PAN-OS (text) format. If set, will ignore URLs which are not compliant with PAN-OS URL format instead of being re-written.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=text&amp;di</code>

Argument	Description	Example
cr	If set, will strip protocols off URLs.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=text&amp;pr</code>
cd	Only with proxysg format. The default category for the exported indicators.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=proxysg&amp;cd=default_category</code>
ca	Only with proxysg format. The categories which will be exported. Indicators not in these categories will be classified as the default category.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=proxysg&amp;ca=category1,category2</code>
tr	Only with PAN-OS (text) format. Whether to collapse IPs. <ul style="list-style-type: none"> <li>• 0 - Do not collapse.</li> <li>• 1 - Collapse to ranges.</li> <li>• 2 - Collapse to CIDRs</li> </ul>	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?q="type:ip and sourceBrand:my_source"&amp;tr=1</code>
tx	Whether to output CSV formats as textual web pages.	<code>https://ext-&lt;cortex-xsoar_instance&gt;/instance/execute/&lt;ExportIndicators_instance_name&gt;?v=csv&amp;tx</code>

#### Define a listening port for long-running integrations

When configuring a long-running integration instance you may need to define a listening port.

- **Integration Instance Running on a Tenant**

If the long-running integration runs on the Cortex XSOAR tenant, you do not need to enter a Listen Port in the instance settings. The system auto-selects an unused port for the long-running integration when the instance is saved.

- **Integration Instance Running on an Engine**

You must set the Listen Port for access when configuring a long-running integration instance on an engine. Use a unique port for each long-running integration instance. Do not use the same port for multiple instances.

## 9 | Incident configuration

### Abstract

Customize how the incident appears, add deduplication rules, and add any other customizations you require for your workflow.

Tailor incidents to match your SOC workflow. Customize incident types, fields, and layouts for optimal clarity. Automate tasks with pre- and post-processing rules for deduplication and streamline access control with RBAC. Easily classify and map incidents, and personalize closed reasons for improved efficiency.

### 9.1 | Incident lifecycle

#### Abstract

An incident goes through various processes in Cortex XSOAR including defining an incident, classification and mapping, pre and post-processing, and running a playbook.

Incidents are potential security data threats that SOC analysts identify and remediate. There are several incident triggers, including:

- SIEM alerts
- Mail alerts
- Security alerts

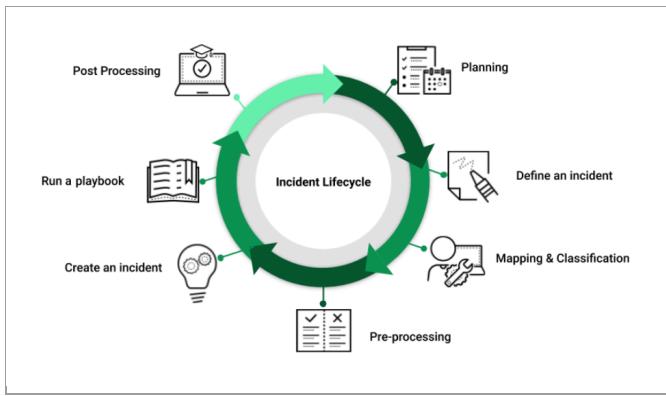
These alerts are generated from third-party services, such as SIEMs, mailboxes, and data.

Cortex XSOAR includes several out-of-the-box incident types, fields, and layouts, which can be customized to suit your use case. You can also create incident types, custom fields, and layouts as necessary. Incidents can be created manually, from a JSON file, the Cortex XSOAR Restful API, or an integration feed.

You can define integrations with your third-party security and incident management vendors. You can trigger events from these integrations that become incidents in Cortex XSOAR. You can run playbooks on these incidents to enrich them with information from other products in your system, which helps you complete the picture.

In most cases, use rules and scripts to determine if an incident requires further investigation or can be closed based on the findings. You can filter the incidents that are ingested into Cortex XSOAR by manually de-duplicating incidents, setting up pre-process rules to perform certain actions, or automatically de-duplicate incidents. This enables your analysts to focus on the minority of incidents that require further investigation. After you close an incident you may want to automate an additional action such as closing a remedy ticket. For more information, see [Use post-processing scripts in an incident](#).

The following diagram explains the incident lifecycle in Cortex XSOAR.



## Planning

Before you begin configuring integrations and ingesting information from third parties, consider the following:

Phase	Description
Incident types	Incident types classify the events that are ingested into Cortex XSOAR. Use out-of-the-box types, or create incident types to classify the different types of attacks with which your organization deals. For more information, see <a href="#">Create an incident type</a> .
Incident fields	Displays information from third-party integrations and playbook tasks when an incident is created or processed. Use out-of-the-box fields, or create fields for your use case. For more information, see <a href="#">Create an incident field</a> .
Incident layouts	Customize your layouts by adding custom or system fields for each incident type, so that the most relevant information is shown for each type. For more information, see <a href="#">Incident layout customization</a> .

This is an iterative process. After you've configured incident types, fields, and layouts, and you've classified and mapped your incident type and fields, start ingesting information, which enables you to assess how you've mapped out your information. As you see the data coming in, you can make adjustments to improve your mapping and gain a deeper understanding of the information you're collecting. Although unmapped information is available in labels, it's significantly easier to work with, when assigned to a specific field and displayed in the appropriate layouts.

## Configure integrations

Configure integrations with third-party products to start fetching events, such as potential phishing emails, authentication attempts, and SIEM events. For more information, see [Configure integrations](#).

## Classification and mapping

Once you configure integrations, you should determine how the events ingested from those integrations will be classified as incidents. For example, you can classify items based on the subject field for email integrations, but for SIEM events, you should classify them by event type. During the planning stage, it's important to define how the information ingested from your integrations will be mapped to the fields you're creating. For more information, see Classification and mapping.

### Pre-Processing

Pre-processing rules enable you to perform certain actions on incidents as they are ingested into Cortex XSOAR. Using rules, you can select incoming events on which to perform actions, for example, link the incoming event to an existing incident, or based on configured conditions, drop the incoming incident altogether. For more information, see Pre-process rules.

### Create an incident

Based on the definitions provided in the Classification and Mapping stage, and the rules you created for pre-processing events, incidents of various types are created. The incidents all appear on the Incidents page, where you can start investigating incidents.

### Run a playbook

Playbooks are triggered when an incident is created or run them manually as part of an investigation. When triggered as part of a created incident, the playbooks for the type of incident that was classified run on the incident. Alternatively, if you are manually running a playbook, select whichever playbook is relevant for the investigation. For example, playbooks can take IP address information from one integration and enrich that IP address with information from additional integrations or sources.

### Post Processing

Once the incident is complete and you are ready to close it, you can run various actions on the incident using a post-processing script. For example, an email is sent to the person who opened the incident informing them that their incident has been resolved or close an incident in a ticketing system. For more information, see Use post-processing scripts in an incident.

## 9.2 | Incident Customization

### Abstract

Create and edit incident types, fields, and layouts in Cortex XSOAR.

Several content packs, such as Cortex XDR by Palo Alto Networks, include out-of-the-box integrations, incident types, fields, and layouts. You may need to customize incident types, fields, and layouts to suit your needs or create new ones to investigate and respond to potential security threats specific to your organization.

You can customize the following:

Option	Description
Incident types	You can create a new incident type or customize the incident type, such as setting the default playbook, adding the layout, and any post-process and indicator extraction rules. You can create, duplicate, import, export, and customize incident types. For more information about creating an incident type, see Create an incident type.
Incident fields	Custom incident fields add specific details or attributes to incidents, helping analysts to investigate and understand potential security threats. You can edit or create an incident field. For more information, see Create an incident field.  After creating an incident indicator field, map the field to the relevant context data. You can add the field to an incident type and view it in an incident layout.
Incident layouts	Custom incident layouts enable you to organize and display specific details about potential threats in a way that makes sense for your organization, making it easier to quickly understand and respond to security issues. You can view, customize, import, and export indicator layouts and add a custom layout to an incident type. For more information, see Incident layout customization.

This is an iterative process. After you initially create your types, fields, and layouts, you can start the process of ingesting information by installing and configuring an integration to fetch incidents.

When you configure an integration instance, you can define a classifier and a mapper for the integration. When an incident is ingested into Cortex XSOAR, the integration assigns the incident type when classified and maps the event data into incident fields. For example, when defining the EWS O365 instance integration, setting the classifier to EWS - Classifier, classifies all incoming incident types as Phishing from the O365 integration.

Consider the following:

- When an incident is ingested, one of the first entries in the War Room is the fields and values returned. You may want some of this information to appear on the Incident Info/Summary page when an analyst starts investigating.
- Review the context data (from Side panels). Context data is a map (dictionary) that stores structured results from data, such as commands, playbooks, and scripts. If there is information in the context data you don't see in the incident, map it into incident fields and display it in the layout. For more information, see Use incident context data.

See this video about creating incident types and fields.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

#### 9.2.1 | Use incident context data

##### Abstract

Use context data to customize your incident layout and to populate your incidents in Cortex XSOAR.

Context data is a map (dictionary) that stores results from data, such as commands, playbooks, and scripts in a structured format. Context data includes keys (strings) and values (strings, numbers, maps, and arrays). Context data at its core is a large JSON structure, which represents all the data that is part of an incident. All incidents have context data.

You can use context data to pass data between playbook tasks, capture important structured data, and display it in the incident layout. Context data acts as an incident data dump from which you can map data into incident fields. When an incident is generated in Cortex XSOAR and a playbook or analyst begins investigating it, context data will be written to the incident to assist with the investigation and remediation process.

When an incident is created, the incident data is stored in the context data, under the `incident` key. When an investigation is opened and integration commands are run, data returned from those commands is also stored outside of the main `incident` key. In the following example, you can see the original incident data stored under the `incident` key and the data from the integrations, such as Wildfire, stored separately within the context data under their keys.

The screenshot shows the 'Context Data' panel in Cortex XSOAR. At the top, there's a search bar labeled 'Search in JSON context data...'. Below the search bar is a tree view of context data items. The root node is 'root' (13 items), which contains the following sub-nodes: WildFire (1 item), HelloWorld (1 item), Alexa (1 item), Domain (9 items), IP (5 items), VirusTotal (1 item), AutoFocus (4 items), URL (1 item), DBotScore (16 items), IPinfo (1 item), XFE (2 items), File (3 items), and incident (95 items).

```

root: [] 13 items
  ▶ WildFire: [] 1 item
  ▶ HelloWorld: [] 1 item
  ▶ Alexa: [] 1 item
  ▶ Domain: [] 9 items
  ▶ IP: [] 5 items
  ▶ VirusTotal: [] 1 item
  ▶ AutoFocus: [] 4 items
  ▶ URL: [] 1 item
  ▶ DBotScore: [] 16 items
  ▶ IPinfo: [] 1 item
  ▶ XFE: [] 2 items
  ▶ File: [] 3 items
  ▶ incident: [] 95 items
  
```

Consider the following when working with context data:

- Add keys and values to the context data, such as the incident status, actions, and ID. This is useful when developing playbooks, and other scripts.
- Add context data to incident fields in a layout to capture important and relevant information to assist with investigation and remediation.

#### Search context data

To view context data from within an incident, click on the Side panels menu and select Context Data from the dropdown. In the Context Data pane, you can use Query to search within the JSON for specific items and expand nested keys.

#### Example 2.

- `${c}`  finds the value of the object `c`.
- `${HelloWorld.Domain(val.domain == 'example.com')}`  shows the full object for the example.com domain, as stored in the context data by the domain command that is part of the HelloWorld integration.
- `${HelloWorld.Domain(val.domain == 'example.com').registrar}`  shows the registrar for the example.com domain, as stored in the context data by the domain command that is part of the HelloWorld integration.
- `${HelloWorld.Alert(val.alert_status === "ACTIVE").alert_id}`  fetches the HelloWorld.Alert.alert\_id of all ACTIVE alerts.

You can also write jQuery scripts using complex logic to access, aggregate, and change context data. For more information, see Cortex XSOAR Transform Language (commonly referred to as DT).

#### Customize incident fields and layouts using context data

When fetching incidents from an integration, some important data may not have been picked up in the incident layout. For example, the context data may return the source user, event type, URL category, and suspicious URL but these fields may not appear as fields or in the layout. For more information about customization, see Incident Customization.

#### Use context data in a playbook

The main use of context data is to pass data between playbook tasks, one task stores its output in the context and the other reads that output from the context and uses it. For more information about how to use context data, including examples and use cases, see Context and Outputs.

In a playbook, you can use context data in the following situations:

- Inputs and outputs in playbook tasks

You can use the information stored in the incident context and apply filters and transformers to context data before using the data in playbook tasks.

- Write playbook data to the incident context

Add a task to use context data to run additional playbooks as required.

- Test playbooks by using the playbook debugger

While running a playbook using the playbook debugger. As context data may be updated during a playbook run, set a breakpoint to view the context data after a specific task, which can be useful for designing and troubleshooting playbooks.

By default, context data for sub-playbooks is stored in a separate context key. When a task in a sub-playbook accesses context data, it does not have direct access to the main playbook data. If, however, the sub-playbook has been configured to share globally, the sub-playbook context data is available to the main playbook and vice versa.

#### NOTE:

Generic polling does not work if a playbook's context data is shared globally. For more information, see Playbook polling.

#### Use context data in a script

In any script that runs in an incident, the data is written to the context. For example, `demisto.executeCommand("set", {"key":<key>", "value":<value>"})`. For more information, see Set Command.

#### Add/delete context data using the CLI

To add context data to an incident, run the Set command in the CLI. The Set command enables you to set a value under a specific key. For more information about the Set command, see Set Command.

In the incident that you are investigating run the !Set command. For example, to add the key and value `hello:world` to the context data, run the following command:

```
!Set key="hello" value="world"
```

#### NOTE:

All incident data stored in incident fields are also stored in the context data. In most cases, however, not all context data is stored in incident fields. Incident fields represent a subset of the total incident data.

In the incident context data you want to delete, run the `DeleteContext` command in the CLI. For example, to delete the key and value `hello:world` from the context data, run the following command:

```
!deleteContext="hello"
```

### 9.2.2 | Create an incident type

#### Abstract

Create and edit incident types in Cortex XSOAR.

You can create an incident type if the incident type does not exist and then classify the incident according to this incident type. Each incident type has a unique set of data relevant to that specific incident type. When you duplicate an incident type, the duplicate is associated with the same set of incident fields that belonged to the original incident type.

By default, when installing incident types from a content pack, incident types are attached, which means they are not editable. If you want to edit the incident type, such as changing the layout or the default playbook, you have the following options:

- Duplicate the incident type

The duplicate type is editable and the original incident type continues to receive content pack updates, but the duplicate does not.

- Detach the incident type

While an incident type is detached, it does not receive content pack updates. If you detach an incident type and make changes, any changes made while it was detached are overwritten by the default values from the content pack. If you want to keep the changes and protect your changes from content pack upgrades, duplicate the incident type before reattaching the original.

- Select **Settings & Info** → **Settings** → **Object Setup** → **Incidents** → **Types** → **New Incident Type**.

- In the **Settings** tab, add the following parameters, as required:

Field	Description
Name	Enter a descriptive name for the type. Try to make the name informative, so users know what the type does before viewing the type details.
Default playbook	Select the default playbook that is associated with the incident type.
Run playbook automatically	Determines if the playbook runs automatically when the incident is ingested.
Layout	Select the incident layout for the incident type.
Post Process using	After incidents have been investigated, select the post-process script to run on these incident types. For more information, see <a href="#">Use post-processing scripts in an incident</a> .
SLA	Determines the SLA for this incident type in any combination of Weeks, Days, and Hours. For more information, see <a href="#">Configure an SLA in an incident type</a> .
Set Reminder at	Optionally configure a reminder for the SLA in any combination of Weeks, Days, and Hours.

- In the **Indicators Extraction Rules** tab, add the required rules.

Indicator extraction rules extract indicators from incident fields and enrich them using commands and scripts. You can view and create indicator extraction rules according to incident fields. For more information, see [Create indicator extraction rules for an incident type](#).

- Save the indicator type.

### 9.2.3 | Create an incident field

#### Abstract

Create custom incident fields in Cortex XSOAR.

Incident fields are used to accept or populate incident data coming from incidents. These fields are added to incident layouts and are mapped using classification and mapping.

Creating incident fields is an ongoing process. You can create fields from information ingested from third-party integrations. As you learn more about your needs and the capabilities of your third-party integrations, you can continually add new fields to capture the most relevant information.

When investigating an incident, an analyst can easily add relevant information to the fields in the layout. Incident fields can be populated by incident team members during an investigation at the beginning of the investigation, or before closing the investigation.

#### NOTE:

In the CLI, you can set and update all system incident fields using the `setIncident` command, of which each field is a command argument.

#### Field types

You can create the following field types:

Field Type	Description
Attachments	Enables the user to add an attachment, such as .doc, malicious files, reports, and incident images.
Boolean	Checkbox
Date picker	Adds the date to the field.
Grid (table)	<p>Include an interactive, editable grid as a field type for selected incident types or all incident types. To see how to create a grid field and to use a script, see <a href="#">Use scripts with a grid field</a>.</p> <p>When you select Grid (table) you can format the table and determine if the user can add rows.</p>
HTML	Create and view HTML content, which can be used in any incident type.
Long text	<ul style="list-style-type: none"><li>Long text is analyzed and tokenized, and entries are indexed as individual words, enabling you to perform advanced searches and use wildcards.</li><li>Long text fields can't be sorted and used in graphical dashboard widgets.</li><li>While editing a long text field, pressing enter will create a new line (case is insensitive).</li></ul> <p>Add a placeholder, if required.</p>
Markdown	Add markdown formatted text as a Template which will be displayed to users in the field after the indicator has been created. Markdown lets you add basic formatting to text to provide a better end-user experience.
Multi select / Array	Select the following options: <ul style="list-style-type: none"><li>Multi-select from a (static) pre-filled list.</li><li>An empty array field for the user to add one or more values as a comma-separated list.</li></ul> <p>Add a placeholder, if required.</p>
Number	Can contain any number. Default is 0.

Field Type	Description
Role	Role assigned to the incident. Determines which users (by role) can view the incident.
Short Text	<ul style="list-style-type: none"> <li>Short text is treated as a single unit of text and is not indexed by word. Advanced search, including wildcards, is not supported.</li> <li>Short text fields are case-sensitive by default but can be changed to case-insensitive when creating the field.</li> <li>While editing a short text field, pressing enter will save and close.</li> <li>Maximum length 60,000 characters.</li> <li>Recommended use is one-word entries. Examples: username, email address, etc.</li> </ul>
Single select	Select a value from a list of options. Add comma-separated values.
Tags	<p>Accepts a single tag or a comma-separated list, not case-sensitive.</p> <p>Add a placeholder, if required.</p>
Timer/SLA	<p>View how much time is left before an SLA becomes past due, as well as configure actions to take if the SLA does pass.</p> <p><b>NOTE:</b> Incidents sorted using an SLA/Timer field are sorted by the due date of the SLA field.</p>
URL	Add a URL when completing the field.
User	A user in Cortex XSOAR.

#### How to create a field

1. Select Settings & Info → Settings → Object Setup → Incidents → Incident Fields → New Field.

To edit an existing incident field, right-click the field name and select Edit.

2. Select the relevant field type.

3. Add the following information:

Parameter	Description
Mandatory	If selected, this field is mandatory when used in a form.
Field Name	<p>A meaningful display name for the field. After you type a name, you will see below the field that the Machine name is automatically populated. The field's machine name is applicable for searching and the CLI.</p> <p><b>NOTE:</b> If you try to create a new incident field with a name that already exists in the system such as <b>Account</b>, you may receive a message like this:  <code>[Could not create incidentfield with ID '' and name 'Account'.Field already exists as a builtin field (100709)].</code>          If so, select a different name as the incident field is already reserved for system use.          You should not create a custom field named <b>reason</b> as it is a saved keyword in the tenant.</p>

Parameter	Description
Tooltip	An optional tooltip for the field.

4. In the Basic Settings tab, define the values according to the selected field type.

Parameter	Description
Placeholder	Optional text to display in the field when it is empty. This text will appear in the layout, but not in the created incident. Available for Short text, Long text, Multi select / Array, and Tags.
Values	A comma-separated list of values that are valid values for the field.

5. If selecting a Timer/SLA field, define the following:

Parameter	Description
SLA	Determine the amount of time in which this item needs to be resolved. If no value is entered, the field serves as a counter.
Risk Threshold	Determine the point in time at which an item is considered at risk of not meeting the SLA. By default, the threshold is 3 days, which is defined in the global system parameter.
Run on SLA Breach	In the Run on SLA Breach field, select the script to run when the SLA time has passed. For example, email the supervisor or change the assignee.  <b>NOTE:</b> Only scripts to which you have added the SLA tag appear in list of scripts that you can select.

6. If you are creating a Grid (table) field, in the Grid tab, define the following values.

- To enable users to add/remove rows in the grid, select the User can add rows field. If selected, the user can add rows but not columns.
- Manage rows and columns. You can move the columns and add/delete rows and columns (using the + and - signs). How you design the grid determines how it appears to users.
- Configure each column by clicking the settings button in each column. Add the column name, select whether the column is mandatory, and the field type. If you select Lock, the value for that field is static (not editable). If you do not select the Lock checkbox (default), users can perform inline editing.

7. In the Attributes tab, define the following:

Field	Description
Script to run when field value changes	The script dynamically changes the field value when script conditions are met. For a script to be available, it must have the <b>field-change-triggered-indicator</b> tag when defining the script.  For more information, see .

Field	Description
Run the field triggered script after the new field value is saved	<p>Leave unchecked for the script to execute before the incident is stored in the database, so the script can modify the incident field value. Useful in most cases including performing validations and starting and stopping Timer/SLA fields.</p> <p>When checked, the script executes after the incident is stored in the database, so that the script cannot modify the incident unless through CLI or API calls.</p> <p>For example, add the <code>emailFieldTriggered</code> script, which runs after the Incident Updates tag is stored in the database (unchecked).</p>
Field display script	<p>Determines which fields display in forms, as well as the values that are available for single-select and multi-select fields. For more information, see Create Dynamic Fields in Incident Forms.</p>
Add to all incident types	<p>Determines for which incident types this field is available. By default, fields are available to all incident types. To change this, clear the Add to all Incident types checkbox and select the specific incident types to which the field is available. For example, you may want to limit the field to Access, Malware and Network incident types.</p>
Default display on	<p>Determines at which point the field is available. For more information, see Incident Field Examples.</p>
Edit Permissions	<p>Determines whether only the owner of the incident can edit this field.</p>
Indexing Make data available for search	<p>Determines if the values in these fields are available when searching.</p> <p><b>NOTE:</b></p> <p>In most cases, Cortex XSOAR recommends that you select this checkbox so values in the field are available for indexing and querying. However, in some cases, to avoid adverse effects on performance, you should clear this checkbox. For example, if you are ingesting an email to an email body field, we recommend that you not index the field.</p>

#### 8. Save the field.

If you subsequently edit the field, you can select Don't show in the incidents layout. If selected, the incident field does not appear in the layout, but the data is displayed in the context data.

#### 9. Add the field to an incident layout.

#### 10. (Optional) In the incident type, map the incident field, so the incident field is automatically updated, without the analyst having to change it.

#### Incident field examples

The following section shows several examples of common fields used in real-life incidents.

#### False positive

Below is an example of a mandatory False Positive field, which will be completed when the incident is closed. The Field can have a value Yes or No. The Administrator can query or run a report based on this field. After this field is added, all incidents need to complete this field, before an incident can be marked closed.

## New Incident Field

Field Type

Single select

Mandatory

\* Field Name  
**False Positive**

Machine name: falsepositive (use in search and command line)

Tooltip  
*Was the incident a false positive?*

**Basic Settings** **Attributes**

\* Values (comma separated)  
No, Yes

"Select" will be displayed as default option, if first is not default

Use first as default

Cancel Save

## New Incident Field

Field Type  
Single select  Mandatory

\* Field Name  
**False Positive**  
Machine name: *falsepositive* (use in search and command line)

Tooltip  
*Was the incident a false positive?*

**Basic Settings** **Attributes**

Script to run when field value changes [Choose script](#) [?](#)

Run the field triggered script after the new field value is saved [?](#)

Field display script [Choose script](#) [?](#)

Add to all Incident types

Default display on:  
 New / Edit  Close  Both

Edit Permissions  
 Only owner can edit

[Cancel](#) [Save](#)

### SLA fields

The following SLA field can be used to trigger a notification when the status affecting the SLA of an incident changes. In this example, if the SLA is breached an email is sent to the owner's supervisor.

**New Incident Field**

**Field Type**  
Timer/SLA

**\* Field Name**  
**Notify On Change**  
Machine name: notifyonchange (use in search and command line)

**Tooltip**  
Notifies you of changes to the status of the incident

**Basic Settings**    **Attributes**

**SLA**  
00 Hours 00 Minutes

**Risk Threshold** ?  
3 Days 00 Hours

**Run on SLA Breach** SendEmailOnSLABreach × ?

Cancel    Save

#### 9.2.3.1 | Incident field trigger scripts

##### Abstract

Associate Cortex XSOAR incident fields with scripts that are triggered when the field changes.

Incident fields can be associated with trigger scripts that check for field change conditions and take actions based on the change. These scripts can perform any action, such as dynamically changing the field value, notifying the responder when an incident severity has been changed, or when the conditions are met. For example, the **ChangeRemediationSLAOnSevChange** script changes the Remediation SLA of an incident, if the severity of the incident changes for any reason.

Scripts can be created in Python, PowerShell, or JavaScript on the Scripts page. To use a field trigger script, you need to add the field-change-triggered tag when creating the script. You can then add the script in the Attributes tab, when you edit or create an incident field. If you did not add the tag when creating the script, it cannot be selected, until you add the tag.

Cortex XSOAR comes out-of-the-box with field change scripts in the Scripts page, such as:

- **ChangeRemediationSLAOnSevChange**: Changes the remediation SLA once a change in incident severity occurs.
- **emailFieldTriggered**: Sends an email to the incident owner when the selected field is triggered.
- **StopTimeToAssignOnOwnerChange**: Stops the Time to Assignment SLA field, as soon as an owner was assigned to an incident.

A common use case is to create a script that only allows automated changes by a playbook not manual changes by a user.

```
args = demisto.args()
user = args["user"]
```

```
if user:
    demisto.executeCommand("setIncident", {"args": {"cliName": args["old"]}})
```

The script checks who made the change using the user field. The cliName argument returns the field name, so that it can be attached to multiple incident fields, and block changes to them, without the need to have a different script for each field.

See the following video about how to create and add scripts to an incident layout:

Terjadi error.

[Cobalah menonton video ini di www.youtube.com](#), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

#### Incident field trigger script arguments

Incident field trigger scripts have the following triggered field information available as arguments (args):

Argument	Description
associatedToAll	Whether the field is associated with all or some incidents. Value: true or false.
associatedTypes	An array of the incident types, with which the field is associated.
cliName	The name of the field when called from the command line.
description	The description of the field.
isReadOnly	Specifies whether the field is non-editable. Value: true or false.
name	The name of the field.
new	The new value of the field.
old	The old value of the field.
ownerOnly	Specifies that only the creator of the field can edit. Value: true or false.
placeholder	The placeholder text.
required	Specifies whether this is a mandatory field. Value: true or false.
selectValues	If this is a multi-select type field, these are the values the field can take.

Argument	Description
system	Whether it is a Cortex XSOAR defined field.
type	The field type.
unmapped	Whether it is not mapped to any incident.
useAsKpi	Whether it is being used for tracking KPI on an incident page.
user	The username of the user who triggered the script.
validationRegex	Whether there is a regex associated for validation the values the field can hold.

#### Script limitations

- Trigger scripts can't close incidents.
- Post-processing scripts can modify an incident, but if a modified field has a trigger script, it is not called.
- Incident modifications executed within a trigger script are only saved to the database after the modifications are completed.

#### Best practices

- Fields that can hold a list (related incidents, multi-select/tag/role type custom fields) will provide an array of the delta. For example, if a multi-select field value has changed from ["a"] to ["a", "b"], the new argument of the script will get a value of ["b"].
- Incident field trigger scripts run as a batch. This means that if multiple incidents are changed in the same way and are set to trigger the same action, it will happen in one batch.
- When writing incident field trigger scripts, avoid scenarios that call the scripts endlessly (for example, a change in field A triggers script X, which changes field B's value, which in turn calls script Y, which changes field A's value).

#### Add an incident field trigger script to an incident field

After creating an incident field trigger script in the Scripts page in Python, PowerShell, or JavaScript, you can then associate it with an incident field.

- Go to Settings & Info → Settings → Indicators → Fields.
- Select the incident field and click Edit.
- In the Attributes tab, under Script to run when field value changes, select the desired indicator field trigger script.

#### NOTE:

Incident field trigger scripts must have the **field-change-triggered** tag to appear in the list.

#### Field-change-triggered with Single Select or Multi Select types

- Go to Settings & Info → Settings → Object Setup → Incident → Incident Fields.
- Click New and create a new Incident field of one of the following types:
  - Single select
  - Multi-select

- Click Basic Settings and in the Values section set the values you want to see in the incident layout dropdown list for this field.

For example, instance1\_id,instance2\_id,instance3\_id,instance4\_id,instance^,id.

- Click Attributes and in Script to run when field value changes, select the script.

Example 3.

This is an example of a single select script.

```
# The custom mapping made for the field
mapping_dict = {
    'instance1_id' : '123456',
    'instance2_id' : '12340987',
    'instance3_id' : '79874534',
    'instance4_id' : '90927834',
    'instance5_id' : '4543452',
}

val = demisto.args()['new'] # when the script will be triggered this field will hold the new value chosen by the user.
mapped_val = mapping_dict.get(val, val) # getting the value from the map.
execute_command('setIncident', {'customFields' :{'Single_select_field_example': mapped_val}}) # set the new incident mapped field
```

Example 4.

This is an example of a multi-select script.

```
mapping_dict = {
    'low' : '1',
    'medium' : '2',
    'high' : '3',
    'critical' : '4',
}

vals = argToList(demisto.args()['new']) # The new value from the user.
mapped_list = [mapping_dict.get(v, v) for v in vals]
execute_command('setIncident', {'customFields' : {'multi_select_field_example': mapped_list}})
```

#### NOTE:

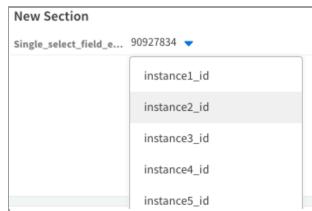
- When creating the script, in the Tags section, type field-change-triggered.
- Choose the name of your custom fields to replace 'Single\_select\_field\_example' or 'multi\_select\_field\_example' in the examples above.

5. Go to Settings & Info → Settings → Object Setup → Incidents → Layouts and add the new incident field to an existing layout or create a new layout.

6. In the incident layout edit page, click Fields and Buttons and drag the new incident field you created to the layout.

7. Save the version.

In the layout display, you will see the values you set in step 3.



8. Select one of the values. The layout will update with the mapped value as set on the script related to the incident field.

Use scripts with a grid field

You can use scripts to manipulate and populate data in the Grid field. In this example, you want analysts who can add comments for the incident during their shift and use a script to automatically populate the Date Logged column with the current date when a user adds a new row to the grid.

1. Create a script called ShiftSummariesChange. The script operates in the following phases:

- The script gets all new rows and sets the Date Logged field to now (current day).
- For each existing row, if the name matches, and the findings column is not updated, the Date Logged column is also updated.
- After creating a grid field, it is saved with the new values using the setIncident command.

```

var newField = args.new ? JSON.parse(args.new) : [];
//if line(s) added, set "datelogged" to now.
if (oldField.length < newField.length) {
    // for each new line change date.
    for(var i=oldField.length; i < newField.length; i++) {
        newField[i].datelogged = new Date().toISOString();
    }
}
var columnName = "findings";
// for each old line if the "columnName" has changed, change date to now.
for(var i=0; i < oldField.length; i++) {
    if (newField[i] && oldField[i].fullname === newField[i].fullname &&
        oldField[i][columnName] !== newField[i][columnName]) {
        newField[i].datelogged = new Date().toISOString();
    }
}
var newVal = {};
newVal[args.cliName] = newField;
executeCommand("setIncident", newVal);

```

2. Add the **field-change-triggered** tag and save the script.

3. Create a Shift Summaries Grid field with the following columns:

- Full name
- Findings
- Status
- Date Logged

Select Date picker with the Lock checkbox, so the script can populate the values for that column. If a column is unlocked (default), the column values can be entered manually (by users), or by a script.

#### NOTE:

Ensure that User can add rows is selected.

4. Add the grid field to a layout, which is attached to an incident type.

Add a row to a grid

During playbook execution, if a malicious finding is discovered you may want to add that finding to a grid. You can use a script in the playbook to add a new row to the grid with the malicious finding.

This is a Python script, which requires two arguments:

- fieldCliName**: The machine name for the field for which you want to add a new row.
- Row**: The new row to add the grid. This is a JSON object in lowercase characters, with no white space.

```

fieldCliName = demisto.args().get('field')
currentValue = demisto.incidents()[0]["CustomFields"][fieldCliName];

if currentValue is None:
    currentValue = [json.loads(demisto.args().get('row'))]
else:
    currentValue.append(json.loads(demisto.args().get('row')))

val = json.dumps({ fieldCliName: currentValue })
demisto.results(demisto.executeCommand("setIncident", { 'customFields': val }))

```

Incident field changes using SLA scripts

You can create scripts that perform specific actions when the SLA is breached in an incident field. For example, you can use the `SendEmailOnSLABreach` script that sends an email to specific users when the script is triggered. For more information, see Automate changes to incident fields using SLA scripts.

#### 9.2.3.2 | Create dynamic fields

##### Abstract

Create dynamic incident fields using an automation script. Create conditional fields.

Dynamic fields can display different data depending on the field value. You can control which fields display in an incident layout, new/edit, and close forms, and which values display for single-select and multi-select fields. You create a script on the Scripts page and then add the script to a field. Scripts support

JavaScript, Python, and PowerShell.

Dynamic fields are useful in the following scenarios:

- You want specific values to appear in a field when the value of another field is different. For example, if the value in the Owner field is **Admin**, the values in the assignee field should be **Jane**, **Joe**, or **Bob**. If the value in the Owner field is anything else, the values in the assignee field should be **Mark**, **Jack**, or **Christine**.
- You can use display scripts to change the value displayed in single-select or multi-select fields in the layout. The field displays a list of options, but when selected, the field may show a different value in the layout than the one selected. For example, in a single-select field, select an incident from a list of incident names, but the field is populated with the incident ID (not the name) of the related incident.
- When assigning an incident to a user, you want to see only relevant data according to the user's role.

#### 1. Create a script.

a. Go to the Scripts page and select New Script.

b. Give the script a descriptive name.

c. Enter a useful description.

d. Under Tags, select **field-display**.

This tag must be applied for the script to be available in the field you want to add the script.

e. Write the script.

Cortex XSOAR comes out-of-the-box with the **hideFieldsOnNewIncident** field-display script, which hides the incident field for new incidents, but appears when editing an incident.

The field script contains the following.

Name	Description
<b>demisto.incidents</b>	The incident in which this script is running.
<b>field</b>	The field attributes. Add metadata to the field, such as <b>cliName</b> , <b>type</b> , <b>select values</b> , etc. For example, <code>[‘field’] [‘cliName’]</code> is the machine learning name of the field.
<b>formType</b>	Enables Cortex XSOAR to process the script in the <b>new</b> , <b>edit</b> , <b>close</b> incident forms. For example, you may want the field to appear in the close form and not in the edit form.
<b>incident.get (‘field’)</b>	The field within the incident. For example, <code>incident.get.(‘owner’)</code> retrieves the <b>owner</b> field. If you create a custom field, you need to change this to <b>CustomFields</b> . For example, for the <b>incidentclassification</b> custom field, type:  <code>if incident.get('CustomFields').get('incidentclassification') .</code>
<b>demisto.results</b>	The results to return.
<b>currentUser</b>	Specifies the current user. For example, if you want the script to check on a role assigned to user and display the appropriate output, type the following:  <code>demisto.executeCommand("getUserByUsername", {"username": demisto.args()["currentUser"] })</code>  Add the information that you want to display according to the user roles.

#### 2. Create an incident field.

a. Select Settings & Info → Settings → Object Setup → Incidents → Incident Fields → New.

If you want to add the script to an existing field, select the field and click Edit.

- b. Under Field Type, select the field type. For example, Single select.
- c. Under Field Name, enter a descriptive name.
- d. Under the Attributes tab, in the Field display script field, select the script you created in step 1.
- e. Complete the remaining field definitions Save the field.

#### Change field values according to groups

The following example shows how to create a script for the Assignee field, which shows different values depending on the values in the Owner field. If the Owner is defined as admin, and the list of available assignees includes one group. If the Owner is defined as anything else, the list of available assignees includes a different group.

1. In the Scripts page, copy the `hideFieldsOnNewIncident` and name it `changeAssigneesPerOwner`.

2. In the Description field, enter the following:

Changes values available in the Assignees field based on the person defined as the owner.

3. Under Tags, add the `field-display` tag.

4. For the script, type the following:

```
incident = demisto.incidents()[0]
field = demisto.args()['field']['cliName']
if incident.get('owner') == 'admin':
    demisto.results({'hidden': False, 'options': ['jane','joe', 'bob']})
else:
    demisto.results({'hidden': False, 'options': ['mark','jack', 'christine']})
```

where

- `demisto.incidents` is the incident in which the script is running.
- `incident.get('owner')` is the field within the incident.
- `demisto.results` tells us whether to hide the field or not, and which values should appear in the field. When the `owner` field is `Admin`, the values are `Jane`, `Joe`, `Bob`. When the `owner` is anyone else, the values are `Mark`, `Jack`, `Christine`.

5. Select Settings & Info → Settings → Object Setup → Incidents → Incident Fields → New .

- Name the field `Assign To`.

The Values field in the Basic Settings tab has been left blank because we hard-coded the values in our script.

- Under the Attributes tab, in the Field display script field, select the `changeAssigneesPerOwner` script we created above.
- Fill in the rest of the field definitions as desired and click Save.

6. Add the field to an incident layout.

7. Create an incident to see what happens when the Owner is set to `Admin` and when the Owner is set to anything else.

#### Hide a field based on context data

In this example, you need to hide a field in the new incident form but display the field when editing the form. You also set field values for a multi-select field in the case of an existing incident.

Before you begin, download the GDPR content pack.

In this example, use the `hideFieldsOnNewIncident` out-of-the-box script.

```
incident = demisto.incidents()[0]
field = demisto.args()['field']
formType = demisto.args()['formType']
if incident["id"] == "":
    # This is a new incident, hide the field
    demisto.results({"hidden": True, "options": []})
else:
    # This is an existing incident, we want to show the field, to know which values to display
    options = []
    # The field type includes the word select, such as Single select or Multi select
    if "Select" in demisto.get(field, "type"):
        # take the options from the field definition
        options = demisto.get(field, "selectValues")
    demisto.results({"hidden": False, "options": options})
```

1. Go to Settings & Info → Settings → Object Setup → Incidents → Incident Fields.

2. Select the `Malicious Cause (if the cause is a malicious attack)` field and click Edit.

3. Under the Field display script field, select the **hideFieldsOnNewIncident** script and click Save.

4. Go to the Incidents page and click New Incident.

5. Under the Type field, select **GDPR DataBreach**.

Scroll down and note that under Mandatory Information, there is no **Malicious Cause** field.

6. Click Create New Incident to save the incident.

7. Select the incident you just created and click Edit.

Scroll down to the Mandatory Information section and note that the **Malicious Cause** field appears and the options for the field are retrieved from the initial field definition.

#### 9.2.3.3 | Troubleshoot incident fields

##### Abstract

Troubleshoot issues when creating incident fields.

###### Troubleshoot conflicts with custom incident fields

When trying to download a content update, you may receive the following message:

Warning: content update has encountered some conflicts

This occurs when a content update has an incident field with the same name as a custom incident field that already exists in Cortex XSOAR.

To resolve this issue, perform the following steps:

Click Install Content to force the update and retain your custom incident field. The content update will install without the system version of the incident field.

###### Troubleshoot closing a case incident after changing the field type

After deleting a field of type Grid (table) and creating a new field of another type (string, long text, etc.), you may receive the following error when trying to close or update an incident:

Cannot convert type []interface {} of '[map[] map[]]' to type string, field: sourceip (8902)

This error occurs with field type changes, if the fields are not compatible types, such as changing the type from long text to boolean or URL to short text. If you create an incident with that field, delete the field, create a new field with the same name but a different type, and then try to close the incident with that field, the error occurs.

For example, create a field of type table/grid and associate it with an incident type. Create an incident with that field, delete that table/grid field, and create a new field with the same name but associate it with a different type (such as short text). When you try to close the incident that has that field, an error may occur.

To resolve this issue, perform the following steps:

1. Go to Settings & Info → Settings → Object Setup → Incidents → Types.

2. Select the incident type that contains the changed field.

3. Click Edit Layouts.

4. Select the tab you want to edit.

5. Add the field you changed to the layout.

6. Save the form.

7. Go to the Incidents page and select the incident.

8. Click Close if you want to close the incident or Edit if you want to edit the incident.

9. In the **Custom Fields** area, reset (delete) the value for the field you changed.

10. Click Close Incident or Update Incident.

#### 9.2.4 | Incident layout customization

##### Abstract

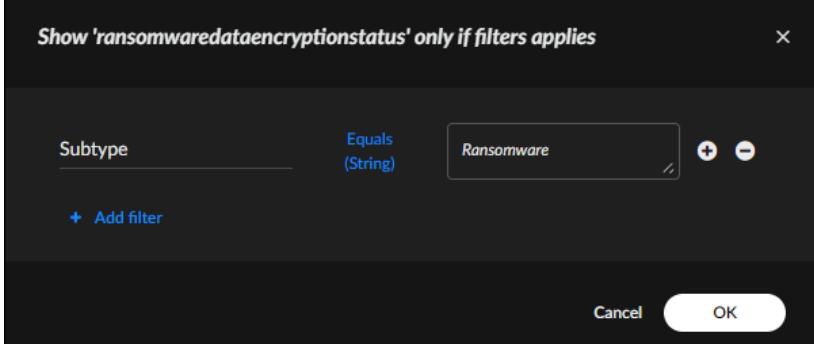
Customize incident layouts in Cortex XSOAR to view relevant information.

Each incident type has a unique set of data relevant to that specific incident type, including layouts. It is important to display the most appropriate data for users. Each out-of-the-box incident comes with a layout. You can customize almost every aspect of the layout, including which tabs appear, in which order they appear, who has permission to view the tabs, what information appears, and how it is displayed.

It's important to build or customize the layout so that you see the information that is relevant to the incident type. For example, in a phishing incident, you may want to see email headers, but not in an access incident. While some information might be appropriate for multiple incident types, its location in one incident may require more prominence than in another incident.

You can see which incident type uses the incident layout in the Types tab under Settings & Info → Settings → Object Setup → Incidents. The incident layout name appears in the Layout column. You can edit the layouts in the Layouts tab.

You can customize the display information including fields for existing incidents, by modifying the sections and fields for the following views:

Section	Description
Incident Summary	<p>The Incident Summary tab displays the information necessary to investigate an incident. You can customize almost every aspect of the layout, including which tabs appear, the order they appear, and who has permission. In each field or tab, you can add filters by clicking on the eye icon, which enables you to add conditions that show specific fields or tabs. For example, if an analyst decides that a Cortex XDR Malware incident is a Ransomware subtype, they may only want fields to appear that show data about the encryption method and not to show information if the Malware subtype is adware.</p>  <p>You may also want to limit specific tabs to certain scenarios. For example, if a user clicks a phishing link, the new tab can contain relevant fields and action buttons for this scenario. You can also add dynamic fields, such as a graph of several bad indicators, their source, and severity. For more information, see Create dynamic fields.</p> <p>Also, you can use queries to filter the information in the dynamic section to suit your exact needs.</p>
New/Edit form	Add, edit, and delete fields and buttons to be displayed when creating or editing an incident.
Close Form	Add, edit, or delete sections, fields, and filters, when closing an incident.
Incident Quick View	Add, edit, and delete sections, fields, and filters in the Incident Quick view section in the incident.

#### NOTE:

There are several out-of-the-box layout sections and fields that you cannot remove, but you can rearrange them in the layout and modify their queries and filters. These layouts need to be duplicated or detached to make changes.

We recommend copying an existing out-of-the-box incident layout so you don't miss any important information.

#### Create an incident layout

1. Go to Settings & Info → Settings → Object Setup → Incidents → Layouts.

You can customize the following tabs:

- Incident Summary
- "New/Edit" Form
- "Close" Form
- Incident Quick View

2. Add a meaningful name for the layout.

3. Customize the tabs by clicking the settings wheel icon and then doing the following:

**NOTE:**

You can click and drag a tab to reorder the tabs.

Action	Description
Rename	You can also edit a tab's name by clicking the tab.
Duplicate	Copies the existing tab.
Delete	Deletes the tab.
Show empty fields	The setting that you configure in the layout becomes the default value seen in the report for the specific tab, which can then be overridden. You can also set a global default value using the <code>UI.summary.page.hide.empty.fields</code> server configuration, which can also be overridden for a specific tab.
Hide tab	Hides the tab. Rather than deleting the tab, you may want to use the tab again for future use.
Format for exporting	Build your layout based on A4 proportions to match the format used for exporting. Selecting this option hides the tab by default, but the tab will remain available for export.
Viewing Permissions	Select which roles can view the tabs.
Display Filter	Add or view a filter applied to the tab. If the filters apply, the specific fields or tabs are shown in the layout. If the mandatory field is not shown in the layout, the user is not obliged to complete it.

4. Do the following:

- Drag and drop the required sections, fields, buttons, and tabs.
- Customize sections and create buttons.
- Add any required filters.
- Create new tabs

5. In the "New/Edit" Form and "Close" Form tabs, drag and drop the required fields and buttons.

You can also edit the Basic Information and the Custom Field sections.

6. Repeat step 3 for the Incident Quick View tab.

7. Save the layout, and add it to the incident type.

**NOTE:**

If the incident type is attached, you need to detach or duplicate it, if you want to add the layout.

8. Create or ingest an incident to test the new layout and verify fields are populated.

9. (Optional) For a customized layout (duplicate or new layout), you can contribute it to the Marketplace.

a. In the Layouts page, right-click the new layout and select Contribute.

b. In the dialog box, select either Save and submit your contribution or Save and download your contribution for later use, which you can view in the Contributions tab in the Marketplace.

If you select Save and submit your contribution your layout is validated and then you are prompted to submit to review. You can also view your contribution in the Marketplace.

#### Edit an incident layout

1. Go to Settings & Info → Settings → Object Setup → Incidents → Layouts.

2. Click the layout name you want to edit.

You can see with the current layout, which is populated with demo data so you can see how the fields fit.

3. If editing a layout that has been installed from a content pack (the layout shows a locked icon), do one of the following:

- Duplicate an incident layout

To add the layout to the incident type, you need to detach the incident type and then add the layout. To duplicate an incident layout, right-click the layout name in the layouts table, and select Duplicate.

- Detach the layout.

When detached, the layout does not receive content pack updates until you reattach it. You do not need to edit the incident type, as the layout name remains the same.

#### TIP:

While a layout is detached, it does not receive content pack updates. If you detach a layout and make changes, any changes made while it was detached are overwritten by the default values from the content pack. If you want to keep the changes and protect your changes from content pack upgrades, duplicate the incident type before reattaching the original.

To detach or reattach an incident layout, right-click the layout name in the layouts table, and select Detach or Attach.

4. Add sections, buttons, and fields, as required.

#### Customize sections

1. Create or edit a layout.

2. From the Sections tab in the Library, drag and drop the following sections:

Section	Description
New Section	After creating a new section, click the <i>Incident type</i> Fields tab and drag and drop the fields as required.
Cortex XSOAR out-of-the-box sections	Out-of-the-box sections such as attachments and evidence.
General Purpose Dynamic Section	You can add a script to the incident layout. For example, assign a script that calculates the total number of entries that exist for an incident, which dynamically updates when new entries are added to the incident.

#### NOTE:

To remove or duplicate a section, select the section, click , and then select Duplicate or Remove.

3. Define section properties by clicking  and then Edit section settings.

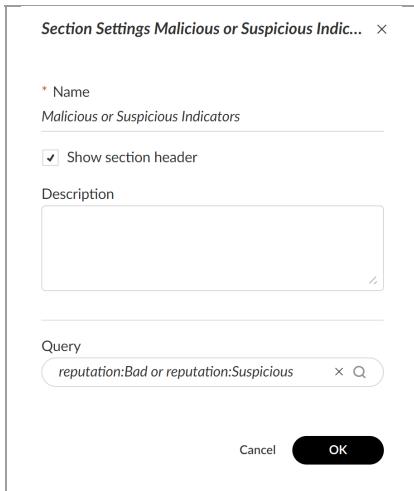
#### TIP:

Limit the number of incident fields to 50 in each section. You can create additional sections as needed.

You can determine how a section in the layout appears in the layout. For example, you may want a section header, or configure the fields to appear in rows or as cards. If some of the field values will be very long, use rows instead of cards. If the field values are short, you might want to use cards so you can fit more fields into a section.

4. If adding the Malicious or Suspicious Indicators section, you can change the information that appears, by editing the Query.

For example, to see all indicators of type IP and with a reputation of Bad that were found by a specific source since January 2nd 2022, enter `Type:IP and reputation:Bad and firstseenbysource:>="2021-01-02T00:00:00 +0200"`



5. Drag and drop fields, and add any filters as required.

6. If relevant, create a new tab and repeat the steps as required.

#### Add a script to the layout

You can add script-based content to the incident layout by adding the General Purpose Dynamic Section in the incident layout builder. The General Purpose Dynamic Section enables you to configure a section in the incident layout from a script. The script can return a simple text, markdown, or HTML, the results of which appear in the General Purpose Dynamic Section.

You can add any required information from a script. For example, you can assign a script that calculates the total number of entries for an incident, which dynamically updates when new entries are added to an incident. You can add a custom widget to the incident page and add note information using a script.

The following are examples of values that can be returned from the General Purpose Dynamic Section script:

##### Text example

```
return_results(<TEXT>)
```

##### Markdown example

```
return_results({
    'ContentsFormat': EntryFormat.MARKDOWN,
    'Type': EntryType.NOTE,
    'Contents': <MARKDOWN_DATA>
})
```

##### HTML example

```
return_results({
    'ContentsFormat': EntryFormat.HTML,
    'Type': EntryType.NOTE,
    'Contents': <HTML_DATA>
})
```

For the `EntryFormat` values see `EntryFormat` in Common Server Python functions.

#### How to add a script to a layout

Before you begin, you need to create a script.

1. Create a script.

For examples of script-based widgets for layouts, see Examples of using scripts in incident layouts

Add the `dynamic-section` tag.

2. Create or edit an incident layout.
3. Drag and drop the General Purpose Dynamic Section into the layout area you want to appear.
4. Select the General Purpose Dynamic Section, click and then click Edit section settings.

5. In the Name and Description fields, add a meaningful name and a description for the dynamic section that explains what the script displays.

6. In the Automation script field, from the dropdown list, select the script that returns data for the dynamic section.

**NOTE:**

Only scripts to which you have added the **dynamic-section** tag appear in the dropdown list.

7. Save the section.

#### Create custom buttons

You can add existing buttons or create buttons and then drag and drop them in the layout.

To add a custom button, create a script and then add the new button to the layout and select a script. These buttons can simplify and assist an analyst in carrying out various tasks. For example, add buttons for an analyst to self-assign an incident, link or unlink an incident, close an incident as a duplicate, or generate a summary report.

For fields (script arguments) that are optional, you can define whether to show them to analysts when they click on buttons. To expose an optional field, select the Ask User checkbox next to the script arguments in the button settings page.

In the following example, add a button to the layout, which self-assigns an incident for an analyst. The Common Scripts content pack includes an **AssignToMeButton** Script.

**NOTE:**

When creating a script for use in an incident layout, the **incident-action-button** tag must be assigned for the script to be available for custom buttons.

The script that runs when an action button is clicked accepts only mandatory arguments through the pop-up window and does not provide an option for any non-mandatory arguments to be filled in when the button is clicked. It is recommended to use a wrapper script to collect and validate arguments in scenarios where there can be a combination of mandatory and non-mandatory arguments for a button.

In the following example, create a button to self-assign an incident for an analyst, and add it to a layout.

1. Create or edit a layout.
2. Create a new section.
3. In the Fields and Buttons tab, drag and drop the New Button into the relevant section.
4. Click to configure.
5. In the Button Display Name, enter Assign to Me.
6. Select a color if required.
7. In the Button Script field, add the **AssignToMeButton** script.
8. Save the Button settings and then save the layout.

**NOTE:**

The Case Management - Generic content pack includes several buttons to use in a layout, such as Assign to Me, Close as Duplicate, and Link incidents. You can also see useful case management incident layouts. For more information, see Case Management - Generic content pack.

#### Add the layout to an incident type

1. Go to Settings & Info → Settings → Object Setup → Incidents → Types.
2. Select the incident type and click Edit.
3. In the Layout field, from the dropdown list, add the customized layout.
4. (Optional) For a customized layout, you can contribute it to the Marketplace.
  - a. In the Layouts page, select the new layout and then click Contribute to Marketplace.
  - b. In the dialog box select either Save and submit your contribution or Save and download your contribution for later use, which you can view in the Contributions tab in Marketplace.

If you select Save and submit your contribution, your layout is validated and you are prompted to submit to review. You can also view your contribution in Marketplace.

#### Add incident access control fields to the layout

In any SOC team, there are various roles and responsibilities. For example, you may have specific teams to deal with threats, such as threat intelligence researchers, security analysts (Tier 1), senior analysts (Tier 2), SOC leads, SOC managers, and SIEM engineers. You have various options to limit access to incidents and investigations.

- Exclude access to incident actions and investigations according to roles using role-based permissions. For example, you may want to limit the ability to change the incident status, or manage the Work plan. For more information, see Role-based permissions.
- Restrict an investigation according to team members. In an investigation, the owner of the incident can restrict the incident to team members only.

Analysts can select Restrict incident (under Actions) to restrict an incident.

#### **NOTE:**

If using a script, use the `restrictInvestigation` command. You need to specify the incident ID of the incident and set the `set` the `Restrict` argument to `True` to restrict the incident or set the `Restrict` argument to `False` to remove restricted from the incident.

- Limit access to investigations, by doing the following:

- Limit investigations according to specific user roles

You can add the `Roles` field to the layout, which enables you to restrict access to all roles other than those you have specifically added. For example, after an investigation is closed, add administrators or those with specialty roles, so only they can reopen or link incidents. The added roles have read and write permission, but all other roles do not have access (unless you have added them in the read-only field).

#### **NOTE:**

- You can also run the `/incident_set roles=<name of role>` or `!setIncident roles =<name of role>` in the CLI, playbook, or script to set the role.
- If you add a role, but the incident has been restricted to team members, and the user is not a team member, the user cannot access the incident regardless of the role. For example, if you restrict the incident to User A and User B team members who are Tier 1 analysts but then try to add Tier 2 analysts (none of whom are team members) to the list of roles, a Tier 2 analyst cannot access the incident.

- Give read-only access to certain user roles

You add the XSOAR Read Only Roles field to the layout, which restricts access to the incident. When granting read-only access, the user can view the incident but not edit. For example, when an incident is in triage (phase 1), you may want all Tier-2 analysts to have read-only access, so that Tier-1 can edit the incident. When the phase changes to phase 2, Tier-1 has read-only access.

Adding a team member overrides the XSOAR Read Only Roles field, so if you add User A, (Tier 1) as a team member, even if you assign Tier-1 as a Read only role, the user still has Read/Write access. You need to remove the user as a Team Member.

Although an analyst can change the XSOAR Read-Only field manually, you can automate the process by creating a custom incident field using Incident Field Trigger Scripts, or create a script and adding a new field button.

#### **NOTE:**

You can also run the `!setIncident xsoarReadOnlyRoles=<name of role>` in the CLI, playbook, or script to set the the user role.

If you assign a role (read and write permission) and assign the same role as read only, the user still has read/write permission. You need to remove the assigned role. If you restrict the incident, the read-only role does not override the restriction. In other words, team members permission takes precedence.

### 9.2.4.1 | Examples of using scripts in incident layouts

#### Abstract

Examples of using scripts in incident layouts in Cortex XSOAR.

The following are examples of scripts that are supported in incident layouts:

#### Charts

A valid result for a chart widget is a list of groups. Each group points to a single entity. For example, in bar charts, each group is a bar. A group consists of the following:

- Name: A string.
- Data: An array of integers.
- Color: A string representing a color that will be used as a default color for that group. It can be the name of the color, a hexadecimal representation of the color, or an RGB color value (optional).
- Groups: A nested list of groups (optional).

#### Horizontal bar

In this example, create a script in Python that displays a horizontal bar of the indicators by severity.

#### Python script

```
data = {
    "Type": 17,
    "ContentsFormat": "bar",
```

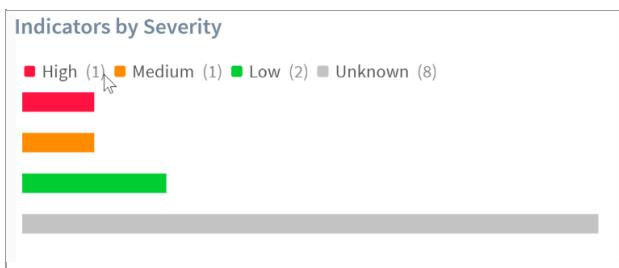
```

"Contents": {
  "stats": [
    {
      "data": [
        1
      ],
      "groups": None,
      "name": "high",
      "label": "incident.severity.high",
      "color": "rgb(255, 23, 68)"
    },
    {
      "data": [
        1
      ],
      "groups": None,
      "name": "medium",
      "label": "incident.severity.medium",
      "color": "rgb(255, 144, 0)"
    },
    {
      "data": [
        2
      ],
      "groups": None,
      "name": "low",
      "label": "incident.severity.low",
      "color": "rgb(0, 205, 51)"
    },
    {
      "data": [
        8
      ],
      "groups": None,
      "name": "unknown",
      "label": "incident.severity.unknown",
      "color": "rgb(197, 197, 197)"
    }
  ],
  "params": {
    "layout": "horizontal"
  }
}
}

demisto.results(data)

```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



Vertical bar

In this example, create a script in Python that displays a vertical bar of the indicators by severity.

#### Python script

```

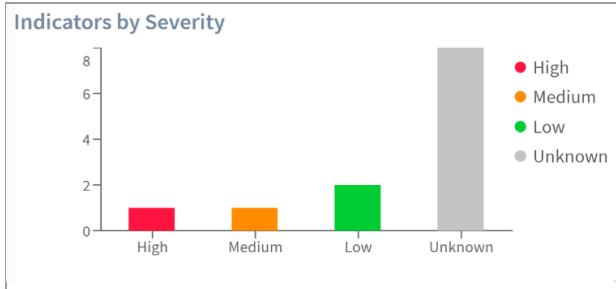
data = {
  "Type": 17,
  "ContentsFormat": "bar",
  "Contents": {
    "stats": [
      {
        "data": [
          1
        ],
        "groups": None,
        "name": "high",
        "label": "incident.severity.high",
        "color": "rgb(255, 23, 68)"
      },
      {
        "data": [
          1
        ],
        "groups": None,
        "name": "medium",
        "label": "incident.severity.medium",
        "color": "rgb(255, 144, 0)"
      },
      {
        "data": [
          2
        ],
        "groups": None,
        "name": "low",
        "label": "incident.severity.low",
        "color": "rgb(0, 205, 51)"
      },
      {
        "data": [
          8
        ],
        "groups": None,
        "name": "unknown",
        "label": "incident.severity.unknown",
        "color": "rgb(197, 197, 197)"
      }
    ]
  }
}

```

```
{
  "data": [
    {
      "name": "low",
      "label": "incident.severity.low",
      "color": "rgb(0, 205, 51)"
    },
    {
      "name": "unknown",
      "label": "incident.severity.unknown",
      "color": "rgb(197, 197, 197)"
    }
  ],
  "params": {
    "layout": "vertical"
  }
}
}

demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



Stacked bar

In this example, create a script in Python that displays a stacked bar showing the successes and failures on specific dates.

#### Python script

```
data = {
  "Type": 17,
  "ContentsFormat": "bar",
  "Contents": [
    {
      "stats": [
        {
          "name": "time1",
          "groups": [
            [
              {
                "name": "Successes",
                "data": [7],
                "color": "rgb(0, 205, 51)"
              },
              {
                "name": "Failures",
                "data": [3],
                "color": "rgb(255, 144, 0)"
              }
            ]
          ]
        },
        {
          "name": "time2",
          "groups": [
            [
              {
                "name": "Successes",
                "data": [9],
                "color": "rgb(0, 205, 51)"
              },
              {
                "name": "Failures",
                "data": [4],
                "color": "rgb(255, 144, 0)"
              }
            ]
          ]
        }
      ],
      "params": {
        "layout": "horizontal"
      }
}
```

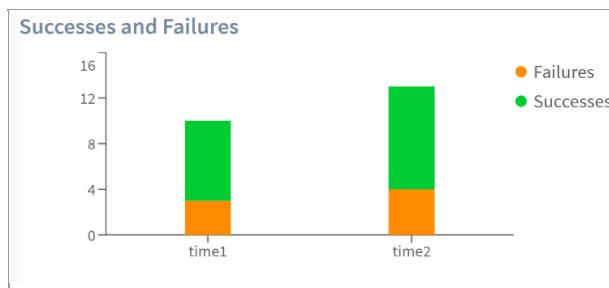
```

        }
    }
}

demisto.results(data)

```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



Line chart

In this example, we create a JavaScript that displays how many GitHub issues were created each week for Content, Documentation, and Platform in a line chart.

#### Java script

```

content = 'red',
platform = 'yellow'
documentation = 'blue'
data = {
    "Type": 17,
    "ContentsFormat": "line",
    "Contents": {
        "stats": [
            {
                "count": 3,
                "data": [
                    3
                ],
                "floatData": [
                    3
                ],
                "groups": [
                    {
                        "count": 3,
                        "data": [
                            3
                        ],
                        "floatData": [
                            3
                        ],
                        "groups": null,
                        "name": "Content",
                        "color": content
                    }
                ],
                "name": "2020-35"
            },
            {
                "count": 32,
                "data": [
                    32
                ],
                "floatData": [
                    32
                ],
                "groups": [
                    {
                        "count": 11,
                        "data": [
                            11
                        ],
                        "floatData": [
                            11
                        ],
                        "groups": null,
                        "name": "Content",
                        "color": content
                    },
                    {
                        "count": 20,
                        "data": [
                            20
                        ],
                        "floatData": [
                            20
                        ],
                        "groups": null,

```

```
        "name": "Platform",
        "color: platform
    },
    {
        "count": 1,
        "data": [
            1
        ],
        "floatData": [
            1
        ],
        "groups": null,
        "name": "Documentation",
        "color: documentation
    }
],
"name": "2020-36"
},
{
    "count": 25,
    "data": [
        25
    ],
    "floatData": [
        25
    ],
    "groups": [
        {
            "count": 15,
            "data": [
                15
            ],
            "floatData": [
                15
            ],
            "groups": null,
            "name": "Platform",
            "color: platform
        },
        {
            "count": 10,
            "data": [
                10
            ],
            "floatData": [
                10
            ],
            "groups": null,
            "name": "Content",
            "color: content
        }
    ],
    "name": "2020-37"
},
{
    "count": 38,
    "data": [
        38
    ],
    "floatData": [
        38
    ],
    "groups": [
        {
            "count": 18,
            "data": [
                18
            ],
            "floatData": [
                18
            ],
            "groups": null,
            "name": "Platform",
            "color: platform
        },
        {
            "count": 20,
            "data": [
                20
            ],
            "floatData": [
                20
            ],
            "groups": null,
            "name": "Content",
            "color: content
        }
    ],
    "name": "2020-38"
},
{
    "count": 48,
    "data": [

```

```
        48
    ],
    "floatData": [
        48
    ],
    "groups": [
        {
            "count": 23,
            "data": [
                23
            ],
            "floatData": [
                23
            ],
            "groups": null,
            "name": "Content",
            color: content
        },
        {
            "count": 25,
            "data": [
                25
            ],
            "floatData": [
                25
            ],
            "groups": null,
            "name": "platform",
            color: platform
        }
    ],
    "name": "2020-39"
},
{
    "count": 59,
    "data": [
        59
    ],
    "floatData": [
        59
    ],
    "groups": [
        {
            "count": 29,
            "data": [
                29
            ],
            "floatData": [
                29
            ],
            "groups": null,
            "name": "platform",
            color: platform
        },
        {
            "count": 30,
            "data": [
                30
            ],
            "floatData": [
                30
            ],
            "groups": null,
            "name": "Content",
            color: content
        }
    ],
    "name": "2020-40"
},
{
    "count": 41,
    "data": [
        41
    ],
    "floatData": [
        41
    ],
    "groups": [
        {
            "count": 20,
            "data": [
                20
            ],
            "floatData": [
                20
            ],
            "groups": null,
            "name": "Content",
            color: content
        },
        {
            "count": 21,
            "data": [

```

```
        21
    ],
    "floatData": [
        21
    ],
    "groups": null,
    "name": "Platform",
    color: platform
}
],
"name": "2020-41"
},
{
"count": 41,
"data": [
    41
],
"floatData": [
    41
],
"groups": [
    {
        "count": 23,
        "data": [
            23
        ],
        "floatData": [
            23
        ],
        "groups": null,
        "name": "Content",
        color: content
    },
    {
        "count": 18,
        "data": [
            18
        ],
        "floatData": [
            18
        ],
        "groups": null,
        "name": "Platform",
        color: platform
    }
],
"name": "2020-42"
},
{
"count": 48,
"data": [
    48
],
"floatData": [
    48
],
"groups": [
    {
        "count": 18,
        "data": [
            18
        ],
        "floatData": [
            18
        ],
        "groups": null,
        "name": "Content",
        color: content
    },
    {
        "count": 30,
        "data": [
            30
        ],
        "floatData": [
            30
        ],
        "groups": null,
        "name": "Platform",
        color: platform
    }
],
"name": "2020-43"
},
{
"count": 34,
"data": [
    34
],
"floatData": [
    34
],
"groups": [

```

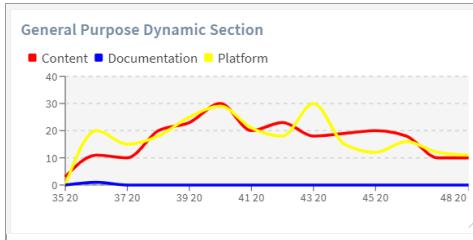
```
{  
    "count": 19,  
    "data": [  
        19  
    ],  
    "floatData": [  
        19  
    ],  
    "groups": null,  
    "name": "Content",  
    "color: content  
},  
{  
    "count": 15,  
    "data": [  
        15  
    ],  
    "floatData": [  
        15  
    ],  
    "groups": null,  
    "name": "Platform",  
    "color: platform  
}  
,  
{"name": "2020-44"  
},  
{  
    "count": 32,  
    "data": [  
        32  
    ],  
    "floatData": [  
        32  
    ],  
    "groups": [  
        {  
            "count": 12,  
            "data": [  
                12  
            ],  
            "floatData": [  
                12  
            ],  
            "groups": null,  
            "name": "Platform",  
            "color: platform  
        },  
        {  
            "count": 20,  
            "data": [  
                20  
            ],  
            "floatData": [  
                20  
            ],  
            "groups": null,  
            "name": "Content",  
            "color: content  
        }  
    ],  
    "name": "2020-45"  
},  
{  
    "count": 34,  
    "data": [  
        34  
    ],  
    "floatData": [  
        34  
    ],  
    "groups": [  
        {  
            "count": 16,  
            "data": [  
                16  
            ],  
            "floatData": [  
                16  
            ],  
            "groups": null,  
            "name": "Platform",  
            "color: platform  
        },  
        {  
            "count": 18,  
            "data": [  
                18  
            ],  
            "floatData": [  
                18  
            ],  
            "groups": null,  
            "name": "Content",  
            "color: content  
        }  
    ],  
    "name": "2020-46"  
},  
{"name": "2020-47"  
},  
{"name": "2020-48"}  
}
```

```
        "name": "Content",
        color: content

    },
    "name": "2020-46"
},
{
    "count": 22,
    "data": [
        22
    ],
    "floatData": [
        22
    ],
    "groups": [
        {
            "count": 12,
            "data": [
                12
            ],
            "floatData": [
                12
            ],
            "groups": null,
            "name": "Platform",
            color: platform
        },
        {
            "count": 10,
            "data": [
                10
            ],
            "floatData": [
                10
            ],
            "groups": null,
            "name": "Content",
            color: content
        }
    ],
    "name": "2020-47"
},
{
    "count": 21,
    "data": [
        21
    ],
    "floatData": [
        21
    ],
    "groups": [
        {
            "count": 11,
            "data": [
                11
            ],
            "floatData": [
                11
            ],
            "groups": null,
            "name": "Platform",
            color: platform
        },
        {
            "count": 10,
            "data": [
                10
            ],
            "floatData": [
                10
            ],
            "groups": null,
            "name": "Content",
            color: content
        }
    ],
    "name": "2020-48"
},
{
    "params": {
        "groupBy": [
            "gitcreated(w)",
            "gitteam"
        ],
        "timeFrame": "weeks"
    }
}
}

return data;
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



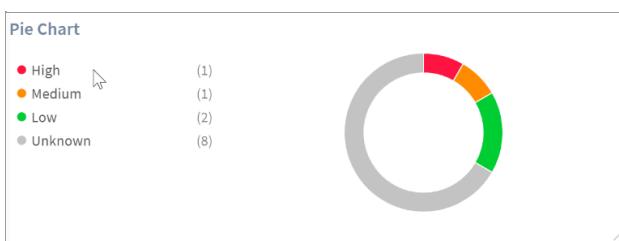
Pie

In this example, create a script in Python that queries and returns a pie chart.

```
data = {
    "Type": 17,
    "ContentsFormat": "pie",
    "Contents": {
        "stats": [
            {
                "data": [
                    1
                ],
                "groups": None,
                "name": "high",
                "label": "incident.severity.high",
                "color": "rgb(255, 23, 68)"
            },
            {
                "data": [
                    1
                ],
                "groups": None,
                "name": "medium",
                "label": "incident.severity.medium",
                "color": "rgb(255, 144, 0)"
            },
            {
                "data": [
                    2
                ],
                "groups": None,
                "name": "low",
                "label": "incident.severity.low",
                "color": "rgb(0, 205, 51)"
            },
            {
                "data": [
                    8
                ],
                "groups": None,
                "name": "unknown",
                "label": "incident.severity.unknown",
                "color": "rgb(197, 197, 197)"
            }
        ],
        "params": {
            "layout": "horizontal"
        }
    }
}

demisto.results(data)
```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays indicator severity as a pie chart:



Duration

In this example, create a script in Python that queries and returns a time duration (specified in seconds), and displays the data as a countdown clock.

```
data = {
    "Type": 17,
```

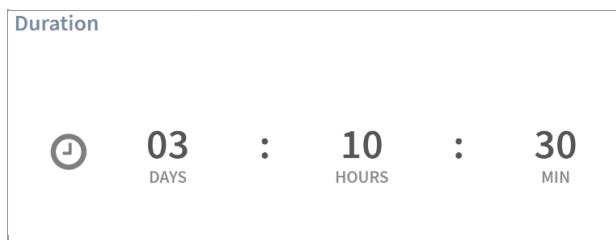
```

"ContentsFormat": "duration",
"Contents": {
  "stats": 60 * (30 + 10 * 60 + 3 * 60 * 24),
  "params": {
    "layout": "horizontal",
    "name": "Lala",
    "sign": "@",
    "colors": {
      "items": {
        "#00CD33": {
          "value": 10
        },
        "#FAC100": {
          "value": 20
        },
        "green": {
          "value": 40
        }
      }
    },
    "type": "above"
  }
}
}

demisto.results(data)

```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays the time duration:



Number

This example shows how to create a single item widget that displays a number.

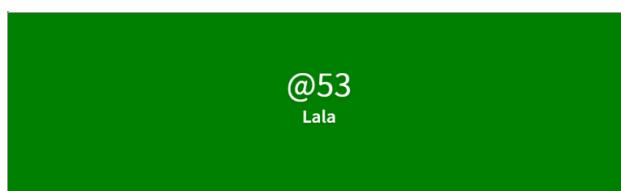
```

data = {
  "Type": 17,
  "ContentsFormat": "number",
  "Contents": {
    "stats": 53,
    "params": {
      "layout": "horizontal",
      "name": "Lala",
      "sign": "@",
      "colors": {
        "items": {
          "#00CD33": {
            "value": 10
          },
          "#FAC100": {
            "value": 20
          },
          "green": {
            "value": 40
          }
        }
      },
      "type": "above"
    }
  }
}

demisto.results(data)

```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



Number Trend

This example shows how to create a single-item widget that displays a number trend.

```

data = {
    "Type": 17,
    "ContentsFormat": "number",
    "Contents": {
        "stats": { "prevSum": 53, "currSum": 60 },
        "params": {
            "layout": "horizontal",
            "name": "Lala",
            "sign": "@",
            "colors": {
                "items": {
                    "#00CD33": {
                        "value": 10
                    },
                    "#FAC100": {
                        "value": 20
                    },
                    "green": {
                        "value": 40
                    }
                }
            },
            "type": "above"
        }
    }
}

demisto.results(data)

```

After you have uploaded the script and created the widget, you can add the widget to an incident layout. The following widget displays:



Add note information

This example shows how to add note information to an incident layout using a script through the API.

1. Install the Cortex REST API content pack and add a Core REST API instance.

2. Go to the Scripts page and add the following script:

```

commonfields:
  id: ShowLastNoteUserAndDate
  version: -1
name: ShowLastNoteUserAndDate
script: |2
function getLastNote(incidentID) {
    var body = {pageSize:1,categories:['notes']};
    var res = executeCommand('demisto-api-post', {uri:'/investigation/' + incidentID, body: body});
    if (isError(res[0])) {
        throw 'demisto-api-post failed for incident #' + incidentID + '\nbody is ' + JSON.stringify(body) + '\n' + JSON.stringify(res);
    }
    if (!res[0].Contents.response.entries) {
        return null;
    }
    var notes = res[0].Contents.response.entries;
    var lastNote = notes[notes.length-1];
    return lastNote;
}

lastNote = getLastNote(incidents[0].id);

if (lastNote) {
    md = ##### Update by ${lastNote.user} on ${lastNote.modified.split('T')[0]}\n`;
    md += `n--n`;
    md += lastNote.contents + `n`;

    return { ContentsFormat: formats.markdown, Type: entryTypes.note, Contents: md } ;
} else {
    return 'N/A';
}
type: javascript
tags:
- dynamic-section
enabled: true
scripttarget: 0
runonce: false
runas: DBotWeakRole

```

3. Add the script to the layout and then add the layout to the incident type.

4. Go to the incident to view the note information.

You can see note information, containing the last user and date.

**showlastnote**

Update by admin on 2020-08-10

---

Incident was modified.

Field changes

Field	Owner
Old Value	admin

## 9.3 | Classification and mapping

### Abstract

Classify and map an integration instance.

The classification and mapping feature enables you to take events and event information ingested from integrations, classify the event as an incident type, and map event information to incident fields in Cortex XSOAR.

#### NOTE:

- Classifiers and mappers can be created with or without association to a specific integration instance and can be assigned to multiple instances. An integration can only have a classifier or only a mapper.
- When creating a classifier and mapper, you can contribute them to the Marketplace.

For more information about classification and mapping, see the following video:

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

### Classification

Classification determines the type of incident that is created for events ingested from a specific integration. For example, Cortex XSOAR might generate alerts from Cortex Traps which you would classify either as a dedicated Traps, Authentication, or Malware incident type.

By classifying the events as different incident types, you can process them with different playbooks in the incident type, which is suited to their respective requirements.

You have the following options for classification:

- You can hard code every alert fetched from the integration to a specific incident type, by selecting the incident type in the integration instance settings. This is useful where you want all alerts classified to a single type, such as phishing and have the same playbook execute on the same incident.
- Most integrations produce a variety of alerts that you may want to send to separate incident types, which may use different playbook/response processes. Create an incident classifier to route alerts from the integration to incident types in Cortex XSOAR. After you create a classifier, add the classifier to an integration. For more information, see [Create an incident classifier](#).

#### NOTE:

To hard code an incident type or select a classifier in an integration instance, you may need to select Fetches incidents in the integration instance settings.

Some content packs include classifiers, which have incident types already classified. For example, the Cortex XDR Incident Handler - Classifier classifies events such as FirstSSOAccess and RDPBruteForce as Cortex XDR Incident types in Cortex XSOAR.

### Mapping

Mapping enables you to map important information from incoming alerts into incident fields for use in playbooks and in layouts, so analysts can view the information when investigating an incident.

Some content packs include mappers, which have fields already mapped. For example, the XDR - Incoming Mapper includes fields such as Hostnames, LastMirroredInTime, and Occurred are already mapped. To create a mapper, see Create an incident mapper.

Mappers enable you to do the following:

- When building playbooks, incidents are much easier to use and allow you to take different actions based on those fields within a playbook.
- Most field types become searchable. For example, if you map the source username to a field, you can query that field with other incidents with the same source username. It is easier to correlate, deduplicate, query, and report.
- Easily add fields to layouts for display and review by the analyst.
- Perform indicator extraction based on the incident type and its fields. Extract specific indicators from specific fields.
- Mirror content in Cortex XSOAR with third-party integrations. This enables you to make changes to an incident in Cortex XSOAR and have that change be reflected in the case managed by the integration. For example, if you are using a case management system such as JIRA or Salesforce, you can close an incident in Cortex XSOAR and have that reflected automatically.

**NOTE:**

The integration must support pulling the integration schema for mirroring to work.

## Using JSON files

When creating a classifier or mapper you can use the following:

- Integration instance: The instance needs to be configured and enabled.
- Select a schema: When supported by the integration, this pulls all of the integration fields from the database.
- Upload a JSON file: If you can't pull samples or the samples do not retrieve sufficient data, upload a formatted JSON file.

The JSON file needs to be in an array of dictionaries, with each alert in its own dictionary. For example:

```
[
  {
    "type": "url allowed",
    "EventID": "5106",
    "urlCategory": "PHISH",
    "sourceIP": "10.8.8.181",
    "occurred": "2024-05-22T08:16:26Z",
    "sourceUser": "james.bond@xsoar.local",
    "url": "https://notthedomainyouarelookingfor.com/login.php",
    "userAgent": "Mozilla/5.0 (WindowsNT6.1;WOW64;rv:27.0) Gecko/20100101Firefox/27.0"
  },
  {
    "type": "url blocked",
    "EventID": "7893",
    "urlCategory": "MALWARE",
    "sourceIP": "10.8.8.127",
    "occurred": "2024-05-22T08:16:26Z",
    "sourceUser": "eve.moneypenny@xsoar.local",
    "url": "https://notthedomainyouarelookingfor.com/login.zip",
    "userAgent": "Mozilla/5.0 (WindowsNT6.1;WOW64;rv:27.0) Gecko/20100101Firefox/27.0"
  }
]
```

### 9.3.1 | Create an incident classifier

#### Abstract

Classify events using a classification key in an integration ingestion. Create incident classifier in Cortex XSOAR

When an integration fetches incidents, it populates the raw JSON object in the incident object. The raw JSON object contains all of the attributes for the event, such as the source of the event and when the event was created. When classifying the event, select an attribute that can determine the event type.

When creating a classifier, you can pull data from the following:

- An existing integration instance

**NOTE:**

Ensure that your instance is configured and enabled but you don't need to fetch incidents.

- Schema

When supported by the integration, this pulls all of the integration fields from the database. You select from these fields to classify the events.

- Upload a JSON file

Upload a formatted JSON file which includes the field you want to classify. If the instance has nothing to fetch or has insufficient data, you can upload a JSON file containing raw data.

1. Go to Settings & Info → Settings → Object Setup → Incidents → Classification & Mapping.

2. Do one of the following:

a. To create a new classifier, select New → Incident Classifier.

b. To edit an existing classifier open the classifier.

If the classifier is installed from a content pack, you need to duplicate and then open it.

3. Enter a name for the classifier so it can be easily identified.

4. Under Get data, select from where you want to pull the event data. You will classify the incident types based on this information.

- Pull from instance

- Select schema

- Upload JSON

5. Under Select Instance, select the integration instance from where you want to pull data.

In the Data fetched from [name of integration instance] section, you will see the raw alert data pulled in from the integration instance. In this example, after configuring the Sample Incident Generator instance, we have pulled in the following data:

The screenshot shows a dark-themed interface titled "Sample Incident - Classifier". At the top, it says "Data fetched from Sample Incident Generator\_instance\_2 (Sample Incident Gene...)" with a "1/5" page indicator and a "Search JSON" button. Below this, the JSON data is displayed in a tree view:

```

root: [] 6 items
  severity: 3
  type: Malware
  phase: 1. Triage
    labels: [] 4 items
      details: Detected Incident with malicious on files on system sample-laptop.
      name: Sample Incident - Malware
  
```

6. To route the alert information to an incident type, select the classification key,

a. In the Data fetched section, click the key you want to map. For example, type.

**TIP:**

Select a key that is common to across all the samples. If a key is selected that will change across all alerts such as SourceIP there could be many values.

In the Unmapped Values section, the selected key returns any unmapped classifier values. For example, type returns Malware, Unclassified, and Phishing.

3 Unmapped Values

Select the field to identify the type of the incident

type (i) x

Result: Malware

Drag classifier values to the incident type on the right.

Malware    Unclassified    Phishing

Direct unclassified events to: Select ▼

b. Drag and drop the unmapped classifier values onto the Incident Types section.

For example, you can see the Malware and Phishing values have been mapped to the relevant incident type.

0 Unmapped Values

Select the field to identify the type of the incident

type (i) x

Result: Malware

Drag classifier values to the incident type on the right.

Malware    Unclassified    Phishing

XSOAR Incident Types

Malware

Malware x

Malware Investigation and Response

mandat

Microsoft Sentinel Incident

my\_custom\_type

Network

PanoramaThreatCoverage

Phishing

Phishing x

c. In the Direct Unclassified events to field, select the incident type for unclassified events.

If you don't choose a default incident type, the classifier uses the default incident type, which is set to the Unclassified incident type. To view the default incident type, go to the Incidents page and add the Default column. You can set a different default incident type as required.

d. (Optional) If there are events that haven't been pulled from the samples you can manually add them to the Incident Types section, by clicking the edit button in the Incident Type field. For example, if you know that the source has a file blocked incident type, click the edit button on the relevant field and type file blocked.

7. Save the classifier.

8. (Optional) Create a mapper, if required.

9. Go to Settings & Info → Settings → Integrations → Instances.

a. Select the integration from which you want to apply the classifier.

b. In the integration settings, under Classifier, select the classifier you created and click Done.

For some instance integrations, you need to click Fetches incidents to add a classifier and mapper.

### 9.3.2 | Create an incident mapper

#### Abstract

Create a mapper and apply it to an integration in Cortex XSOAR.

You can create the following incident mappers:

- Incoming mapper: Maps all fields you pull from the integration to the incident fields.
- Outgoing mapper: Maps incident fields with fields in the integration to which you are pushing the data. This is useful for mirroring.

You can map your fields to incident types irrespective of the integration or classifier, which means that you can create mapping before defining an instance or ingesting incidents. By doing so, when you do define an instance and apply a mapper, the incidents that come in are already mapped.

When you create a mapper you can select the following incident types, which show the incident fields to map.

- Common Mapping: Defines how fields associated to all incident types are mapped.
- Specific mapping: Defines how fields associated with the specific incident type are mapped.

Specific mapping overrides any mapping done in Common Mapping. When an incident is ingested, common mapping and then specific mapping are applied.

#### TIP:

It is recommended to map all of the fields that are common to all incident types by selecting Common Mapping and then map additional fields that are specific to each incident type.

When mapping a list, we recommend you map to a multi-select field. Short text fields do not support lists. If you do need to map a list to a short text field, add a transformer in the relevant playbook task, to split the data back into a list.

You can also use Auto Map to automatically map fields, based on the same or similar names from the integration instance. For example, Severity can be mapped to Importance.

Some out-of-the-box fields are entirely controlled by Cortex XSOAR, and cannot be mapped, such as:

- Dbot Status
- Dbot Closed
- Dbot Total Time
- Close Notes
- Feed Based

#### NOTE:

Anything that you do not map will be discarded. If you want the data or you are not sure at this stage whether you want to map the data in the future, unmapped data can be placed into labels. Labels are unmapped and unsearchable data that is associated with the incident. Although you can use labels in playbooks, it is recommended that you map the required data, otherwise, all of the raw data goes into labels including both mapped and unmapped data.

To turn labels on, go to  Settings → Advanced and deselect Do not map JSON fields into labels for selected incident type. This means

How to create a mapper

1. Go to Settings & Info → Settings → Objects Setup → Incidents → Classification & Mapping.

2. Click New and select the mapper that you want to create.

- Incident Mapper (Incoming)
- Incident Mapper (Outgoing)

3. Enter a name for the mapper, so it can be easily identified.
4. Under Get data, select from where you want to pull the information.

- Pull from the instance: Select an existing integration instance.

When classifying or mapping data and using the integration instance to retrieve data, the instance must be configured and enabled. You don't need to fetch incidents.

- Select the schema: When supported by the integration, this will pull all of the fields for the integration from the database. This enables you to see all of the fields for each given event type that the integration supports. For example, the Palo Alto Networks Cortex XDR - Investigation and Response integration supports a schema.
- Upload JSON: Upload a formatted JSON file that includes the field you want to map.

If the instance has nothing to fetch or the integration instance has insufficient data, upload a JSON file containing raw data.

#### NOTE:

If creating an outgoing mapper you can only select a schema or upload a JSON file.

5. Under Select Instance, select the integration instance from where you want to pull data.

On the right-hand side, in the Data fetched from [name of integration instance] section, you will see the raw alert data pulled in from the integration instance. In this example, after configuring the XSOAR Engineering Training Instance (from the XSOAR Engineering Training content pack), we have pulled in the following data:

The screenshot shows a JSON viewer interface with the title "Data fetched from XSOAR Engineer Training\_instance\_1 (XSOAR Engineer Training)". At the top, there are navigation arrows for page 1/15 and a search bar labeled "Search JSON". Below the title, the JSON data is displayed under a "root" node, which contains 8 items. The data includes fields such as type (url allowed), eventID (2012), urlCategory (SPAM), sourceIP (10.8.8.138), occurred (2024-05-22T07:56:26Z), sourceUser (q@xsoar.local), url (https://xsoar.pan.dev/77/getnewbike), and userAgent (Mozilla/5.0(WindowsNT6.1;WOW64;rv:27.0)Gecko/20100101Firefox/27.0).

```
root: {} 8 items
type: url allowed
eventID: 2012
urlCategory: SPAM
sourceIP: 10.8.8.138
occurred: 2024-05-22T07:56:26Z
sourceUser: q@xsoar.local
url: https://xsoar.pan.dev/77/getnewbike
userAgent: Mozilla/5.0(WindowsNT6.1;WOW64;rv:27.0)Gecko/20100101Firefox/27.0
```

#### NOTE:

If creating an outgoing mapper, data from the integration schema appears on the left-hand side of the page.

6. Select the incident type you want to map.

By default the Incident type is set to Common mapping, which includes fields that are common to all of the incident types. This saves you time having to define these fields individually in each incident type.

#### NOTE:

When using common mapping it shows fields that are relevant for all incident types. If you created incident fields that are specific to incident types, these fields do not appear, and you need to select the relevant type.

7. (Outgoing mapper only) In the Incident samples section select the incident you want to map.

If you don't have any ingested incidents, select Playground.

8. Start mapping the fields.

- (Optional) Automatically map fields. Click Auto Map for Cortex XSOAR to map fields with common or similar names. For example, Cortex XSOAR can map Importance to Severity or sourceIP to Source IP.

You can Auto Map at any time. These settings do not override any manual mapping.

- Manually map fields.

1. Select the field you want to map and click Choose data path.

2. On the right-hand side click the relevant key.

In this example, you can see that we have mapped Event ID and Event Type.

The screenshot shows the Cortex XSOAR Mapper interface. On the left, there's a search bar with 'eve' and a dropdown 'Show: All'. Below it are two sections: 'Event ID' and 'Event Type'. Under 'Event ID', there's a 'Result: 2012' entry with 'eventType' highlighted in blue. Under 'Event Type', there's a 'Result: url allowed' entry with 'type' highlighted in blue. On the right, a sidebar titled 'root: [] 8 items' lists various event properties: type: url allowed, eventID: 2012, urCategory: SPAM, sourceIP: 10.8.8.138, occurred: 2024-05-22T07:56:26Z, sourceUser: @cortex.local, url: https://cortex.pan.dev/v7.7/getnewbike, userAgent: Mozilla/5.0(Windows NT 6.1; WOW64) AppleWebKit/537.36(Gecko/20100101) Firefox/27.0.

If creating an outgoing mapper, on the right-hand side, click the relevant incident field to map from.

#### **NOTE:**

Some fields are automatically mapped when you start defining the mapper if Cortex XSOAR recognizes it has something similar. Although it appears as Show:Unmapped to make sure, map the item.

9. Add any filters and transformers.

- a. Click the mapped field and then click the curly brackets.

- b. Add the filters and transformers, as required. For more information, see Transformer considerations, categories, and built-in transformers.

- c. Save the filters and transformers.

10. Repeat this process for the other incident types for which this mapping is relevant.

When selecting an incident type, you can copy the mapper that you created previously. This is useful if you are mapping to multiple incident types through your classifier, as you will need to perform mapping on each incident type or through common mapping.

11. Save the mapper.

12. Go to Settings & Info → Settings → Integrations → Instances.

- a. Select the integration to add the mapper.

- b. In the integration settings, under Mapper, select the mapper you created and click Done.

If you can't see the Mapper field, select Fetches Incidents.

#### **NOTE:**

It is recommended that you turn off Fetches Incidents, as soon as the integration fetches incidents until you have configured a playbook to run on the incident type.

After the integration instance starts fetching incidents go to the Incidents page to see how the classifier and mapper performed. The incident type should be populated with the correct type. You can also add relevant fields in the incident table to see if they are mapped correctly. You can also view the information including incident and labels in Context Data from (Side panels) when investigating an incident.

## 9.4 | Set up incident mirroring

### Abstract

Set up integrations such as ServiceNow v2 to mirror ServiceNow incidents to Cortex XSOAR.

When mirroring incidents, you can make changes in a third-party application, such as ServiceNow and Jira that will be reflected in Cortex XSOAR, or vice versa. You can also attach files from either of the systems, which will then be available in the other system.

Setting up your integration to mirror incidents from the third-party application in Cortex XSOAR includes the following:

- Configure mirroring for triggering incidents originating from your third-party application or from another fetching integration. For example, see the ServiceNow v2 integration.
- Configure incoming and outgoing mappers. For more information, see Classification and mapping.
- Configure account roles for API calls. For example, see the ServiceNow v2 integration.
- Run mirroring commands. For more information about mirroring commands, see the Mirroring Integration or your third-party integration, for example ServiceNow v2.

## 9.5 | Incident deduplication in Cortex XSOAR

### Abstract

Deduplicate incidents either manually or automatically in Cortex XSOAR. Mark as duplicate using pre-process rules or playbooks.

When ingesting incidents, you may ingest several incidents that are duplicated. Cortex XSOAR provides the following deduplication capabilities:

- Manual deduplication

During an investigation, on the Incidents page, an analyst can manually deduplicate incidents. For more information, see Incident management.

- Automatic deduplication

Option	Description
Pre-process rules	Set up pre-process rules to deduplicate incidents as soon as they are ingested into Cortex XSOAR.
Playbooks	There are several out-of-the-box playbooks you can run to identify and close duplicate incidents. Alternatively, you can use these playbooks as the basis for customized de-duplication playbooks. For example, instead of automatically closing the duplicate incidents, an analyst can review the duplicated incidents. The Dedup - Generic v4 playbook Identifies duplicate incidents using the machine learning model (used mainly for phishing). For more information, see Dedup - Generic v4.
Scripts	<p>Automate deduplication by creating a script or using one of the out-of-the-box scripts, such as:</p> <ul style="list-style-type: none"> <li>FindDuplicateEmailIncidents: Used to find duplicate emails for phishing incidents including malicious, spam, and legitimate emails, and whether to close them as duplicates. For more information, see FindDuplicateEmailIncidents</li> <li>DBotFindSimilarIncidents: Finds past similar incidents based on incident fields' similarity. Includes an option to display indicators similarity. For more information, see DBotFindSimilarIncidents.</li> <li>DBotFindSimilarIncidentsByIndicators: Finds similar incidents based on indicators' similarity. Indicators' contribution to the final score is based on their scarcity. For more information, see DBotFindSimilarIncidentsByIndicators.</li> </ul> <p><b>NOTE:</b></p> <p>The DBotFindSimilarIncidents and DBotFindSimilarIncidentsByIndicators are used in the Dedup - Generic v4 playbook.</p>

## 9.6 | Pre-process rules

### Abstract

Create pre-process rules to perform actions on incidents as soon as they are ingested.

Pre-process rules enable you to perform certain actions on incidents as soon as they are ingested (after classification and mapping) but before the incident is created in Cortex XSOAR. These rules enable you to drop, deduplicate, link, or close incoming incidents based on specific criteria. For example link the incoming incident to an existing incident, or under preconfigured conditions, drop the incoming incident altogether.

When creating pre-process rules you can test them on existing incidents to see how they perform.

Creating a pre-process rule consists of a three-part process using the preprocess wizard.

1. Select the incident field and value you want the rule to apply.
2. Select the action to perform on the incident, such as link and drop.
3. Add the criteria to compare existing incidents with the new incident, including the time range and oldest and newest incidents.

After you create a rule in the Pre-Process Rules tab, you can do the following:

- View, edit, copy, or delete the pre-process rule.
- Enable/disable the pre-process rule.

#### **NOTE:**

Rules are executed in the order they appear (from top to bottom). You can drag and drop rules as required. Only one rule is applied per incident.

#### Rule actions for pre-process rules

The following table describes the rule action for pre-process rules.

Option	Description
Link and close	Creates an entry in the Linked Incidents table of the existing incident to which you link, and closes the incoming incident. If an existing incident matching the defined criteria is not found, an incident is created for the incoming event.
Close	Closes the incoming incident. The incident will be created, but the associated playbook doesn't run.
Drop	Drops the incoming incident and no incident is created. Used for incidents that have low severity, no severity, or they have no value and don't need to be investigated.
Drop and update	Drops the incoming event, and updates the Dropped Duplicate Incidents table of the existing incident that you define. In addition, a War Room entry is created. If an existing incident matching the defined criteria is not found, an incident is created for the incoming event.
Link	Creates an entry in the Linked Incidents table of the existing incident to which you link.
Run a script	<p>Select a script to run on the incoming incident.</p> <p><b>NOTE:</b></p> <p>Pre-Process rules that use system-based scripts such as <code>GetIncidentsByQuery</code>, by default, are run according to the defined role (Limited User). For example, if the <code>GetIncidentsByQuery</code> script runs with the Limited User role, it also runs with the Limited User role in the Pre-Process rule. You can change the default by either detaching the script and updating the RunAs field such as DbotRole, or create a wrapper script with the required role set in the RunAs field. The wrapper script calls the system-based script. The system-based when called by the wrapper script runs with the role assigned to the wrapper script.</p> <p>Pre-processing scripts can access sensitive incident data. As best practice, we recommend assigning a Role for the pre-processing script to allow only trusted users to edit it.</p>

#### Create a pre-process rule

Pre-processing rules enable you to perform certain actions on incidents as they are ingested into Cortex XSOAR. You can, for example, link an incoming incident to an existing incident, or under certain conditions, drop the incoming incident altogether.

Before you begin, search for incidents that you want the pre-process rule to apply and click Investigate, so that those incidents are available for testing.

1. Select Settings & Info → Settings → Object Setup → Incidents → Pre-Process Rules → New Rule.
2. In the Rule Name field, type a name for the rule.

Give a meaningful name that helps you identify what the rule does. This will be useful when viewing the list of rules.

3. In step 1 Conditions for Incoming incident to apply the rule for incidents, do the following:

- a. Select a field and value.

For example, if you want to apply the rule to a phishing incident type:

Field	Filter	Value
Type	Equals (String)  <b>NOTE:</b> For more information about filters, see Filter considerations, categories, and built-in filters.	Phishing

b. Add an AND statement to your filter, if required.

For example, if you are running a phishing awareness campaign, add Email Subject and in the value field, type the relevant text.

c. If you want an OR statement, click the + sign.

For example, you may want the rule to apply to blocked or spam alerts.

**NOTE:**

If you want to remove an ADD or OR statement, click the - sign.

4. In step 2 Action, select the action to take if the incoming incident matches the rule.

5. If relevant, complete section 3.

This section enables you to link or update an incoming event and drop or update the incident depending on the selected criteria.

Section	Options						
Link to	<p>Relevant when selecting Link and close and Link</p> <ul style="list-style-type: none"> <li>Determine if you want to link to the oldest or newest incident.</li> <li>Select the time range</li> <li>Select if you want to search for closed incidents.</li> <li>Select the incident field and value you want to link. For example, if you want to link the Email Subject field of the existing incident to the new incident, do the following:</li> </ul> <table border="1"> <thead> <tr> <th>Field</th> <th>Filter</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Email Subject</td> <td>Is identical (Incoming incident)</td> <td>to incoming incident (this field is prepopulated)</td> </tr> </tbody> </table>	Field	Filter	Value	Email Subject	Is identical (Incoming incident)	to incoming incident (this field is prepopulated)
Field	Filter	Value					
Email Subject	Is identical (Incoming incident)	to incoming incident (this field is prepopulated)					
Update	<p>Relevant when selecting Drop and update</p> <p>Drops the incoming event and updates the incident defined:</p> <ul style="list-style-type: none"> <li>Determine if you want to link to the oldest or newest incident.</li> <li>Select the time range</li> <li>Select if you want to search for closed incidents.</li> <li>Select the incident field and value you want to drop and update.</li> </ul>						
Script	<p>Choose a script</p> <p>From the dropdown list, select the script to run on the incoming incident. Only scripts that were tagged preProcessing appear in the drop-down list.</p>						

6. (Optional) In a remote repository environment, you can view the relevant dependencies to ensure that all necessary dependencies are propagated or pushed to the remote repository.

7. (Optional) To check that the rules, click Test.

Testing is useful to check that you are receiving the desired results before putting a rule into production. We recommend you fetch data from an existing incident as a sample incident against which the rule can run. You can also manually enter JSON to use as a test sample or edit the JSON from an existing incident using the Edit button.

8. Save the rule.

### Pre-process rules examples

#### Drop incidents

When you run a phishing awareness campaign and send training emails to your employees, you want your employees to report the emails but you don't want to investigate. In this example, we create a condition for incoming incidents with the email subject You've Won the Best Employee Award, and drop those incidents without linking them to another incident.

The screenshot shows the configuration of a pre-process rule. It is divided into two main sections: 'Conditions for Incoming Incident' and 'Action'.

**1 Conditions for Incoming Incident:** This section is titled 'Determines if the rule is applied to an incoming incident. For the rule to apply to all incoming incidents, remove all conditions.' It contains one condition entry:

Email Subject	Equals (String)	You've Won the Best Employee Award	<span style="color: blue;">+</span> <span style="color: red;">-</span>
---------------	--------------------	------------------------------------	--

**2 Action:** This section is titled 'Action performed on the incoming incident. All sets of conditions must be met for the action to be performed.' It contains one action entry:

Drop	<span style="color: blue;">▼</span>
------	-------------------------------------

#### Drop and update incidents

In this example, you want a pre-process rule to do the following:

- Apply to incidents that are ingested from the Sample Incident Generator.
- Drop incoming events and update the incident, if the name of the existing incident is identical to the incoming incident.

You can add multiple conditionals to check for duplicates (not just the incident name) such as a Threat ID, incident ID, email, and host.

1. Create a pre-process rule.

\* Rule name  
Sample Incident Generator Drop and Update

## 1 Conditions for Incoming Incident

Determines if the rule is applied to an incoming incident. For the rule to apply to all incoming incidents, remove all conditions.

SourceBrand Equals (String) Sample Incident Generator + -

[+ Add filter](#)

## 2 Action

Action performed on the incoming incident. All sets of conditions must be met for the action to be performed.

Drop and update

## 3 Update

Drop the incoming event and update the incident defined in the following filter. If no matches are found, an incident is created.

Link to oldest incident Created within the last 30 Days  Search closed incidents

AND

Name of existing incident Is identical (Incoming incident) to incoming incident + -

[+ AND](#)

2. (Optional) Test the rule to ensure that it is working correctly.

Sample Incident Generator Drop and Update - Testing

Conditions for Incoming Incident  
sourceBrand equals Sample Incident Generator

Action  
Drop and update oldest matching incident

Compare to the following set of existing incidents:  
Created within the last 30 days AND name is identical to \${incident.name}

Incoming incident test sample  
Data is taken from incident #118608 Sample Incident - Phishing

```
root: {} 63 items
custom_status:
lastOpen: 2024-05-30T06:02:13.127961721Z
autime: 1716445476152000000
details: Hi, got this in my inbox today. From: somerandomemail@nodomain.net Sent: Sunday, January 24, 2016 19:43 To: Bob<bob@demisto.int>; Subject: Cloud Services Invoice Dear customer, Thank you for signing up for Acme Crea...
rawType: Phishing
dueDate: 2024-06-02T06:24:36.152870512Z
attachment: null
retained: false
```

Test

Test result  
Incident will be dropped. incident #112993 will be updated.

Done testing

3. (Optional) Go to an incident and check the Context Data (Side panels).

Review the droppedCount key (line 52).

```

CONTEXT DATA
Search in JSON context data...
Collapse | Expand
34  isPlayground: false
35  resolution_status: ""
36  sizeInBytes: 0
37  canvases: NULL
38  rawCategory: ""
39  sla: 0
40  dbotDirtyFields: NULL
41  created: "2024-05-20T08:56:27.881Z"
42  linkedIncidents: NULL
43  modified: "2024-06-04T06:20:37.22Z"
44  isDebug: false
45  closeNotes: ""
46  parent: ""
47  name: "Sample Incident - Phishing"
48  sourceBrand: "Sample Incident Generator"
49  closed: "0001-01-01T00:00:00Z"
50  account: ""
51  parentXDRIncident: ""
52  droppedCount: 5
53  rawJSON: ""
54  sourceInstance: "Sample Incident Generator_instance_2"
55  lastJobRunTime: "0001-01-01T00:00:00Z"
56  activated: "0001-01-01T00:00:00Z"
57  reminder: "0001-01-01T00:00:00Z"
58  status: 1
59  dbotMirroredSince: "0001-01-01T00:00:00Z"

```

Drop incidents and drop and update existing incidents

Watch the following video to see how to drop blocked or spam incidents and drop and update incidents.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

## 9.7 | Use post-processing scripts in an incident

### Abstract

You can set up a post-processing script to run after an incident has been remediated, but before the incident is closed in Cortex XSOAR.

Post-processing scripts perform actions on an incident after it is remediated but before it is closed by an analyst or automatically in a script or playbook. For example, after remediating an incident, an analyst may want to perform additional actions on the incident, such as closing a ticket in a ticketing system, sending an email, or preventing an incident from being closed without an assigned owner. You can create a post-processing script to cover these scenarios.

The following content packs include post-processing scripts:

- Common Scripts: Includes the `GenerateInvestigationSummaryReport` script, which generates a report when an investigation is closed.
- Case Management - Generic: Includes the `CloseLinkedIncidentsPostProcessing`, which closes any linked incidents when the incident is closed.

You can search for post-processing scripts on the Scripts page by using the Tags filter and typing post-processing.

You need to create a post-process script and then add the script to the incident type.

Example 5.

For an example of creating post-processing scripts that prevent an incident from being closed without an assigned user or the close notes not being filed out correctly, together with a Service Now example, see the following video:

Terjadi error.

---

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

#### Create a post-processing script

1. Select Scripts → New Script.
2. Type a name for the post-processing script and click Save.
3. In the Tags field, from the dropdown list select post-processing.
4. Add arguments as required.

Argument	Description
closed	The incident closed time.
status	The status of the incident
openDuration	The open incident duration between the created and closed dates.
closeNotes	The close notes of the incident
ClosingUserId	The username of the user who closed the incident, or DBot if the incident was closed by DBot (for example, through a playbook).
closeReason	The close reason for the incident.
N/a	Any other field values passed in at closure, whether through the incident close form, the CLI, or a playbook task.

5. Save the script.
6. Add the script to the incident type.

#### Example 6.

The following script example requires the user to verify all To Do tasks before closing an incident. Before you start, you need to configure and enable a Cortex XSOAR REST API instance. For more information, see Core REST API.

```
inc_id = demisto.incidents()[0].get('id')
tasks = list(demisto.executeCommand("core-api-get", {"uri": "/todo/{}".format(inc_id)})[0]['Contents']['response'])

if tasks:
    for task in tasks:
        if not task.get("completedBy"):
            return_error("Please complete all ToDo tasks before closing the incident")
            break
```

#### Example 7.

In this example, create a post-processing script for Service Now incidents using a SNOW instance, where there are required fields to resolve and close (such as Resolution Code and Resolution Notes).

This script works with the defaults from Service Now and resolves and closes the mirrored ticket in Service Now.

```

commonfields:
  id: c8eeeb6c-3622-4bcb-897a-d183625609fd
  version: 20
vcShouldKeepItemLegacyProdMachine: false
name: ServiceNowCloseIncidentTicket
script: !-
  # return the args and incident details to the war room, useful for seeing what you have available to you
  # args can be called with demisto.args().get('argname')

  # debugging
  # demisto.results(demisto.args())
  # demisto.results(demisto.incident())

  # get the close notes and reason from the XSOAR Incident
  close_reason = demisto.args().get('closeReason')
  close_notes = demisto.args().get('closeNotes', 'No close notes provided')
  servicenow_sysid = demisto.incident().get("dbotMirrorId", False)

  # map XSOAR close reasons to Service Now close codes
  close_code_map = {
    "False Positive": "Not Solved (Not Reproducible)",
    "Resolved": "Solved (Permanently)",
    "Other": "Solved (Work Around)",
    "Duplicate": "Solved (Work Around)"
  }

  close_code = close_code_map.get(close_reason, "Solved (Work Around)")

  # handle if there is no service now sys_id, resolve and close snow ticket
  if servicenow_sysid:
    demisto.results(demisto.executeCommand("servicenow-update-ticket", {"id":servicenow_sysid,"close_code":close_code,"state":6,"close_notes":close_notes}))
    demisto.results(demisto.executeCommand("servicenow-update-ticket", {"id":servicenow_sysid,"state":7}))

  else:
    demisto.results("No ServiceNow sys_id found, doing nothing...")

type: python
tags:
- post-processing
- training
comment: Post processing script to resolve and close Service Now tickets if the XSOAR
  Incident is closed.
enabled: true
scripttarget: 0
subtype: python3
timeout: 80ns
pswd: ""
runonce: false
dockerrimage: demisto/python:1.3-alpine
runas: Administrator

```

#### NOTE:

If there is an additional custom argument defined for a post-processing script, arguments such as `closeNotes`, `closeReason`, `closed`, and `openDuration`, are not available in the `demisto.args()` dictionary. In this case, there are two options:

1. Remove the additional custom argument from Script settings and instead add it as a field on the Close Form for the incident type. This results in the additional argument being passed to the post-processing script.
2. Manually add the default system arguments such as `closeNotes`, `closeReason`, `closed`, and `openDuration` to the Script settings, in addition to the custom argument. If not added, the code example above `close_notes = demisto.args().get('closeNotes', 'No close notes provided')` always returns "No close notes provided".

7. Add the post-processing script to the incident type.

- a. Go to Settings & Info → Settings → Object Setup → Incidents → Types.
- b. Click the incident type you want to add the post-processing script.
- c. In the Post process using field, from the drop-down, select the script.
- d. Save the incident type.

After you add a post-processing script to the incident type, the incident type will use the post-processing script.

#### NOTE:

If a post-processing script returns an error, the incident does not close.

## 9.8 | Customize incident close reasons

### Abstract

Customize close reasons for incidents by adding a server configuration in Cortex XSOAR.

The default incident close reason values are:

- False Positive
- Resolved
- Duplicate
- Other

To customize the incident close reason, you need to add a new server configuration.

1. Select Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.

2. Add the following key and value:

Key	Value
<code>incident.closereasons</code>	A comma-separated list. For example, <code>False Positive,Resolved,Duplicate,Low Priority,Invalid,Other</code> .

## 9.9 | Configure inline value fields

### Abstract

Remove the checkmark when an analyst edits specific fields in a layout.

By default, when editing the following inline values in an incident, the changes are not saved until you confirm your changes (clicking the check mark icon in the value field):

- Dropdown values, such as Owner and Severity.
- Text values, such as Asset ID. (You can only edit when you click the pencil in the value field).

These icons provide an additional level of security before you make changes to the fields in incidents, indicators, and threat intel reports. If you want to allow users to change to inline fields without clicking the check mark, you need to add a server configuration.

1. Select Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.

2. Add the following server configuration.

Key	Value
<code>inline.edit.on.blur</code>	Set the server configuration to <code>true</code> , which enables you to make changes to inline fields without clicking the check mark. The changes are automatically saved when clicking anywhere on the page or when navigating to another page. For text values you can also click anywhere in the value field to edit.

## 9.10 | Export an incident to CSV using the UTF8-BOM format

### Abstract

Export an incident using Cyrillic characters. Export an incident to CSV using UTF8-BOM format. Server configuration.

By default, when exporting an incident to a CSV format, Cortex XSOAR generates the report in UTF8 format. If you want to export an incident that contains Cyrillic characters, such as Russian, Greek, etc., you need to change the format to UTF8-BOM. This also changes exporting an indicator to the UTF8-BOM format.

### NOTE:

When changing the format to UFT8-BOM you also change the format for indicators.

1. Select Settings & Info → Settings → System → Server Settings.
2. In the Server Configuration section, click Add Server Configuration.
3. Add the following key and value.

Key	Value
Export.utf8bom	true

4. Save the configuration

## 10 | Playbooks

### Abstract

Playbooks are a series of tasks, conditions, automation, commands, and loops that run in a predefined flow, which are at the heart of the Cortex XSOAR system.

Automate complex workflows by using playbooks to streamline repetitive tasks. Create or customize playbooks, define inputs and outputs, integrate custom scripts, and rigorously test with the built-in debugger for flawless execution.

### 10.1 | What is a playbook?

#### Abstract

Cortex XSOAR playbooks enable you to structure and automate many of your security processes. Parse incident information, interact with users, and remediate.

Playbooks are a series of tasks, automations, conditions, commands, and loops that run in a predefined flow to save time and improve the efficiency and results of the investigation and response process. They are at the heart of the Cortex XSOAR system, because they enable you to automate many security processes, including handling investigations and managing tickets. For example, a playbook task can parse the information in an incident, whether it is an email or a PDF attachment.

Playbooks have different task types for each action you want to take. For example:

- Use manual tasks when an analyst needs to confirm information or escalate an incident.
- Use conditional tasks to validate conditions based on values or parameters and take appropriate direction in the playbook workflow.
- Use communication tasks to interact with users in your organization.
- Use automation tasks to automatically remediate an incident by interacting with a third-party integration, open tickets in a ticketing system such as Jira, or detonate a file using a sandbox.

You can also structure and automate security responses that were previously handled manually.

You define the logical flow of your playbook when you design your use case. After developing and testing the playbook, it then runs during investigation and response.

#### NOTE:

Cortex XSOAR currently does not support the IoT Security Third-party Integrations Add-on . For more information, see the IoT Security documentation.

### 10.2 | Playbook development checklist

#### Abstract

Follow the playbook development flow to create playbooks that structure and automate many of your security processes.

The playbook development checklist follows the logical flow for developing a playbook.



We recommend that you review the following steps to successfully implement your playbook.

Step	Details	See More
Step 1. Plan your playbook	During the initial planning stage when designing your use case, start defining the playbook flow. Consider the process you want to automate and the steps and the decisions during the process. These steps and decisions become the playbook tasks.	See topic
Step 2. Develop your playbook	Consider whether to customize an existing playbook or create a new playbook from scratch. Create playbook tasks, inputs, and outputs. Maintain playbook versioning to keep track of playbook development history.	See topic
Step 3. Customize your playbook	Fine tune your playbook for your needs, including extracting indicators, extending context, and adding incident fields to the system.	See topic
Step 4. Debug your playbook	Debug errors in your playbook. Use playbook metadata to troubleshoot playbook performance.	See topic

## 10.3 | Plan your playbook

### Abstract

Considerations when planning your playbook.

When defining the work flow of your playbook, consider the following:

- What actions do you need to take?
- What conditions do you need along the way? Are these conditions manual or automatic?
- Do you need to include looping?
- Are there any time-sensitive aspects to the playbook?
- When is the incident considered remediated?

### Example 8. Review the Phishing use case

Review the following workflow for a phishing use case. Also, review the playbooks in the Phishing content pack to see how they work.

- Detection
- Identification
- Analysis
- Remediation

Each of these high-level processes can contain a number of sub-processes that require step-by-step actions, all of which can be automated with either customized or new playbooks.

### Example 9. Review the Default Playbook

The Default Playbook provides generic capabilities for automated incident enrichment and severity calculations that you can adjust for your needs. Watch this video for more details.

Terjadi error.

---

Cobalah menonton video ini di  
[www.youtube.com](http://www.youtube.com), atau aktifkan  
 JavaScript jika dinonaktifkan di browser  
 Anda.

## 10.4 | Develop your playbook

### Abstract

Create a new playbook or customize an existing one based on your organization's needs.

When developing your playbook, you can either customize an existing out-of-the-box playbook from a content pack or create a new playbook from scratch.

Developing a new playbook from scratch enables a tailored solution for your use case, whereas customizing an out-of-the-box playbook can save time, reduce complexity, and be a more efficient way to meet your organization's specific security and incident response needs.

Follow these steps to develop a playbook.

### Task 1. Customize or create a playbook

You can configure an existing playbook or create a new playbook.

#### Customize an out-of-the-box playbook

Search for a playbook that is included out-of-the-box with Cortex XSOAR or after downloading from Marketplace.

In the Cortex XSOAR Playbooks page, use free text in the search box to search for playbooks. You can search using part or all of the playbooks' names or description. You can also search for an exact match of the playbook name by putting quotation marks around the search text. For example, searching for "**Block Account - Generic**" returns the playbook with that name.

You can also search for more than one exact match by including the logical operator "or" in-between your search texts in quotation marks. For example, searching for "**Block Account - Generic**" or "**NGFW Scan**" returns the two playbooks with those names. Wildcards are not supported in free text search.

#### TIP:

Browse Marketplace to check for out-of-the-box playbooks that you can customize for your process. For an extensive list of available out-of-the-box playbooks, see Generic Playbooks.

#### Attach and detach playbooks

When installing a playbook from a content pack, by default, the playbook is attached, which means that it is not editable (apart from some input values).

To edit the playbook, you need to detach or make a duplicate. While it is detached, the playbook is not updated by the content pack. This may be useful when you want to update the playbook without breaking customization. If you want to update the playbook type through content pack updates, you need to reattach the playbook, but any changes are overridden by the content pack on upgrade. If you open an attached playbook in a tab, it can be detached from within the editor page.

If you want to keep the changes, duplicate the playbook before reattaching it.

#### Create a playbook

1. Go to Playbooks and click + New Playbook.

2. Enter a name for the playbook and click Save.

A blank playbook opens with the Playbook Triggered task that holds the playbook inputs and outputs.

#### NOTE:

To open multiple playbooks at the same time, edit the first playbook and then click New next to the playbook name to create a new tab. You can either create a new playbook, or add an existing one.

### Task 2. Configure playbook settings

Configure playbook settings as relevant, including:

- Name and description
- Tagging
- Access
- Whether to associate the playbook with an incident type. This needs to be set under the Settings & Info → Settings → Object Setup → Incidents → Types tab.
- Whether to run the playbook in Quiet Mode

For more information, see [Configure the general playbook settings](#).

### Task 3. Set playbook inputs and outputs

Depending on the task type that you select, and the script that you are running, your playbook task may have inputs and outputs.

Inputs are data pieces that are present in the playbook or task. The inputs are often manipulated or enriched and they produce outputs. Outputs are objects whose entries will serve the tasks throughout the playbook, and they can be derived from the result of a task or command.

At the beginning of any playbook, click the Playbook Triggered task and enter the playbook inputs and outputs, grouping them as relevant.

For more information, see [Playbook inputs and outputs](#).

#### Task 4. Add tasks

Playbook tasks are the building blocks of playbooks. Tasks enable you to run scripts and sub-playbooks, communicate with end users, set conditions, and store relevant data. Tasks can be reused across playbooks and you can copy, cut, paste, and delete tasks within or between playbooks using keyboard shortcuts. To see a list of keyboard shortcuts, see [Keyboard shortcuts](#).

You can also open a sub-playbook task and click Open sub-playbook to open the sub-playbook in a new tab.

**NOTE:**

To open multiple playbooks at the same time, edit the first playbook and then click the New icon next to the playbook name to create a new tab. You can either create a new playbook, or add an existing one.

Once you add tasks to your playbook, connect the tasks in their logical order by dragging and dropping a wire from one task to another.

Task Type	Description
Section	<p>Use a section header task to group related tasks to organize and manage the flow of your playbook.</p> <p>Section headers can also be used for time tracking between phases in a playbook. This data can be used to display in dashboards and report time trends.</p> <p>For example, in a phishing playbook you would have a section for the investigative phase of the playbook such as indicator enrichment, and a section for communication tasks with the user who reported the phishing.</p> <p>For more information, see <a href="#">Create a section header</a>.</p>
Standard	<p>Standard tasks can be manual tasks such as manual verification to prompt an analyst to verify the severity or classification of an incident before proceeding with automated actions. They can also be automated tasks such as parsing a file or enriching indicators.</p> <p>Automated tasks are based on scripts that exist in the system. These scripts can be created by you or come out-of-the-box as part of a content pack. For example, the <code>!ad-get-user</code> command retrieves detailed information about a user account using the Active Directory Query V2 integration.</p> <p>You can also automatically remediate an incident by interacting with a third-party integration, open tickets in a ticketing system such as Jira, or detonate a file using a sandbox.</p> <p>For more information, see <a href="#">Create a standard task</a>.</p>
Conditional	<p>Use conditional tasks to validate conditions based on values or parameters and take appropriate direction in the playbook workflow, like a decision tree in a flow chart.</p> <p>For example, a conditional task may ask whether indicators are found. If yes, you can have a task to enrich them, and if not you can proceed to determine that the incident is not malicious. Alternatively, you can use conditional tasks to check if a certain integration is available and enabled in your system. If yes, you can use that integration to perform an action, and if not, you can continue on a different branch in the decision tree.</p> <p>Conditional tasks can also be used to communicate with users through a single question survey, the answer to which determines how a playbook will proceed.</p> <p>For more information, see <a href="#">Create a conditional task</a>.</p>
Communication	<p>Use a communication task to interact with users through a survey, for example to collect responses or escalate an incident.</p> <p>All responses are collected and recorded in the incident context data, from a single user or multiple users. You can use the survey questions and answers as input for subsequent playbook tasks.</p> <p>You can collect responses in custom fields, for example, a grid field.</p> <p>For more information, see <a href="#">Create a communication task</a>.</p>

## Task 6. Add custom playbook features

You can customize your playbook to do the following.

Custom Action	Description
Customize the SOC name	Customize the name of the SOC that appears in the survey header.
Add a sub-playbook	<p>Playbooks can be divided into two categories, depending on their use.</p> <ul style="list-style-type: none"> <li>Parent playbooks are playbooks that run as the main playbook of an incident. For example, Phishing - Generic v3.</li> <li>Sub-playbooks are playbooks that are nested under other playbooks. They appear as tasks in the parent playbook flow and are indicated by the sub-playbook icon. A sub-playbook can also be a parent playbook in a different use case. For example, IP Enrichment - Generic v2.</li> </ul>
Field mapping	<p>You can map output from a playbook task directly to an incident field. This means that the value for an output key populates the specified field per incident. This is a good alternative to using a task with a set incident command.</p> <p>You can map when you select a script in a Standard or Conditional task. For more information, see Create a standard task.</p>
Filter and transform data	<p>Filters extract relevant data to help focus on relevant information and discard irrelevant or unnecessary data.</p> <p>Transformers take one value and transform or render it to another value or format.</p>
Use scripts	<p>Perform specific automated actions using commands that are also used in playbook tasks and in the War Room.</p> <p>Configure script error handling.</p>
Extract indicators	Extract indicators from incident fields and enrich them using commands and scripts defined for the indicator type.
Extend context	Save additional data from the raw response of commands that return data.
Set and update incident fields	Use the setIncident script in a playbook task to set and update incident fields.
Use playbook polling	Configure a playbook to stop and wait for a process to complete on a third-party product, and continue when it is done.

## Task 7. Test and debug the playbook

The debugger provides a test environment where you can make changes to data and playbook logic and view the results in real-time to test and troubleshoot playbooks. You can see exactly what is written to the context at each step and which indicators are extracted.

For more information, see [Debug your playbook](#).

## Task 8. Manage playbook content

Manage playbook content by either using a remote repository, or by saving versions of your playbook in Cortex XSOAR to maintain version history. For more details, see [Manage playbook content](#).

### 10.4.1 | Playbook tasks

#### Abstract

Create playbook tasks and link them to form the playbook flow.

Tasks are the building blocks of playbooks. Cortex XSOAR supports different task types for different actions to be taken in a playbook. Each task type requires different information and provides different capabilities. Choose your task type based on what you want to accomplish in the task. For example, for enrichment, you might want to run an enrichment sub-playbook or a command that returns additional information for an indicator.

When developing a playbook, you create the relevant playbook tasks and link them to form the playbook flow. There are different task types according to the actions you want to take, and each task can receive and generate data in the form of inputs and outputs.

Task Type	Description
Section	<p>Use a section header task to group related tasks to organize and manage the flow of your playbook.</p> <p>Section headers can also be used for time tracking between phases in a playbook. This data can be used to display in dashboards and report time trends.</p> <p>For example, in a phishing playbook you would have a section for the investigative phase of the playbook such as indicator enrichment, and a section for communication tasks with the user who reported the phishing.</p> <p>For more information, Create a section header.</p>
Standard	<p>Standard tasks can be manual tasks such as manual verification to prompt an analyst to verify the severity or classification of an incident before proceeding with automated actions. They can also be automated tasks such as parsing a file or enriching indicators.</p> <p>Automated tasks are based on scripts that exist in the system. These scripts can be created by you or come out-of-the-box as part of a content pack. For example, the <code>!ad-get-user</code> command retrieves detailed information about a user account using the Active Directory Query V2 integration.</p> <p>You can also automatically remediate an incident by interacting with a third-party integration, open tickets in a ticketing system such as Jira, or detonate a file using a sandbox.</p> <p>For more information, see Create a standard task.</p>
Conditional	<p>Use conditional tasks to validate conditions based on values or parameters and take appropriate direction in the playbook workflow, like a decision tree in a flow chart.</p> <p>For example, a conditional task may ask whether indicators are found. If yes, you can have a task to enrich them, and if not you can proceed to determine that the incident is not malicious. Alternatively, you can use conditional tasks to check if a certain integration is available and enabled in your system. If yes, you can use that integration to perform an action, and if not, you can continue on a different branch in the decision tree.</p> <p>Conditional tasks can also be used to communicate with users through a single question survey, the answer to which determines how a playbook will proceed.</p> <p>For more information, see Create a conditional task.</p>
Communication	<p>Use a communication task to interact with users through a survey, for example to collect responses or escalate an incident.</p> <p>All responses are collected and recorded in the incident context data, from a single user or multiple users. You can use the survey questions and answers as input for subsequent playbook tasks.</p> <p>You can collect responses in custom fields, for example, a grid field.</p> <p>For more information, see Create a communication task.</p>

#### 10.4.1.1 | Playbook inputs and outputs

##### Abstract

Cortex XSOAR playbooks and tasks have inputs (data from incident or integration) and outputs that can then be used as input in other tasks.

Playbooks and playbook tasks have inputs, which are pieces of information supplied to the system to carry out automated workflows.

An input may come from an incident, such as the role to assign an incident to, or an input can be provided by an integration, for example the Active Directory integration can be used in a task to extract a user's credentials.

You see the playbook inputs by clicking the top task Playbook Triggered in the playbook.

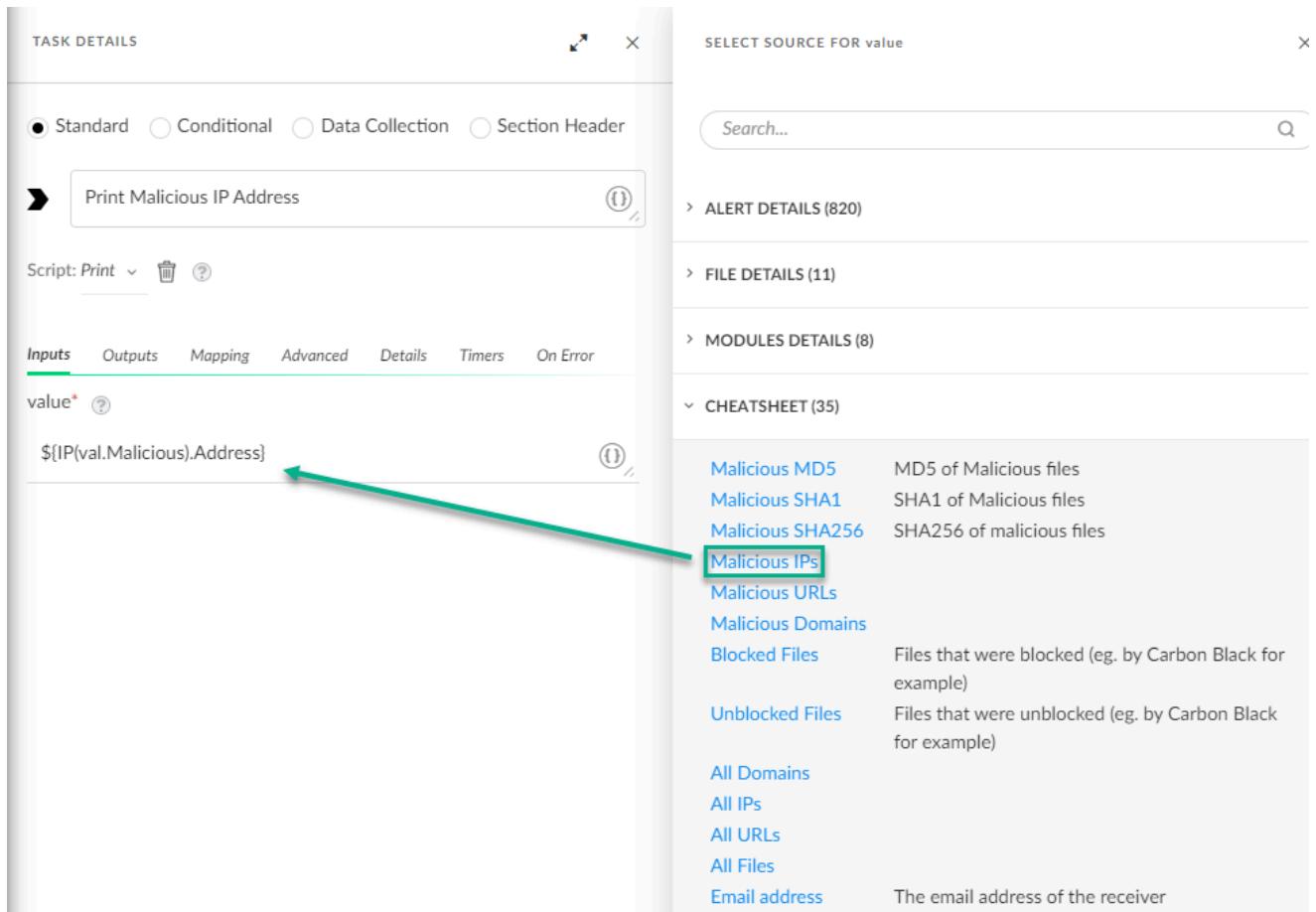
Use the task cheat sheet to use context keys in playbook inputs and outputs

When you create your playbook task inputs, the task cheat sheet enables quick access to system and custom fields to populate playbook task inputs and outputs.

1. Click .

The cheat sheet opens displaying incident fields.

2. Select an incident field, it populates the task input with the corresponding context key.



#### Example 10.

The following example uses incident context data as the playbook input from the Access Investigation - Generic playbook.

Click the top task Playbook Triggered. The playbook is triggered based on incident context data.

#### Inputs

The first two inputs are `SrcIP`, retrieved from the `incident.src` key, and `DstIP`, retrieved from the `incident.dest` key.

**PLAYBOOK INPUTS AND OUTPUTS**

From context data  From indicators [?](#)

**Inputs** **Outputs**

[Expand All](#) | [Collapse All](#)

**General (Inputs group)**  
Generic group for inputs

Name	SrcIP
Value	Get <a href="#">incident.src</a> Where No filters applied Transformers No transformers applied 
Description	The source IP address from which the incident originated.
<input type="checkbox"/> Mandatory	
Name	DstIP
Value	Get <a href="#">incident.dest</a> Where No filters applied Transformers No transformers applied 
Description	The target IP address that was accessed.
<input type="checkbox"/> Mandatory	

## Outputs

The Access Investigation - Generic playbook creates an output object that can be used in subsequent playbook tasks.

For example, the Access Investigation - Generic playbook `Endpoint.IP` output creates a list of endpoint IP addresses which can later be enriched by an IP enrichment task, and the `Endpoint.MAC` output creates a list of endpoint MAC addresses which can be used to get information about the hosts that were affected by the incidents.

Outputs can also be data that was extracted or derived from the inputs. For example, the Access Investigation - Generic playbook contains the Account Enrichment - Generic v2.1 sub-task, which uses the account username (and optionally domain) as input to Active Directory to retrieve user information as output, such as the user's email address, manager, and any groups to which they belong.

An output can then serve as input for a subsequent task. For example, in the Account Enrichment - Generic v2.1 sub-task, the Get account info from Active Directory task output `Account.Username` is used as an input for the Active Directory - Get User Manager Details task to retrieve manager details for that user.

## Group playbook inputs and outputs

Playbook input and output fields are collected into groups. This organizes the inputs and outputs, providing clarity and context to understand which inputs are relevant to which playbook flow.

For example, the following playbook inputs are grouped under Mailbox selection.

The screenshot shows the 'PLAYBOOK INPUTS AND OUTPUTS' configuration interface. It displays three input fields under the 'Mailbox selection' group:

- Listener Mailbox**: Value is set to \${File(val.Malicious).MD5}.
- SearchAndDelete**: Value is set to False.
- SearchAndDelete2**: Value is currently empty.

Below these fields are two expandable sections:

- Sub-playbooks activation**: Describes selecting which sub-playbooks to activate.
- Analyst Assignment**: Describes selecting the logic for analyst assignment.

At the bottom of the interface are buttons for '+ Add Input', '+ Add Input Group', 'Cancel', and 'Save'.

## Playbook group permissions

Users with permission to edit playbooks can add, edit, and delete groups and input and output fields. Users without this permission can only view groups, inputs, and outputs.

### Work with playbook groups

You can do the following with groups:

- Add or delete a group. Deleting a group deletes all the fields defined in the group.
- Change the name and/or description of the group.
- Change the order groups appear by dragging.
- Collapse and expand a group.

### How to add a new group

1. Click + Add Input Group or + Add Output Group.
2. Enter a group name and description and click the check mark.
3. Add fields to the group.

#### NOTE:

If you do not add any fields, the group will be deleted when you click Save.

### Manage input or output fields within a group

You can do the following with input or output fields within a group:

- Add, edit, or delete fields within a group. Input or output fields are always part of a group.
- Move fields between groups by dragging.
- Change field order within a group by dragging.

### How to add an input or output field in a group

## Inputs

1. Within a group, click + Add Input at the bottom of the list of input fields. You may need to scroll down to see it.
2. Enter the input field Name (required), Value, and Description.
3. When you are done adding fields, click Save.

## Outputs

1. Within a group, click + Add Output or + Add Manually at the bottom of the list of output fields. You may need to scroll down to see these options.
  - If you click + Add Output, select from the outputs from previous tasks.
  - If you click + Add Manually, enter the context path and description for the output.
2. When you are done adding fields, click Save.

### 10.4.1.2 | Create a section header

#### Abstract

Section header tasks are used to manage the flow of your playbook and help you organize your tasks efficiently.

Section headers are used to manage the flow of your playbook and help you organize your tasks efficiently. You create a section header to group a number of related tasks.

Section headers can also be used for time tracking between phases in a playbook. When you start time tracking, apply the Start action for the section header. Because you are using this to time track a particular phase of an investigation, add a stop timer section header when the phase completes. The time tracking data can be used to display in dashboards and report time trends.

1. In a playbook, click + to create a task.
2. Select the Section Header option.
3. Enter a meaningful name in the Task Name field for the section header.
4. Configure the relevant fields.

Tab	Fields In The Tab
Details	Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.
Timers	For a time tracking header, select the action to take when the timer is triggered (start, stop, or pause). <ul style="list-style-type: none"> <li>• Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.</li> <li>• Add Trigger: You can add other trigger timer fields from the drop down.</li> </ul>

5. Click Save.

### 10.4.1.3 | Create a standard task

#### Abstract

Define a standard playbook task in Cortex XSOAR.

Standard tasks can be manual tasks such as manual verification to prompt an analyst to verify the severity or classification of an incident before proceeding with automated actions. They can also be automated tasks such as parsing a file or enriching indicators.

1. In a playbook, click + to create a task.
2. Select the Standard option.
3. Enter a meaningful name in the Task Name field for the task that corresponds to the data you are collecting.
4. Select the options you want to configure for the Standard task.

Standard tasks include the following field and tabs.

Field / Tab	Settings
Choose script field	From a drop down list, select a script for the playbook to run. In the following tabs you can set:

Field / Tab	Settings
	<ul style="list-style-type: none"> <li>Inputs: Each script has its own set of input arguments (or none). You can set each argument to a specific value (by typing directly on the line under the argument name) or you can click the curly brackets to define a source field to populate the argument.</li> <li>Outputs: Each script has its own set of output arguments (or none).</li> <li>Mapping: <ul style="list-style-type: none"> <li>Map the output from a playbook task directly to an incident field.</li> </ul> <p>The value for an output key populates the specified field per incident. This is a good alternative to using a task with the <code>setIncident</code> command.</p> <p><b>NOTE:</b></p> <p>The output value is dynamic and is derived from the context at the time that the task is processed. As a result, parallel tasks that are based on the same output may return inconsistent results.</p> <ol style="list-style-type: none"> <li>In the Mapping tab, click Add custom output mapping.</li> <li>Under Outputs, select the output parameter whose output you want to map. Click the curly brackets to see a list of the output parameters available from the script.</li> <li>Under Field to fill, select the field that you want to populate with the output.</li> <li>Click Save.</li> </ol> <ul style="list-style-type: none"> <li>Advanced: Includes the following fields. <ul style="list-style-type: none"> <li>Using: Choose which integration instance will execute the command, or leave empty to use all integration instances.</li> <li>Extend context: Append the extracted results of the action to the context. For example, <code>"newContextKey1=path1::newContextKey2=path2"</code> returns "<code>[\path1:'aaa',path2: 'bbb', newContextKey1: 'aaa',newContextKey2:'bbb']</code>"</li> <li>Ignore outputs: If set to true, will not store outputs into the context (besides the extended outputs).</li> <li>Execution timeout (seconds): Sets the command execution timeout in seconds.</li> <li>Indicator Extraction mode: Choose when to extract indicators: <ul style="list-style-type: none"> <li>None: Do not perform indicator extraction</li> <li>Inline: Before other playbook tasks</li> <li>Out of band: While other tasks are running</li> </ul> </li> <li>Mark results as note</li> <li>Mark results as evidence</li> <li>Run without a worker</li> <li>Skip this branch if this script/playbook is unavailable</li> <li>Quiet Mode: When in quiet mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn quiet mode on or off at the task or playbook level.</li> </ul> </li> <li>Details: Includes the following fields. <ul style="list-style-type: none"> <li>Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.</li> <li>Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.</li> </ul> </li> <li>Timers: Includes the following fields. <ul style="list-style-type: none"> <li>Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.</li> <li>Add Trigger: You can add other trigger timer fields from the drop down.</li> </ul> </li> <li>On Error: Includes the following fields.</li> </ul> </li> </ul>

Field / Tab	Settings
	<ul style="list-style-type: none"> <li>◦ Number of retries: How many times the task should retry running if there is an error. Default is 0.</li> <li>◦ Retry interval (seconds): How long to wait between retries. Default is 30 seconds.</li> <li>◦ Error handling: How the task should behave if there is an error. Options are: <ul style="list-style-type: none"> <li>▪ Stop</li> <li>▪ Continue</li> <li>▪ Continue on error path(s)</li> </ul> <p>This option configures the task to handle potential errors that may occur when executing the current task's script.</p> </li> </ul>
Manual task settings tab	<ul style="list-style-type: none"> <li>• Default assignee: Assign an owner to this task.</li> <li>• Only the assignee can complete the task: Stop the playbook from proceeding until the task assignee completes the task. By default, in addition to the task assignee, the default administrator can also complete the blocked task. You can also block tasks until a user with an external email address completes the task.</li> <li>• Task SLA: Set the SLA in granularity of weeks, days, hours, and minutes.</li> <li>• Set task Reminder at: Set a reminder for the task in granularity of weeks, days, hours, and minutes.</li> </ul>
Advanced tab	<p>Quiet Mode: Determines whether this task uses the playbook default setting for quiet mode. When in quiet mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn quiet mode on or off at the task or playbook level.</p>
Details tab	<ul style="list-style-type: none"> <li>• Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.</li> <li>• Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.</li> </ul>
Timers tab	<ul style="list-style-type: none"> <li>• Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.</li> <li>• Add Trigger: You can add other trigger timer fields from the drop down.</li> </ul>

#### 10.4.1.4 | Create a conditional task

##### Abstract

Create a conditional task in a playbook.

Conditional tasks are used for determining different paths for your playbook. For example, in a playbook for handling phishing emails, a conditional task can be used to check if an email contains suspicious attachments. If the attachment is identified as malicious, the playbook can automatically quarantine the email; otherwise, it can proceed to manual review by a security analyst.

##### Conditional task types

You can create different types of conditional tasks.

- Built-in: Creates a logical statement using an entity from within the playbook. For example, in an access investigation playbook, you can determine that if the Asset ID of the person whose account was being accessed exists in a VIP list, set the incident severity to High. Otherwise, proceed as normal.
- Manual: Creates a conditional task that must be manually resolved. For example, a security analyst is prompted to review and validate a suspicious file. The playbook task might involve instructions for the analyst to analyze the file, determine if it is malicious, and provide feedback or take specific actions based on their assessment.
- Ask: Creates a single-question survey communication task, the answer to which determines how a playbook proceeds. For more details about ask tasks, see Create a communication task.
- Choose script: Creates a conditional task based on the result of a script. For example, check if an IP address is internal or external using the IsIPInRanges script. When using a script, the inputs and outputs are generated by the automation script.

#### How to create a conditional task

1. In a playbook, click + Create Task.
2. Select the Conditional option.
3. In the Task Name field, type a meaningful name for the task that corresponds to the data you are collecting.
4. Select the relevant conditional task option. Some field configurations are required, and some are optional.

#### Built-in

- Condition: Define one or more logical conditions for the task.
- Details: Includes the following fields.
  - Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.
  - Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.
- Timers: Includes the following fields.
  - Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.
  - Add Trigger: You can add other trigger timer fields from the drop down.
- Advanced: Determines whether this task uses the playbook default setting for Quiet Mode. When in Quiet Mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn Quiet Mode on or off at the task or playbook level.
- On Error: Includes the following fields.
  - Number of retries: How many times the task should retry running if there is an error. Default is 0.
  - Retry interval (seconds): How long to wait between retries. Default is 30 seconds.

#### Manual

- Manual task settings: Includes the following fields.
  - Default assignee: Assign an owner to this task.
  - Only the assignee can complete the task: Stop the playbook from proceeding until the task assignee completes the task. By default, in addition to the task assignee, the default administrator can also complete the blocked task. You can also block tasks until a user with an external email address completes the task.
  - Task SLA: Set the SLA in granularity of weeks, days, hours, and minutes.
  - Set task Reminder at: Set a reminder for the task in granularity of weeks, days, hours, and minutes.
- Advanced: Determines whether this task uses the playbook default setting for Quiet Mode. When in Quiet Mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn Quiet Mode on or off at the task or playbook level.
- Details: Includes the following fields.
  - Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.
  - Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.
- Timers: Includes the following fields.
  - Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.
  - Add Trigger: You can add other trigger timer fields from the drop down.

Ask

- Message: Includes the following fields.
  - Ask by: The method for sending the message and survey. Options are:
    - Task (can always be completed directly in the Workplan)
    - Generated link (appears in the context data)
    - Email
  - To: The message and survey recipients. You can define by:
    - Selecting from a predefined drop down list.
    - Manually typing email addresses for users and/or external users.
    - Clicking the context icon to define recipients from a context data source.
  - CC of the email: A CC email address.
  - Subject of the email: The message subject that displays to message recipients. You can write the survey question in the subject field or in the message body field.
  - Message body: The text that displays in the body of the message. This field is optional, but if you don't write the survey question in the subject field, include it in the message body. This is a long-text field.
  - Reply options: Reply options are sent via the selected channels as options for an answer.
  - Require users to authenticate: Enable this option to have your SAML or AD authenticate the recipient before allowing them to answer. You must first set up an authentication integration instance and check Use this instance for external users authentication only in the integration instance settings.
- Timing: Includes the following fields.
  - Retry interval (minutes): Determines the wait time between each execution of a command. For example, the frequency (in minutes) that a message and survey are resent to recipients before the response is received.
  - Number of retries: Determines how many times a command attempts to run before generating an error. For example, the maximum number of times a message is sent. If a reply is received, no additional retry messages will be sent.
  - Task SLA: Set the SLA in granularity of weeks, days, and hours.
  - Set task Reminder at: Set a task reminder in granularity of weeks, days, and hours.
  - Complete automatically if SLA passed without a reply: Select this checkbox to complete the task if the SLA is breached before a reply is received. You can select yes or no.
- Advanced: Includes the following fields.
  - Using: Choose which integration instance will execute the command, or leave empty to use all integration instances.
  - Extend context: Append the extracted results of the action to the context. For example, "newContextKey1=path1::newContextKey2=path2" returns "[path1:'aaa',path2:'bbb', newContextKey1: 'aaa',newContextKey2:'bbb']"
  - Ignore outputs: If set to true, will not store outputs into the context (besides the extended outputs).
  - Execution timeout (seconds): Sets the command execution timeout in seconds.
  - Indicator Extraction mode: Choose when to extract indicators:
    - None: Do not perform indicator extraction
    - Inline: Before other playbook tasks
    - Out of band: While other tasks are running
  - Mark results as note
  - Mark results as evidence
  - Run without a worker
  - Skip this branch if this script/playbook is unavailable
  - Quiet Mode: When in quiet mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn quiet mode on or off at the task or playbook level.
- Details: Includes the following fields.

- Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.
- Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.

Choose script

From a drop down list, select a script for the playbook to run. In the following tabs you can set:

- Inputs: Each script has its own set of input arguments (or none). You can set each argument to a specific value (by typing directly on the line under the argument name) or you can click the curly brackets to define a source field to populate the argument.
- Outputs: Each script has its own set of output arguments (or none).
- Mapping:

Map the output from a playbook task directly to an incident field.

The value for an output key populates the specified field per incident. This is a good alternative to using a task with a set incident command.

#### NOTE:

The output value is dynamic and is derived from the context at the time that the task is processed. As a result, parallel tasks that are based on the same output may return inconsistent results.

1. In the Mapping tab, click Add custom output mapping.
  2. Under Outputs, select the output parameter whose output you want to map. Click the curly brackets to see a list of the output parameters available from the automation.
  3. Under Field to fill, select the field that you want to populate with the output.
  4. Click Save.
- Advanced: Includes the following fields.
    - Using: Choose which integration instance will execute the command, or leave empty to use all integration instances.
    - Extend context: Append the extracted results of the action to the context. For example, "newContextKey1=path1::newContextKey2=path2" returns "[path1:'aaa',path2: 'bbb', newContextKey1: 'aaa',newContextKey2:'bbb']"
    - Ignore outputs: If set to true, will not store outputs into the context (besides the extended outputs).
    - Execution timeout (seconds): Sets the command execution timeout in seconds.
    - Indicator Extraction mode: Choose when to extract indicators:
      - None: Do not perform indicator extraction
      - Inline: Before other playbook tasks
      - Out of band: While other tasks are running
    - Mark results as note
    - Mark results as evidence
    - Run without a worker
    - Skip this branch if this script/playbook is unavailable
    - Quiet Mode: When in quiet mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn quiet mode on or off at the task or playbook level.
  - Details: Includes the following fields.
    - Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.
    - Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.
  - Timers: Includes the following fields.
    - Timer.start: The trigger for starting to send a message or survey to recipients. You can change this trigger or add a trigger for Timer.stop or Timer.pause. Select the trigger timer field from the drop down.
    - Add Trigger: You can add other trigger timer fields from the drop down.
  - On Error: Includes the following fields.

- Number of retries: How many times the task should retry running if there is an error. Default is 0.
- Retry interval (seconds): How long to wait between retries. Default is 30 seconds.
- Error handling: How the task should behave if there is an error. Options are:
  - Stop
  - Continue
  - Continue on error path(s)

This option configures the task to handle potential errors that may occur when executing the current task's script.

#### 5. Click Save.

##### 10.4.1.5 | Create a communication task

###### Abstract

Communication tasks in playbooks enable you to send surveys and collect data. Ask task, data collection task.

Communication tasks enable you to send surveys to users, both internal and external, to collect data for an incident. The collected data can be used for incident analysis, and also as input for subsequent playbook tasks. For example, you can send a scheduled survey requesting analysts to send specific incident updates or send a single (stand-alone) question survey to determine how an issue was handled.

###### About ask tasks

An ask task is a type of conditional task that sends a single question survey, the answer to which determines how a playbook proceeds. If you send the survey to multiple users, the first answer received is used, and subsequent responses are disregarded. For more information about ask task settings, see Create a conditional task.

Because this is a conditional task, you need to create a condition for each of the answers. For example, if the survey answers include, Yes, No, and Maybe, there should be a corresponding condition (path) in the playbook for each of these answers.

Users interact with the survey directly from the message, meaning the question appears in the message and they click an answer from the message.

The survey question and the first response is recorded in the incident context data. This enables you to use this response as the input for subsequent playbook tasks.

For all ask conditional tasks, a link is generated for each possible answer the recipient can select. If the survey is sent to more than one user, a unique link is created for each possible answer for each individual recipient. These links are visible in the context data of the incident's Work Plan. The links appear under Ask.Links in the context data.

###### Example 11. Send a survey

In this example, the message and survey will be sent to recipients every hour for six hours, until a reply is received (it is repeated every 60 minutes, 6 times). The SLA is six hours. If the SLA is breached, the playbook will proceed according to the Yes condition.

TASK DETAILS ✖

Standard  Conditional  Data Collection  Section Header

◆ Was the Alert escalated?  ⓘ

Built-in  Manual  Ask  Choose automation

Message      Timing      Advanced      Details

Retry interval (minutes)      Number of retries  ⓘ

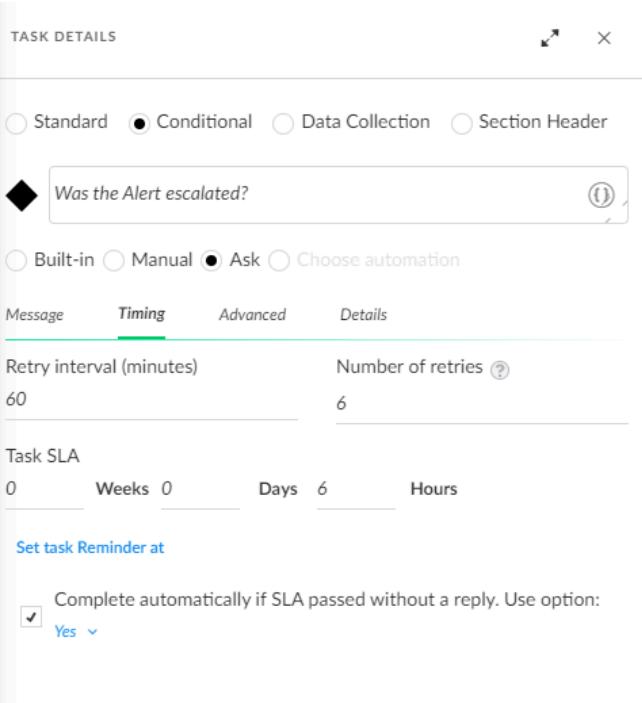
60      6

Task SLA

0 Weeks 0 Days 6 Hours

[Set task Reminder at](#)

Complete automatically if SLA passed without a reply. Use option:  
 Yes ▼



#### Example 12. Send email to users

In this example, a message and survey are sent by email to all users with the Analyst role. We are not including a message body because the message subject is the survey question we want recipients to answer. There are three reply options, Yes, No, and Not sure. In the playbook, we will only add conditions for the Yes and No replies. We require recipient authentication, which first involves setting up authentication.

**TASK DETAILS**

Standard  Conditional  Data Collection  Section Header

**email survey**

Built-in  Manual  Ask  Choose script

**Message**   **Timing**   **Advanced**   **Details**

**Ask by** [Preview](#)

Select communication options:

Task (can always be completed directly in the workplan)  
 Generated link (appears in the context data)  
 Email

\* To  
 [\(\)](#) [v](#)

CC of the email  
[Select from predefined values or add your own](#) [\(\)](#) [v](#)

\* Subject of the email  
 [\(\)](#)

Message body [Text](#)  
 [\(\)](#)

Reply Options [?](#)

Yes

No

[+ Add reply option](#)

Require users to authenticate [?](#)

#### Create a data collection task

The data collection task is a multi-question survey (form) that survey recipients access from a link in the message. Users do not need to log in to access the survey, which is located on a separate site.

All responses are collected and recorded in the incident context data, whether you receive responses from a single user or multiple users. This enables you to use the survey questions and answers as input for subsequent playbook tasks. If responses are received from multiple users, data for multi-select fields and grid fields are aggregated. For all other field types, the response received most recently will override previous responses as it displays in the field. All responses are always available in the context data.

For all data collection tasks, a single link is generated for each recipient of the survey. These links are visible in the context data of the incident's Work Plan. The links appear in the context data under the Links section of that survey.

You can include the following types of questions in the survey.

- Stand alone questions. These are presented to users directly in the message, and from which users answer directly in the message (not an external survey).
- Field-based questions. These are based on a specific incident field (either system or custom), for example, an Asset ID field. The response (data) received for these fields automatically populates the field for this Incident. For single-select field based questions, the default option is taken from the field's defined default.

#### How to create a Data Collection task

1. In a playbook, click + to create a new task.
2. Select the Data Collection option.

3. Enter a meaningful name in the Task Name field for the task that corresponds to the data you are collecting.

4. Select the communication options you want to use to collect the data.

Tabs and configuration fields

Tab	Configuration Fields In The Tab
Message	<ul style="list-style-type: none"> <li>• Ask by: The method for sending the message and survey. Options are:           <ul style="list-style-type: none"> <li>◦ Task (can always be completed directly in the Workplan)</li> <li>◦ Generated link (appears in the context data): A link to the data collection survey is available in the context data of the task.</li> <li>◦ Email: If you select this option, enter below the subject and message of the email and the email addresses of the users who should receive this message or survey.</li> </ul> </li> <li>• To: The message and survey recipients. You can define by:           <ul style="list-style-type: none"> <li>◦ Selecting from a predefined drop down list.</li> <li>◦ Manually typing email addresses for users and/or external users.</li> <li>◦ Clicking the context icon to define recipients from a context data source.</li> </ul> </li> <li>• CC of the email: A CC email address.</li> <li>• Subject of the email: The message subject that displays to message recipients. You can write the survey question in the subject field or in the message body field.</li> <li>• Message body: The message question body to be used in the notification sent to the given users along with the reply options.</li> <li>• Require users to authenticate: Enable this option to have your SAML or AD authenticate the recipient before allowing them to answer. You must first set up an authentication integration instance and check Use this instance for external users authentication only in the integration instance settings.</li> </ul>
Questions	<ul style="list-style-type: none"> <li>• Web Survey Title: The title displayed for the web survey.</li> <li>• Short Description: A description displayed above the questions on the web survey. Click Preview to see how it displays.</li> <li>• Question: A question to ask recipients.</li> <li>• Answer Type: The field type for the answer field. Options are:           <ul style="list-style-type: none"> <li>◦ Short text</li> <li>◦ Long text</li> <li>◦ Number</li> <li>◦ Single Select (requires you to define a reply option)</li> <li>◦ Multi select/Array (requires you to define a reply option)</li> <li>◦ Date picker</li> <li>◦ Attachments</li> </ul> </li> <li>• Mandatory: If this checkbox is selected for a question, survey recipients will not be able to submit the survey until they answer this question.</li> <li>• Help Message: The message that displays when users hover over the question mark help button for the survey question.</li> <li>• Placeholder: A sample value displayed until a real value is entered.</li> </ul> <p><b>NOTE:</b></p> <p>You can drag questions to rearrange the order in which they display in the survey.</p>

Tab	Configuration Fields In The Tab
Timing	<ul style="list-style-type: none"> <li>Retry interval (minutes): Determines the wait time between each execution of a command. For example, the frequency (in minutes) that a message and survey are resent to recipients before the response is received.</li> <li>Number of retries: Determines how many times a command attempts to run before generating an error. For example, the maximum number of times a message is sent. If a reply is received, no additional retry messages will be sent.</li> <li>Task SLA: Set the SLA in granularity of weeks, days, and hours.</li> <li>Set task Reminder at: Set a task reminder in granularity of weeks, days, and hours.</li> <li>Complete automatically if: <ul style="list-style-type: none"> <li>Reached task SLA (with or without a reply): This option is grayed out.</li> <li>Received &lt;enter a number&gt; reply</li> </ul> </li> </ul>
Details	<ul style="list-style-type: none"> <li>Tag the result with: Add a tag to the task result. You can use the tag to filter entries in the War Room.</li> <li>Task description (Markdown supported): Provide a description of what this task does. You can enter objects from the context data in the description. For example, in a communication task, you can use the recipient's email address. The value for the object is based on what appears in the context every time the task runs.</li> </ul>
Advanced	<ul style="list-style-type: none"> <li>Using: Choose which integration instance will execute the command, or leave empty to use all integration instances.</li> <li>Extend context: Append the extracted results of the action to the context. For example, "newContextKey1=path1::newContextKey2=path2" returns "[path1:'aaa',path2: 'bbb', newContextKey1:'aaa',newContextKey2:'bbb']"</li> <li>Ignore outputs: If set to true, will not store outputs into the context (besides the extended outputs).</li> <li>Execution timeout (seconds): Sets the command execution timeout in seconds.</li> <li>Indicator Extraction mode: Choose when to extract indicators: <ul style="list-style-type: none"> <li>None: Do not perform indicator extraction</li> <li>Inline: Before other playbook tasks</li> <li>Out of band: While other tasks are running</li> </ul> </li> <li>Mark results as note</li> <li>Mark results as evidence</li> <li>Run without a worker</li> <li>Skip this branch if this script/playbook is unavailable</li> <li>Quiet Mode: When in quiet mode, tasks do not display inputs and outputs or extract indicators. Errors and warnings are still documented. You can turn quiet mode on or off at the task or playbook level.</li> </ul>

5. (Optional) To customize the look and feel of your email message, click Preview.

You can determine the color scheme and how the text in the message header and body appear, as well as the appearance and text of the button the user clicks to submit the survey.

Data collection task examples

Stand-alone question with a single-select answer

In this example, we create a stand-alone question, with a single-select answer. This question is not mandatory. If we selected the First option is default checkbox, the reply option "0" is the default value in the answer field.

TASK DETAILS

\* Question ⓘ  
How many alerts did you close in this shift? ⓘ

\* Answer Type ⓘ  
▼ Single select  Mandatory

Reply Options ⓘ

0	ⓘ	<input checked="" type="checkbox"/> First option is default
1-3	ⓘ	ⓘ
4-5	ⓘ	ⓘ
6	ⓘ	ⓘ

+ Add reply option

Field-based using a custom field

In this example, we create a question based on a custom grid field that we mark as mandatory. For the question field, we included a descriptive sentence explaining how to fill in the grid.

TASK DETAILS

+ Add reply option

Help Message ⓘ

QUESTION 2: PLEASE DETAIL THE CLOSED INCIDENTS FOR YOUR SHIFT.

\* Question ⓘ Detach from field  
Please detail the Closed Incidents for your shift. ⓘ

\* Field associated with this question ⓘ  
Analyst Status Report: Closed Incidents  Mandatory

Help Message ⓘ

Placeholder ⓘ

+ Add Question + Add Question based on field ⓘ Top of page

Cancel Save

The screenshot shows the 'Task Details' dialog box. At the top, there's a header with a close button and a 'Help Message' link. Below the header, a question is defined: 'QUESTION 2: PLEASE DETAIL THE CLOSED INCIDENTS FOR YOUR SHIFT.' This question is marked as mandatory. It has a 'Question' field containing the text 'Please detail the Closed Incidents for your shift.', a 'Field associated with this question' dropdown set to 'Analyst Status Report: Closed Incidents', and a 'Mandatory' checkbox checked. There are also 'Help Message' and 'Placeholder' fields. At the bottom of the dialog, there are buttons for '+ Add Question', '+ Add Question based on field', and 'Top of page'. A 'Cancel' button and a prominent 'Save' button are at the very bottom.

#### Configure communication task authentication

When sending a form in a communication task, you can configure user authentication to ensure only authorized users gain access to the form.

The authorized users are usually external users not in Cortex XSOAR, and they will not be able to access anything else in Cortex XSOAR.

#### Set up playbook communication task authentication

1. Set up your SSO if it is not already configured. See [Authenticate users using SSO](#) for more details.
2. In the Task details of your playbook communication task, check [Require users to authenticate](#) to have your SAML or AD authenticate the recipient before allowing them access to the form.

**TASK DETAILS**

Standard  Conditional  Data Collection  Section Header

#1 Untitled Task (i)

Attributes: Quiet Mode

**Message** Questions Timing Details Advanced

Ask by Preview

Select communication options:

Task (can always be completed directly in the workplan)  
 Generated link (appears in the context data)  
 Email

\* To  
 Select from predefined values or add your own (i) ▾

CC of the email  
 Select from predefined values or add your own (i) ▾

\* Subject of the email  
(i) ▾

Message body (i) Text

(i) ▾

Link to web form will be placed automatically at the bottom of your message

Require users to authenticate (i)

#### 10.4.1.6 | Configure script error handling in a playbook

##### Abstract

When defining a task, you can decide if the playbook continues, stops, or continues on an error path.

You can determine how the playbook behaves if there are script errors during execution.

When defining a standard task that uses a script or a conditional task that uses an script, you can define how a playbook task continues by selecting one of the following options:

- Stop: The playbook stops, if the task errors during execution. For example, if the task requires a manual review, you may want the playbook to stop until completion.
- Continue: The playbook continues to execute if the task errors. For example, the playbook task requires EWS, but EWS is not required for the playbook to proceed.
- Continue on error path: If a task errors, the playbook continues on an error path.

The error path may be useful if you want to take action on an error, like clean-up, retry, etc. You may also want to handle errors in different ways. For example, in case of a quota expired error you may want to retry in 1 minute, but if you receive an internal error 500, you may want to stop the playbook.

You may want to create a separate path when an analyst manually reviews the incident and research is needed outside Cortex XSOAR. Once an analysis is complete, you can add a task to consider escalating to a customer and, if so, generate a report which can be attached to a ticket system such as Jira or ServiceNow.

Instead of a playbook waiting on manual input, which displays an error state, such as missing an argument in a script, you can add a separate path for these kinds of issues.

##### NOTE:

Use the `GetErrorsFromEntry` script (part of the Common Scripts Pack) to check whether the given entry returns an error and returns an error message. For example, when using the script in a playbook, you can fetch the error message from a given task, such as a runtime error. You can then add a step in the playbook flow to send those error messages to the relevant stakeholder through Slack, email, opening a Jira ticket, etc.

When errors are created, they are added to context under **task.id.error**.

How to set up error handling in your playbook

1. When adding the connector from this task to the following task, a dialog box appears which enables you to select one of the following paths:

In a playbook, edit or create a new task by clicking +.

If the task library is expanded, click + Create Task.

2. Select the task type you want to create or edit.

**NOTE:**

You can set up script error handling when running a script in a Standard task or a Conditional task. For more information about error handling settings for these tasks, see Playbook tasks.

Built-in Conditional tasks have On Error settings for number of retries and retry interval, but not Error Handling.

3. For new tasks, in the Task Name field, type a meaningful name for the task that corresponds to the data you are collecting.

4. Click the On Error tab.

5. In the Number of retries field, type the number of times the tasks attempts to run before generating an error.

6. In the Retry Interval (seconds) field, type the wait time between retrying the task.

7. In the Error Handling field, select one of the following:

- Stop
- Continue
- Continue on error path(s)

8. Click Save.

9. When adding the connector from this task to the following task, a dialog box appears which enables you to select one of the following paths:

- Standard Path: When adding a task to this path, it executes without any exceptions.

If you select the Standard Path, the task continues on this path and executes without exceptions.

- Error Path: When adding a task to this path, it executes where the source task errors during execution.

If you select Error path, if the task errors, the playbook continues with this path.

#### 10.4.2 | Customize a playbook for a phishing use case example

Abstract

Customize an existing playbook based on your organization's needs.

If your use case involves investigating a type of phishing event, you can customize a playbook from the Phishing content pack, for example the Phishing - Generic V3 playbook. For an overview of the Phishing - Generic V3 playbook, go to minute 4:06 in this video.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

Do the following to customize the Phishing - Generic v3 playbook.

Task 1. Edit the playbook

1. Go to Playbooks and search for Phishing - Generic v3.
2. To edit the playbook (add/delete tasks, sub-playbooks, or the flow, etc.) choose Detach Playbook from the three button menu.

**NOTE:**

If you want to receive content updates for the playbook, duplicate rather than detach the playbook. If you duplicate the playbook, change the default playbook for the Phishing incident type from Phishing - Generic v3 to your new playbook name.

Task 2. Review and change the playbook inputs

Click Playbook Triggered at the top of the playbook.

1. Role: By default, the least busy user is assigned to the incident. If you set a role, the incident will only be assigned to users with that role.
2. SearchAndDelete: Turn on or off the Search and Delete Process in EWS O365. This is part of the remediation process. If set to true, in case of a malicious email, the Search and Delete sub-playbook looks for other instances of the email and deletes them pending analyst approval. We will leave the default option, **False**.
3. BlockIndicators: This is part of the remediation process. It automatically blocks malicious indicators in relevant integrations. For our example, we keep this as **False**.
4. AuthenticationEmail: Whether to authenticate the email. Leave as the default, **true**. See **Authenticate the email** under the investigation stage.
5. OnCall: Whether to assign a user that is currently on shift. Change this to **True**.
6. SearchAndDeleteIntegration: This setting is only relevant if SearchAndDelete is set to **true**. If you later decide to use the SearchAndDelete option, change the integration here from **EWS** to **O365**.  
If you use enable SearchAndDelete and set the SearchAndDeleteIntegration to O365, continue with the O365 inputs such as O365DeleteType.
7. CheckMicrosoftHeaders: If using EWS O365, the Bulk Confidence Level (BCL), Spam Confidence Level (SCL) and the Phishing Confidence Level (PCL) values on the Microsoft headers are considered as part of severity and email classification (whether the email is spam). These values help security teams determine whether the email is coming from a spam, phishing or bulk sender. Leave as the default, **True**.
8. InternalDomains: When the Email Address Enrichment Generic v2.1 sub-playbook runs, it uses the internal domain entered here to determine if the email was reported from an internal or external email address.
9. GetOriginalEmail: Used to retrieve the original email, when the phishing email is forwarded and not attached. Change this to **True** only if you have permissions in EWS O365 to execute global search (eDiscovery). This input is used to determine if the Get Original Email - Generic v2 sub-playbook should run.
10. Click Save.

After the playbook starts, the detection timer begins, which detects how long it takes to detect the incident. You can change which timer to start but we will use the default.

#### Task 3. Review the engage with user stage of the playbook

The Engage with User stage stores the name of the user. The Email Address Enrichment - Generic v2.1 sub-playbook here receives the email address of the user reporting the phishing email. If there is an email address from an internal domain, the sub-playbook uses Active Directory to find the name of the user reporting the phishing email. When we engage with the user, we can then address them by name.

The playbook does two tasks simultaneously:

- Engages with the user
- Triage

An email is sent to the reporting user acknowledging that the incident was received. You can update the message (body) by clicking the Acknowledge incident was received task.

#### Task 4. Review the triage section of the playbook

The Triage section extracts relevant information such as indicators from a file, detonates the file, and uses machine learning to predict the phishing type and update the incident with predictions. Triage includes the following tasks:

- **Process Email - Generic v2**

Triage starts with the **Process Email - Generic v2** playbook, which processes the email and extracts relevant information from files including attachments. This playbook branches according to whether the email is attached. To view the playbook, hover over the task and then click the eye icon below the task.

- **Extract email artifacts and attachments** task

Email artifacts and attachments are extracted by this task, which uses the `ParseEmailFilesV2` script. This task takes the email file and extracts all email addresses, subject, file attachments, etc. The task does not finish until everything has been extracted (inline). The results are then entered into the incident layout.

- **Get Original Email - Generic v2** sub-playbook

While best practice is to attach a file containing the email when reporting spam or phishing, in some cases, users will forward the email instead.

If the original email is necessary for the investigation and not attached and the `GetOriginalEmail` input in the main playbook is set to `True`, the Get Original - Generic v2 sub-playbook obtains the original email and then adds details to the incident, such as sender, text body, size, body, etc.

- **Headers** section

If email headers were extracted, the headers are displayed (you can later use them for authentication). At the same time attachment information is added, such as size, MD5 hashes, number of attachments, etc.

- **Email Screenshot** section

If the email is HTML-formatted (not in all cases), it creates an image out of that HTML data to show how the email was seen by the user using `Rasterize`.

- After extracting the relevant information, we return to the Phishing - Generic v3 main playbook. The following tasks are undertaken at the same time:

- **Detonate File - Generic** sub-playbook

Files are executed in an isolated sandbox environment and their behavior is analyzed. Any sandbox integrations that you have enabled run and provide details about the file reputation, whether the email is forwarded or attached as a file. If, for example, the email is forwarded (not attached as a file), if the email contains a PDF file, then that PDF file will go through detonation/indicator extraction. In this example we use **Palo Alto Networks WildFire** to detonate files. The payload is triggered so files can be isolated and analyzed.

**NOTE:**

Detonation is different from file enrichment. For example, VirusTotal file enrichment provides reputation information about the file, based on the file hash. If VirusTotal does not recognize the file hash, no enrichment information is returned, unless the actual file is submitted for analysis. A sandbox integration, by contrast, runs the file in an isolated environment and provides exact information about the file execution.

If your sandbox integration is not here, you can add it.

- **Detonate URL** sub-playbook

This sub-playbook uses integrations to safely detonate URLs in a sandbox environment and analyze the website behavior.

- **Extract Indicators from File - Generic v2** sub-playbook

If a file is attached, you can extract more indicators from the file. Indicators may exist in the file and not the email. You can extract text based files, Word, PDF, or other supported files, like PPT files (sometimes these contain executable macros). For PDFs, we utilize the image OCR, which extracts text from images inside PDF files.

- **Phishing - Machine Learning Analysis** sub-playbook

The Phishing - Machine Learning Analysis sub-playbook performs two functions.

- Predict Phishing Type - The playbook uses your custom trained phishing model to predict the phishing type. If you do not have a custom trained phishing model, it uses a pre-trained out-of-the-box phishing model instead. If the model is able to predict the phishing type, the incident is updated with the prediction.
    - Predict Phishing URLs - If the `Rasterize` integration is enabled, the `DBotPredictURLPhishing` script predicts phishing URLs.

#### Task 5. Set up indicator enrichment

After extracting indicators from an email and a detonated file (if there is a file attachment) we need to enrich the indicators. This gives us additional information about the indicators that have been extracted. The Entity Enrichment - Phishing v2 playbook contains the following sub-playbooks:

- **File Enrichment - Generic v2:** Enriches the file using the File Enrichment - VirusTotal (API v3) playbook. If there is a SHA256 hash and Cylance Protect v2 is enabled, it enriches the file using Cylance Protect v2.
- **IP Enrichment - External - Generic v2:** Checks whether there are internal and external IP addresses and enriches external IP addresses using the !IP command, VirusTotal automation (if enabled) and Threat Crowd (if enabled).
- **Email Address Enrichment - Generic v2.1:** Checks whether email addresses are internal or external. For internal email addresses, additional information is retrieved using Active Directory. For external email addresses, this sub-playbook checks for a domain list input and for domain squatting (such as using a similar domain).
- **URL Enrichment - Generic v2:** Checks for URLs, verifies SSL & captures screenshots using the Rasterize integration.
- **Domain Enrichment - Generic v2:** Enriches domain using Cisco Umbrella (if enabled), and VirusTotal (if enabled).

As we are using VirusTotal, there is no need to customize these sub-playbooks. If you use different enrichment integrations these need to be added. These playbooks add to the DBot score (reputation of the indicator).

#### Task 6. Review the investigation section

- Microsoft's Headers Check

At the beginning of the playbook (Playbook Triggered), in the Inputs section, we left CheckMicrosoftHeaders as the default, **True**.

The Process Microsoft's Anti-Spam Headers playbook finds the SCL, BCL and PCL values (if they exist) in the Microsoft headers, calculates the severity based on those scores and classifies whether the email is spam or phishing. You can change the minimum severity of each score.

- Email Campaign Search

The Detect & Manage Phishing Campaigns sub-playbook uses the **FindEmailCampaign** automation, which utilizes machine-learning to identify existing incidents in Cortex XSOAR that are part of the same campaign of the currently investigated incident. You can customize the inputs as required.

If the sub-playbook finds that the incident is part of a campaign, it generates campaign-related data which you can observe in the linked Phishing Campaign incident, and take actions related to the campaign.

- Email Authenticity Check

At the beginning of the playbook (Playbook Triggered), in the Inputs section, we left AuthenticateEmail as the default, **True**.

Using DKIM, DMARC and SPF we check to see if the email is coming from its alleged source, or whether the email has been tampered with.

The result of the authenticity check is added to the incident field using the **setIncident** script.

- Domain-squatting

If domain-squatting occurred, the result is saved to the incident field.

- Email Indicators Hunting

At the beginning of the playbook (Playbook Triggered), in the Inputs section, we left HuntEmailIndicators as the default, **True**.

The Phishing - Indicators Hunting sub-playbook runs to hunt malicious indicators found in other emails and optionally, automatically create new incidents for each found email if EmailHuntingCreateNewIncidents is set to **True**.

The results of the previous tasks are used by the **Calculate Severity - Generic v2** sub-playbook, which calculates and assigns the incident severity based on the highest returned severity level from the following:

- Authenticity of the email (whether it passed the authenticity check).
- Severity of the critical assets according to the **Calculate Severity - Critical Assets v2** sub-playbook. This playbook checks critical users, critical user groups, critical groups, critical endpoint groups or critical endpoints. You can define the critical users in your organization by editing the inputs. If one critical entity is involved in the incident, it will raise the severity to critical.
- The current incident severity - (if it already has a severity level).
- The DBot Score from tasks that run in the parent playbook or sub-playbooks, (such as process email, extract indicators from file, detonation playbooks, machine learning, etc.)
- The Microsoft Headers Severity (if a value is returned).

The incident severity is determined by the highest returned score.

The incident is now assigned to an analyst. Incidents can be assigned according to a role such as Analyst, by the least busy user (less-busy-user), randomly, by user online, etc. For this task, by default, the incident is assigned to the least busy user.

The final task in the investigation section determines Is the email malicious?

The incident severity determines if the email is malicious. If the severity is equal or greater than 2 (medium is 2, high is 3, critical is 4), it is considered malicious. We can change this criteria if necessary.

#### Task 7. Handle email as undetermined or malicious

This stage of the process depends on whether the email is undetermined or malicious.

- **Undetermined**

This is a manual task. If the severity is low or unknown it is regarded as undetermined. The analyst manually reviews the incident and decides whether it is malicious. If not, the analyst updates the user (who sent the email) that the email is safe and then closes the investigation.

- **Malicious email**

If malicious, we update the user that the email is malicious and then start the remediation process. If the incident was part of a phishing campaign we also update the user that the email is part of a malicious campaign.

#### Task 8. Set up remediation

The last part of the process is remediation. A timer starts at this point to track remediation time.

- The **Search and Delete Emails Generic v2** sub-playbook searches and deletes the email from all users across the organization. This sub-playbook runs if the original email was retrieved and SearchAndDelete is set to **True** in the playbook inputs. During setup, we kept the default setting, **False**. If you decide to use this playbook with O365, change the SearchAndDelete setting to **True**, change SearchAndDeleteIntegration from **EWS** to **O365**, and configure the O365 - Security And Compliance - Content Search v2 integration.
- The **Block Indicators - Generic v2** sub-playbook contains sub-playbooks to blocks IPs, files, emails, and domains. For example, the Block IP - Generic v2 sub-playbook blocks IP addresses using one or more of the following integrations (depending on which integrations you have configured) PAN-OS, MineMeld, Zscaler, CheckPoint FW, and Fortinet. You can also customize the playbook to add additional integrations.
- Manually remediate the incident.

To choose the remediation method(s), go to the Playbook Triggered task (at the beginning of this playbook), and set the BlockIndicators and the SearchAndDelete fields. If one or both are set to true, the playbook follows those branches. If the email is found to be malicious, the analyst assigned to the incident is prompted to manually remediate the incident, regardless of whether the search and delete emails and/or block indicators branches are executed.

After finishing the remediation section, the timer stops and the investigation is closed.

## 10.5 | Customize your playbook

### Abstract

Customize your playbook to extract indicators, extend context, add incident fields, filter and transform data, run scripts, and perform triggered actions, sub-playbook loops, and polling.

You can customize your playbook to do the following.

Custom Action	Description
Customize the SOC name	Customize the name of the SOC that appears in the survey header.
Add a sub-playbook	Sub-playbooks are playbooks that are nested under other playbooks.
Filter and transform data	Filters extract relevant data to help focus on relevant information and discard irrelevant or unnecessary data. Transformers take one value and transform or render it to another value or format.
Use scripts	Perform specific automated actions using commands that are also used in playbook tasks and in the War Room. Configure script error handling.
Extract indicators	Extract indicators from incident fields and enrich them using commands and scripts defined for the indicator type.

Custom Action	Description
Extend context	Save additional data from the raw response of commands that return data.
Set and update incident fields	Use the setIncident script in a playbook task to set and update incident fields.
Use playbook polling	Configure a playbook to stop and wait for a process to complete on a third-party product, and continue when it is done.

### 10.5.1 | Configure general playbook settings

#### Abstract

Configure the name, description, tags, and Quiet Mode as well as enable or disable a playbook.

You can edit general playbook settings such as the name of the playbook, who can edit and run the playbook, and whether Quiet Mode is turned on.

1. Go to Playbooks and click the playbook that you want to edit.
2. If it is a content pack playbook, detach or duplicate the playbook by clicking the ellipsis icon.

If you detach the playbook and want to keep any changes, ensure that you duplicate the playbook before reattaching.

3. Click the settings wheel icon.

4. Edit the following settings as relevant.
  - a. In the BASIC section, change the name and description.

**NOTE:**

You cannot change the name of a detached playbook.

- b. Add any tags as required by either typing a new tag or selecting from the list.

Tags help you search for a particular playbook, such as Malware.

- c. Add roles for edit access to the playbook.
- d. If you want to disable a playbook, deselect the Enabled checkbox.

If disabled, you cannot associate it with an incident or an incident type.

- e. In the ADVANCED section, determine whether the playbook runs in quiet mode.

When Quiet Mode is selected, playbook tasks do not display inputs and outputs and do not extract indicators.

Playbook tasks are not indexed so you cannot search on the results of specific tasks. All of the information is still available in the context data, and errors and warnings are written to the War Room.

**TIP:**

Quiet mode is recommended for scenarios that involve a lot of information that might adversely affect performance, for example, processing indicators from threat intel feeds.

In the War Room, you can run the `!getInvPlaybookMetadata` command to analyze the size of playbook tasks in a specific incident Work Plan to determine whether to implement quiet mode for playbooks or tasks.

5. Click Save all tabs.

### 10.5.2 | Customize the SOC name

#### Abstract

Add a server configuration to customize the name of the security operations center (SOC) that appears in communication tasks.

The default name that appears in the survey header is Your SOC team. Follow these steps to customize the name of the SOC.

1. Go to Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.
2. Add the `soc.name` server configuration, and the display name of your SOC as the value.

This name is used in the default message and email of the communication tasks, and the web survey for all communication tasks.

### 10.5.3 | Configure a sub-playbook

#### Abstract

Configure a sub-playbook, also to run in a loop.

Playbooks can be divided into two categories, depending on their use.

- Parent playbooks are playbooks that run as the main playbook of an incident. For example, Phishing - Generic v3 and Malware Investigation & Response Incident Handler.
- Sub-playbooks are playbooks that are nested under other playbooks. They appear as tasks in the parent playbook flow and are indicated by the sub-playbook icon . A sub-playbook can also be a parent playbook in a different use case. For example, IP Enrichment - Generic v2 and Retrieve File From Endpoint - Generic v3. These playbooks are usually used as part of a bigger investigation.

Since sub-playbooks are building blocks that can be used in other playbooks and use cases, you should define generic inputs for them.

Inputs can be passed to sub-playbooks from the parent playbook, used and processed in the sub-playbook, and sent as output to the parent playbook.

#### NOTE:

Any change made to a sub-playbook impacts the parent playbook in the next run of the parent playbook.

See this video for an example of creating a sub-playbook.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

#### Sub-playbook loops

Looping uses sub-playbooks to create loops within a parent playbook. When running the loop, the values are calculated based on the context data for the sub-playbook and not the parent playbook.

#### NOTE:

Consider the following when adding a loop:

- The maximum number of loops (default is 100). A high number of loops or a high wait time combined with a large number of incidents may affect performance.
- Periodically check looping conditions to ensure they are still valid for the data set.
- When the task input is an array, it is iterated automatically (no need to define a loop).

How to create a sub-playbook loop

1. In the Playbooks page, select the parent playbook that contains the sub-playbook task you want to run in a loop.

2. Click Edit.

If the playbook is installed from a content pack, you need to either detach or duplicate the playbook before editing.

3. Select the task that contains the sub-playbook for which you want to create the loop.

4. Click the Loop tab.

5. Click one of the following options to define loop settings:

- None: (Default) The sub-playbook does not loop.
- Built-in: Use built-in functions to define loop settings:

Option	Description
Exit when	Enables you to define when to exit the loop. Click {} and expand the source category. Hover over the required source and click <b>Filter &amp; Transform</b> to the left of the source to manipulate the data.
Equals (String)	Select the operator to define how the values should be evaluated.
Max iterations	The number of times the loop should run.
Sleep	The number of seconds to wait between iterations. recommends that you balance between the number of iterations and the number of seconds to wait between iterations so you don't overload the server.

- For each input: Runs the sub-playbook based on defined inputs. Enter the number of seconds to wait between iterations.
- Choose Loop automation: Select the automation from the drop-down list to define when to exit the loop. The parameters that appear are applicable to the selected automation.

6. To save the changes, click OK.

#### Example: Exit looping after running for each input

In the parent playbook (the task that contains the sub-playbook), you can configure to exit a loop running the sub-playbook automatically when the last item in the sub-playbook input is executed.

- If the input is a single item, the sub-playbook runs once, but if the input is a list of items (such as a list of incident IDs), the sub-playbook runs as many times as there are items in the list. Each iteration of the sub-playbook uses the next item in the list as the input.
- If there are multiple input lists with the same amount of items, the sub-playbook runs once for each set of inputs.
- If there are multiple input lists with different amounts of items, the sub-playbook runs the first set of inputs, followed by the second, third, and so on, until the end.

For example:

Input	Value
Input x	1,2,3,4
Input y	a,b,c,d
Input z	9

The first loop: 1, a, 9

The second loop: 2, b

The third loop: 3, c

The fourth loop: 4, d

The following example shows how a sub-playbook loop works using the Palo Alto Networks Cortex XDR - Investigation and Response integration.

After you install the Palo Alto Networks Cortex XDR - Investigation and Response content pack, configure the Palo Alto Networks Cortex XDR - Investigation and Response integration to fetch incidents. By default, the integration uses the Cortex XDR classifier, which automatically classifies Cortex XDR incident types. In this example, we are using the Cortex XDR incident type which runs the Cortex XDR incident handling v3 playbook.

**NOTE:**

Verify the integration is enabled to fetch incidents.

1. Go to Incidents, open a Cortex XDR incident, and go to the Work Plan tab.

You can see the incident uses the Cortex XDR incident handling v3 playbook.

2. The playbook starts retrieving incident data from Cortex XDR and finds similar incidents by fields. If similar incidents are found, the analyst can close them as duplicates.

3. If the alert is not a duplicate, the playbook continues to Loop on alert id - Alert enrichment.

4. The playbook runs the Cortex XDR Alerts Handling sub-playbook in a loop, by categorizing and enriching alerts until completion.

- Under the Inputs Results tab, you can see the `alert_ID` that the playbook processes.

Results (4)	Comments (0)	Errors (0)	<u>Input Results (2)</u>	Outputs (7)	Duration
incident_id	5413				
alert_id	[ "1640680", "1640666", "1640677", "1640665", "1640653", "1640657", "1640652", "1640642", "1640641", "1640640", "1640632" ]				

To view the looping settings, go to Playbooks and open the Cortex XDR Alerts Handling playbook. In the Inputs tab, view the playbook returns incident and alerts IDs. In the Loop tab, the For Each Input option is selected. This means the playbook iterates over all defined playbook inputs until complete.

- The playbook determines if the alert is malware, a port scan, or anything else and enriches according to the category.
  - If the alert is malware, the Malware sub-playbook runs.
  - If the alert is a port scan, the Port Scan sub-playbook runs.
  - If the alert is not malware or port scan, the playbook completes the processing.
- The applicable sub-playbook processes the enriched information and outputs the problematic endpoints.
- After completing the processing of an alert ID, the playbook iterates through the remaining inputs until all alert IDs have been processed (looping).
- Go to the Cortex XDR Alerts Handling playbook task and click the Results tab. You can see information returned and the number of times the playbook looped.

The screenshot shows the Cortex XDR interface. At the top, there's a 'Task details' section with a green checkmark icon. Below it is a table with columns for variable names and values, including file\_signature\_vendor\_name (Microsoft Corporation), file\_wildfire\_verdict (BENIGN), is\_malicious (false), is\_manual (false), is\_process (true), low\_confidence (false), and type (HASH). Below this is a log history section for a DBot entry dated April 13, 2022, at 11:19 AM. The log message says '#55: Choose playbook by category' and provides a command: 'Executing conditions: Label: Malwa...'. It also states 'Condition result: "None"'. Below this is another DBot entry for task #15, also dated April 13, 2022, at 11:19 AM. This log message says '#15: Cortex XDR Alerts Handling' and indicates 'ForEach Loop 15 completed 4 times'.

## 10.5.4 | Filter and transform data

### Abstract

Use filters and transformers to manipulate data. Use filters and transformers in playbook tasks or when mapping an instance.

In Cortex XSOAR, data is extracted and collected from various sources, such as playbook tasks, command results, and fetched incidents, and presented in JSON format. The data can be manipulated by using filters and transformers.

### Filters

Filters enable you to extract relevant data which you can use elsewhere in Cortex XSOAR. For example, if an incident has several files with varying file types and extensions, you can filter the files by file extension or file type, and use the filtered files in a detonation playbook. You can filter as many objects as required. Cortex XSOAR automatically calculates the context root to which to filter. You can change the context root as necessary.

#### CAUTION:

You can change the context data root to filter, but it is not recommended to select a different root, as it affects the filter results. The drop-down list displays the filter root for backward compatibility.

### Transformers

Transformers modify or format data to make it suitable for further processing or presentation. For example, you can convert a date in non-Unix format to Unix format. Another example is applying the **count** transformer, which renders the number of elements. When you have more than one transformer, they apply in the order that they appear. You can reorder them using click-and-drag.

Add filters and transformers in a playbook task

1. Create or edit a playbook task.
2. In the field you want to add a filter or transformer (for example, inputs or outputs), click the curly brackets and then select Filters and Transformers.
3. In the Get field, type or select data you want to filter or transform. For example, `EWS.Items.Name`.
4. (Optional) To filter the data, do the following.
  - a. In the Filter section, click Add filter.

When adding a filter, the context root to filter is automatically populated.

- b. Select the data you want to filter.
- c. Select the filter operators.
- d. Add the value.
- e. Click the checkbox to save the filter.

5. (Optional) To apply transformers to the field, click Add transformer.

- a. Click the transformer and select the relevant transformer.

By default, the transformer is set to `To_upper_case(String)`. Click it to pick a different transformer, for example to change the date format for when incidents occurred.

- b. Select the transformer operators.
- c. Click the tick box to save.

6. (Optional) To test the filter or transformation click Test and select the investigation or add it manually.

Example: Filter items with an EXE extension

In this example, we want to filter all EWS Item names that have the extension `exe`.



1. From the Filters & transformers window, in the Get field, type `EWS.Items.Name` to extract all Item names in EWS.

The context root to filter is **EWS.Items**.

**2 Filter** (Get subset of the data, e.g. File.Type is PDF) **EWS.Items** ▾  
Where all of the following are true for **EWS.Items**

2. In the Filter section, click Add filter.
3. In the left-hand side, add **Extension** to the filter.
4. Select Equals (String) → ignore case.
5. In the right-hand side add **exe**.

**2 Filter** (Get subset of the data, e.g. File.Type is PDF) **EWS.Items**  
Where all of the following are true for **EWS.Items**

General	String	Number
<a href="#">Contains</a>	<a href="#">Doesn't end with</a>	<a href="#">Doesn't equal</a>
<a href="#">Doesn't Contain</a>	<a href="#">Doesn't equal</a>	<a href="#">Equals</a>
<a href="#">Has length of</a>	<a href="#">Doesn't include</a>	<a href="#">Greater or equal</a>
<a href="#">In</a>	<a href="#">Doesn't start with</a>	<a href="#">Greater than</a>
<a href="#">Is defined</a>	<a href="#">Ends with</a>	<a href="#">Less or equal</a>
<a href="#">Is empty</a>	<a href="#">Equals</a>	<a href="#">Less than</a>
<a href="#">Is not empty</a>	<a href="#">Has length</a>	
<a href="#">Not defined</a>	<a href="#">In list</a>	

Ignore case

6. Click the tick box to save the filter.

7. Click Test.

You should see Item names are filtered with the extension **exe**.

Example (advanced): Filter hostname for the last resolved time

In this example, we want to see the **LastResolved** time only from the **demisto.com** host name.

This is part of the data where we want to filter:

```
{
  "IP": [
    {
      "Address": "192.168.10.96",
      "AutoFocus": {
        "Resolutions": [
          {
            "Hostname": "79463wwfqq.dattolocal.net",
            "LastResolved": "2022-08-02 04:01:02"
          },
          {
            "Hostname": "demisto.com",
            "LastResolved": "2022-09-10 09:47:17"
          },
          {
            "Hostname": "securesense.call4pchelp.com",
            "LastResolved": "2022-04-22 11:49:06"
          }
        ]
      }
    },
    {
      "Address": "192.168.10.96",
      "AutoFocus": {
        "Resolutions": [
          {
            "Hostname": "79463wwfqq.dattolocal.net",
            "LastResolved": "2022-08-02 04:01:02"
          },
          {
            "Hostname": "securesense.call4pchelp.com",
            "LastResolved": "2022-04-22 11:49:06"
          }
        ]
      }
    }
  ]
}
```

```

        "Hostname":"demisto.com",
        "LastResolved":"2022-09-10 09:47:17"
    },
    {
        "Hostname":"securesense.call4pchelp.com",
        "LastResolved":"2022-04-22 11:49:06"
    }
]
}
]
}
]
}

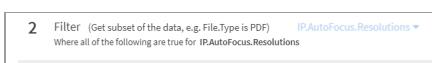
```

- From the Filters & transformers window, in the Get field, type **IP.AutoFocus.Resolutions.LastResolve**.



- In the Filter section, click Add filter.

Cortex XSOAR automatically calculates that the context root to filter is **IP.AutoFocus.Resolutions**.

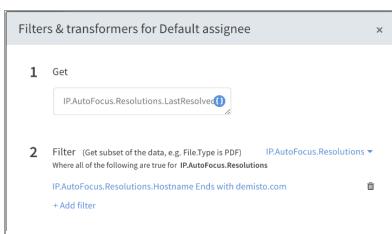


- In the left-hand side, add **Hostname** to the filter.

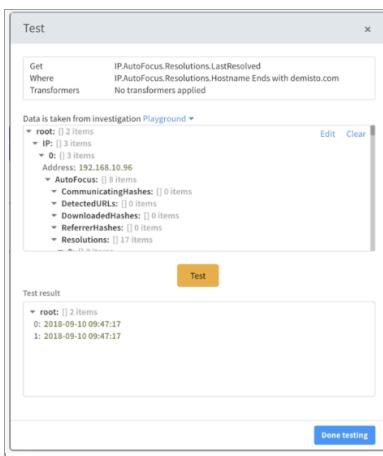
- Select Equals (String) → Ends with

- In the right-hand side add **demisto.com**.

- Click the checkbox to save.



- Click Test.



#### Create custom filters and transformers

If you require a filter or transformer that is not provided out-of-the-box, you can create your own by creating a script and then adding to the operators window.

- Select Incident Response → Automation → Scripts → New Automation.

- Type a meaningful name for the script, and click Save.

- To create a filter operator script, do the following:

- In the Tags field, add the **filter** tag.

If you want a custom transformer that operates on an entire array rather than on each individual item, you need to add the **entirelist** tag.

b. In the Arguments section, add the following arguments:

Argument	Description
left	Mark as mandatory. This argument defines the left-side value of the transformer operation. In this example, this is the value being checked if it falls within the range specified in the right-side value.
right	Mark as mandatory. This argument defines the right-side value of the transformer operation. In this example, this is the range to check if the left-side value is in.

c. Add the script syntax and save.

4. To create a transformer operator script do the following:

a. In the Tags field, add the **transformer** tag.

b. In the Arguments section, add the following arguments:

Argument	Description
value	Mark as mandatory. The value to transform. In this example, this is the UNIX epoch timestamp to convert to ISO format.

c. Add the script syntax and save.

5. Go to the filters and transformers window and select the operator.

#### 10.5.4.1 | Filter considerations, categories, and built-in filters

##### Abstract

Filters in playbook tasks are defined built-in according to categories.

You can use built-in filters to define your filter, they are grouped by category. Before defining a filter, consider the following.

##### Filter considerations

- Filters try to cast the transformed value and arguments to the appropriate type. The task fails if casting fails. For example, "a" Equals {"some": "object"} => Error
- If the filter's left-side value expects a single item but receives a list, the filter passes if at least one item meets the requirements. For example, ["a", "b", "c"] Equals "b" => true.
- If the filter's left-side value expects a list but receives a single item, it converts it to a list with a single item. For example, "a" Contains "a" => True.
- Some custom filters are implemented as scripts with the **filter** tag. You can find examples in the playbook automation task description.
- Filters in conditional tasks do not iterate the items of the root. Instead, they fetch the left-side value and the right-side value and compare them.

##### Filter categories and built-in filters

When adding a filter, clicking the default Equals (String) field opens a search window showing the available built-in filters. They are defined by category as follows:

##### General

General filters such as Contains, Doesn't Contain, In, and Is empty.

Filter	Description
Contains	Tests whether the value on the left is contained in the value on the right. Can be used for any kind of object (not limited to a string).

Filter	Description
Doesn't Contain	Tests whether the value on the left is NOT contained in the value on the right. Can be used for any kind of object (not limited to a string).
Has length of	Tests whether a list specified on the left has the number of items specified on the right.
In	Tests whether the value on the left is contained in the object on the right.
Is defined	Tests whether a key on the left exists in context. <b>NOTE:</b> Is defined considers false and empty strings and lists to be defined values. If you don't want those to be included as defined, use Is not empty.
Is empty	Tests whether the value of a key is empty.
Is not empty	Tests whether the value of a key is NOT empty.
Not defined	Tests whether a key on the left does NOT exist in context. <b>NOTE:</b> Not defined considers false and empty strings and lists to be defined values. If you don't want those to be included as defined, use Is empty.
Not in	Tests whether the value on the left is NOT contained in the object on the right.

**String**

Determines the relationship between the left-side string value and the right-side string value, such as starts with, includes, and in the list. The string filter returns partial matches as True.

Filter	Description
Doesn't end with	Tests whether the string on the left is NOT the end of the string on the right.
Doesn't equal	Tests whether the strings are NOT the same.
Doesn't include	Tests whether the string on the right is NOT a substring of the string on the left.
Doesn't start with	Tests whether the string on the right is NOT the beginning of the string on the left.
Ends with	Tests whether the string on the left is the end of the string on the right.
Equals	Tests whether the strings are the same.
Has length	Tests whether the two strings have the same length.

Filter	Description
In list	Tests whether the string on the left is in the list on the right.
Includes	Tests whether the string on the right is a substring of the string on the left.
Matches - regex	Tests whether the string on the left matches the regex on the right. Uses Go-style regex.
Not in list	Tests whether the string on the left is NOT a substring of the string on the right.
Starts with	Tests whether the string on the right is the beginning of the string on the left.
StringContainsArray	Tests whether a substring or an array of substrings on the left is within a string array on the right. Supports single strings as well. For example, for substrings ['a', 'b', 'c'] in string 'a' the script returns true.

**Number**

Determines the relationship between the left-side number value and the right-side number value, such as Equals, Greater than, and Less than.

Filter	Description
Doesn't equal	Tests whether the number on the left does NOT equal the number on the right.
Equals	Tests whether the number on the left equals the number on the right.
Greater or equal	Tests whether the number on the left is greater than or equal to the number on the right.
Greater than	Tests whether the number on the left is greater than the number on the right.
InRange	Tests whether the number on the left is within a range specified on the right. For example, if the left value is 4, and the range on the right is 1,8, the condition is true.
Less or equal	Tests whether the number on the left is less than or equal to the number on the right.
Less than	Tests whether the number on the left is less than the number on the right.

**Date**

Determines whether the left-side time value is earlier than, later than, or the same time as the right-side time value.

Filter	Description
After	Tests whether the date on the left is after the date on the right.

Filter	Description
AfterRelativeDate	Tests whether the date on the left occurred after the provided relative time (such as '6 months ago') on the right. Returns True or False.
Before	Tests whether the date on the left is before the date on the right.
Same as	Tests whether the two dates are the same.

Supported time and date formats

Format	Example
ANSIC	Tues Jan _2 15:04:05 2019
UnixDate	Tues Jan _2 15:04:05 MST 2019
RubyDate	Tues Jan 02 15:04:05 -0700 2019
RFC822	02 Jan 19 15:04 MST
RFC822Z	02 Jan 19 15:04 -0700 // RFC822 with numeric zone
RFC850	Tuesday, 02-Jan-19 15:04:05 MST
RFC1123	Tues, 02 Jan 2019 15:04:05 MST
RFC1123Z	Tues, 02 Jan 2019 15:04:05 -0700 // RFC1123 with numeric zone
RFC3339	2019-01-02T15:04:05Z07:00
RFC3339Nano	2019-01-02T15:04:05.999999999Z07:00
Kitchen	3.04PM
Stamp	Jan _2 15:04:05
StampMilli	Jan _2 15:04:05.000
StampMicro	Jan _2 15:04:05.0000000
StampNano	Jan _2 15:04:05.000000000

Boolean

Determines whether a field is true or false, or the string representation is true or false.

Filter	Description
Is false	Tests whether the value on the left evaluates to false.
Is true	Tests whether the value on the left evaluates to true.

Other

Miscellaneous filters, including CheckIfSubdomain and IsInCidrRanges.

Filter	Description
CheckIfSubdomain	Tests whether the value on the left is a subdomain of the value on the right.
CIDRBiggerThanPrefix	Tests whether the CIDR prefix on the left is bigger than the defined maximum prefix on the right.
GreaterCidrNumAddresses	Tests whether the number of available addresses in IPv4 or IPv6 CIDR on the right is greater than the input given on the left.
IsInCidrRanges	Tests whether the IPv4 address on the left is contained in at least one of the comma-delimited CIDR ranges on the right. Multiple IPv4 addresses can be passed in a comma-delimited list and each address is tested.
IsNotInCidrRanges	Tests whether the IPv4 address on the left is NOT contained in at least one of the comma-delimited CIDR ranges on the right. Multiple IPv4 addresses can be passed in a comma-delimited list and each address is tested.
IsRFC1918Address	Tests whether an IPv4 address on the left is in the private RFC-1918 address space (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) on the right.
LowerCidrNumAddresses	Tests whether the number of available addresses in IPv4 or IPv6 CIDR on the right is less than the input given on the left.

#### 10.5.4.2 | Transformer considerations, categories, and built-in transformers

##### Abstract

Use transformers in playbook tasks according to the following considerations.

You can use built-in transformers to define your transformer, they are grouped by category. Before defining a transformer, consider the following.

##### Transformer considerations

- Transformers try to cast the transformed value (and arguments) to the necessary type. Tasks will fail if casting has failed, for example `{"some": "object"}` To upper case => `Error`.
- Some transformers are applied on each item of the result. For example, `a, b, c` To upper case => `A, B, C`.
- Some transformers operate on the entire list. For example, `a, b, c` count => `3`.
- Some custom transformers are implemented as scripts with the `transformer` tag. You can find examples in the playbook automation task description.

#### Transformer categories and built-in transformers

When adding a transformer, clicking the default To upper case (String) field opens a search window showing the available built-in transformers. They are defined by category as follows.

Transformer Category	Description	Built-In Transformers																					
General	Generic transformers	<p>General built-in transformers</p> <table border="1"> <thead> <tr> <th>Name</th><th>Description</th><th>Example</th></tr> </thead> <tbody> <tr> <td>Unique</td><td>Returns a de-duped version of a list.</td><td>a, b, a, c, d, a, b =&gt; a, b, c, d</td></tr> <tr> <td>Slice</td><td> <p>Returns part of a specified list in a range of <b>from</b> index (included) through <b>to</b> index (not included)</p> <p><b>from:</b> Zero based index at which to begin extraction (default: 0).</p> <p><b>to:</b> Zero based index before which to end extraction (default: list length).</p> </td><td>a, b, c, d from: 1, to: 3 =&gt; b, c</td></tr> <tr> <td>Slice by item</td><td> <p>Returns part of a list specified in a range of from item (included) through to item (not included).</p> <p><b>from:</b> Item from which to begin the extraction. If not specified, extracts from the beginning of the list.</p> <p><b>to:</b> Item before which to end the extraction. If not specified, extracts from the end of the list.</p> </td><td>a, b, c, d from: b, to: d =&gt; b, c</td></tr> <tr> <td>Sort</td><td> <p>Sorts an entire list. Supports strings and numbers.</p> <p><b>descending:</b> <b>true</b> to sort in descending order, default is false.</p> </td><td> b, c, a =&gt; a, b, c  <b>2.1, 1.2, 3.4</b> descending: <b>true</b>  =&gt; <b>3.4, 2.1, 1.2</b> </td></tr> <tr> <td>Get index</td><td> <p>Get item at the given index.</p> <p><b>index:</b> Index of the item to get.</p> </td><td> b, c, a index: 0 =&gt;b  b, c, a index -1 =&gt; nil </td></tr> <tr> <td>Splice</td><td> <p>Adds or removes items to/from an array.</p> <p><b>index:</b> (required) Zero-based index at which to begin add/remove items.</p> <p><b>deleteCount:</b> Number of elements to remove from 'index', default is 0.</p> <p><b>item:</b> Item to add to the array after 'index' position.</p> </td><td> a, b, c, d,index: 1 deleteCount: 2=&gt; a, d  a, b, c, d, index: 2 item: w  =&gt; a, b, c, w, d </td></tr> </tbody> </table>	Name	Description	Example	Unique	Returns a de-duped version of a list.	a, b, a, c, d, a, b => a, b, c, d	Slice	<p>Returns part of a specified list in a range of <b>from</b> index (included) through <b>to</b> index (not included)</p> <p><b>from:</b> Zero based index at which to begin extraction (default: 0).</p> <p><b>to:</b> Zero based index before which to end extraction (default: list length).</p>	a, b, c, d from: 1, to: 3 => b, c	Slice by item	<p>Returns part of a list specified in a range of from item (included) through to item (not included).</p> <p><b>from:</b> Item from which to begin the extraction. If not specified, extracts from the beginning of the list.</p> <p><b>to:</b> Item before which to end the extraction. If not specified, extracts from the end of the list.</p>	a, b, c, d from: b, to: d => b, c	Sort	<p>Sorts an entire list. Supports strings and numbers.</p> <p><b>descending:</b> <b>true</b> to sort in descending order, default is false.</p>	b, c, a => a, b, c <b>2.1, 1.2, 3.4</b> descending: <b>true</b> => <b>3.4, 2.1, 1.2</b>	Get index	<p>Get item at the given index.</p> <p><b>index:</b> Index of the item to get.</p>	b, c, a index: 0 =>b b, c, a index -1 => nil	Splice	<p>Adds or removes items to/from an array.</p> <p><b>index:</b> (required) Zero-based index at which to begin add/remove items.</p> <p><b>deleteCount:</b> Number of elements to remove from 'index', default is 0.</p> <p><b>item:</b> Item to add to the array after 'index' position.</p>	a, b, c, d,index: 1 deleteCount: 2=> a, d a, b, c, d, index: 2 item: w => a, b, c, w, d
Name	Description	Example																					
Unique	Returns a de-duped version of a list.	a, b, a, c, d, a, b => a, b, c, d																					
Slice	<p>Returns part of a specified list in a range of <b>from</b> index (included) through <b>to</b> index (not included)</p> <p><b>from:</b> Zero based index at which to begin extraction (default: 0).</p> <p><b>to:</b> Zero based index before which to end extraction (default: list length).</p>	a, b, c, d from: 1, to: 3 => b, c																					
Slice by item	<p>Returns part of a list specified in a range of from item (included) through to item (not included).</p> <p><b>from:</b> Item from which to begin the extraction. If not specified, extracts from the beginning of the list.</p> <p><b>to:</b> Item before which to end the extraction. If not specified, extracts from the end of the list.</p>	a, b, c, d from: b, to: d => b, c																					
Sort	<p>Sorts an entire list. Supports strings and numbers.</p> <p><b>descending:</b> <b>true</b> to sort in descending order, default is false.</p>	b, c, a => a, b, c <b>2.1, 1.2, 3.4</b> descending: <b>true</b> => <b>3.4, 2.1, 1.2</b>																					
Get index	<p>Get item at the given index.</p> <p><b>index:</b> Index of the item to get.</p>	b, c, a index: 0 =>b b, c, a index -1 => nil																					
Splice	<p>Adds or removes items to/from an array.</p> <p><b>index:</b> (required) Zero-based index at which to begin add/remove items.</p> <p><b>deleteCount:</b> Number of elements to remove from 'index', default is 0.</p> <p><b>item:</b> Item to add to the array after 'index' position.</p>	a, b, c, d,index: 1 deleteCount: 2=> a, d a, b, c, d, index: 2 item: w => a, b, c, w, d																					

Transformer Category	Description	Built-In Transformers		
		Name	Description	Example
		Index of	Returns the first index of the element in the array, or -1 if not found.  <b>item:</b> Item to locate in the array.  <b>fromLast:</b> true to get the index from last. (default is false).	<code>a, b, a, c, d, a, b, item: b =&gt; 1</code>  <code>a, b, a, c, d, a, b, item: a fromLast: true =&gt; 5</code>  <code>a, b, a, c, d, a, b, item: w =&gt; -1</code>
		Get field	Extracts a given field from the given object.  <b>field:</b> (required) The field to extract from the result	<code>{"name": "john", "color": "white"} field: "color" "white"</code>
		Stringify	Converts the given item to a string.	<code>{ "name": "john", "color": "white" } =&gt; '{"name": "john", "color": "white"}'</code>
		Count	Returns the number of elements.	<code>b, c, a =&gt; 3</code>  <code>null =&gt; 0</code>  <code>a =&gt; 1</code>
		Join	Concatenates all elements.  <b>separator:</b> Specifies a string to separate each pair of adjacent elements of the array, default is an empty string.	<code>b, c, a separator: , =&gt; b,c,a</code>  <code>b, c, a =&gt; bca</code>

Transformer Category	Description	Built-In Transformers																					
String	<p>String transformers</p> <p><b>NOTE:</b> To make regex case non-sensitive, use the <code>(?i)</code> prefix (for example <code>(?i)yourRegexText</code>).</p>	<p>String built-in transformers</p> <table border="1"> <thead> <tr> <th>Name</th><th>Description</th><th>Example</th></tr> </thead> <tbody> <tr> <td>replace match</td><td> <p>Returns a string with some or all matches of a regex pattern, and replaces with a specified string.</p> <p>regex: A regex pattern to be replaced by the replaceWith argument.</p> <p>replaceWith: The string that replaces the string specified in the toReplace argument, default is an empty string. Detailed RegEx syntax can be found at <a href="https://github.com/google/re2/wiki/Syntax">https://github.com/google/re2/wiki/Syntax</a>.</p> </td><td> <pre>pluto,is,not,a,planet regex: ,," replaceWith: ;; =&gt;"pluto;is;not;a;planet"</pre> <p>"pluto is not a planet" regex .*to replaceWith vega =&gt; vega is not a planet</p> </td></tr> <tr> <td>Substring</td><td> <p>Returns a subset of a string between one index and another, or through the end of the string.</p> <p>from (required): An integer between 0 and the length of the string, specifying the offset into the string of the first character to include in the returned substring.</p> <p>to (optional): An integer between 0 and the length of the string, which specifies the offset into the string of the first character not to include in the returned substring.</p> </td><td> <pre>pluto is not a planet from: 4 to: 10 =&gt; o is n"</pre> </td></tr> <tr> <td>Split</td><td> <p>Splits a string into an array of strings, using a specified delimiter string to determine where to make each split.</p> <p>delimiter: Specifies the string which denotes the points at which each split should occur, default delimiter is <code>,</code>.</p> </td><td> <pre>hello world,bye bye world =&gt; hello world, bye bye world  hello world delimiter =&gt; hello, world</pre> </td></tr> <tr> <td>Split &amp; trim</td><td> <p>Splits a string into an array of strings and removes whitespace from both ends of the string, using a specified delimiter string to determine where to make each split.</p> <p>Arguments: delimiter: Specifies the string which denotes the points at which each split should occur (default delimiter is <code>,</code>).</p> </td><td> <pre>hello &amp; world delimiter: &amp; =&gt; hello, world</pre> </td></tr> <tr> <td>From string</td><td> <p>Returns a subset of a string from the first from string occurrence.</p> <p>from (required): String to substring from.</p> </td><td> <pre>pluto is not a planet from: pluto is =&gt; not a planet</pre> </td></tr> <tr> <td>To string</td><td> <p>Returns a subset of a string until the first to string occurrence.</p> <p>to (required): String to substring until.</p> </td><td> <pre>pluto is not a planet to: a planet =&gt; pluto is not</pre> </td></tr> </tbody> </table>	Name	Description	Example	replace match	<p>Returns a string with some or all matches of a regex pattern, and replaces with a specified string.</p> <p>regex: A regex pattern to be replaced by the replaceWith argument.</p> <p>replaceWith: The string that replaces the string specified in the toReplace argument, default is an empty string. Detailed RegEx syntax can be found at <a href="https://github.com/google/re2/wiki/Syntax">https://github.com/google/re2/wiki/Syntax</a>.</p>	<pre>pluto,is,not,a,planet regex: ,," replaceWith: ;; =&gt;"pluto;is;not;a;planet"</pre> <p>"pluto is not a planet" regex .*to replaceWith vega =&gt; vega is not a planet</p>	Substring	<p>Returns a subset of a string between one index and another, or through the end of the string.</p> <p>from (required): An integer between 0 and the length of the string, specifying the offset into the string of the first character to include in the returned substring.</p> <p>to (optional): An integer between 0 and the length of the string, which specifies the offset into the string of the first character not to include in the returned substring.</p>	<pre>pluto is not a planet from: 4 to: 10 =&gt; o is n"</pre>	Split	<p>Splits a string into an array of strings, using a specified delimiter string to determine where to make each split.</p> <p>delimiter: Specifies the string which denotes the points at which each split should occur, default delimiter is <code>,</code>.</p>	<pre>hello world,bye bye world =&gt; hello world, bye bye world  hello world delimiter =&gt; hello, world</pre>	Split & trim	<p>Splits a string into an array of strings and removes whitespace from both ends of the string, using a specified delimiter string to determine where to make each split.</p> <p>Arguments: delimiter: Specifies the string which denotes the points at which each split should occur (default delimiter is <code>,</code>).</p>	<pre>hello &amp; world delimiter: &amp; =&gt; hello, world</pre>	From string	<p>Returns a subset of a string from the first from string occurrence.</p> <p>from (required): String to substring from.</p>	<pre>pluto is not a planet from: pluto is =&gt; not a planet</pre>	To string	<p>Returns a subset of a string until the first to string occurrence.</p> <p>to (required): String to substring until.</p>	<pre>pluto is not a planet to: a planet =&gt; pluto is not</pre>
Name	Description	Example																					
replace match	<p>Returns a string with some or all matches of a regex pattern, and replaces with a specified string.</p> <p>regex: A regex pattern to be replaced by the replaceWith argument.</p> <p>replaceWith: The string that replaces the string specified in the toReplace argument, default is an empty string. Detailed RegEx syntax can be found at <a href="https://github.com/google/re2/wiki/Syntax">https://github.com/google/re2/wiki/Syntax</a>.</p>	<pre>pluto,is,not,a,planet regex: ,," replaceWith: ;; =&gt;"pluto;is;not;a;planet"</pre> <p>"pluto is not a planet" regex .*to replaceWith vega =&gt; vega is not a planet</p>																					
Substring	<p>Returns a subset of a string between one index and another, or through the end of the string.</p> <p>from (required): An integer between 0 and the length of the string, specifying the offset into the string of the first character to include in the returned substring.</p> <p>to (optional): An integer between 0 and the length of the string, which specifies the offset into the string of the first character not to include in the returned substring.</p>	<pre>pluto is not a planet from: 4 to: 10 =&gt; o is n"</pre>																					
Split	<p>Splits a string into an array of strings, using a specified delimiter string to determine where to make each split.</p> <p>delimiter: Specifies the string which denotes the points at which each split should occur, default delimiter is <code>,</code>.</p>	<pre>hello world,bye bye world =&gt; hello world, bye bye world  hello world delimiter =&gt; hello, world</pre>																					
Split & trim	<p>Splits a string into an array of strings and removes whitespace from both ends of the string, using a specified delimiter string to determine where to make each split.</p> <p>Arguments: delimiter: Specifies the string which denotes the points at which each split should occur (default delimiter is <code>,</code>).</p>	<pre>hello &amp; world delimiter: &amp; =&gt; hello, world</pre>																					
From string	<p>Returns a subset of a string from the first from string occurrence.</p> <p>from (required): String to substring from.</p>	<pre>pluto is not a planet from: pluto is =&gt; not a planet</pre>																					
To string	<p>Returns a subset of a string until the first to string occurrence.</p> <p>to (required): String to substring until.</p>	<pre>pluto is not a planet to: a planet =&gt; pluto is not</pre>																					

Transformer Category	Description	Built-In Transformers		
		Name	Description	Example
		concat	Returns a string concatenated with given prefix and suffix.  prefix: A prefix to concat to the start of the argument.  suffix: A suffix to concat to the end of the argument.	night prefix good => good night  night suffix shift=> night shift
Number	Number transformers	Number built-in transformers		
		Name	Description	Example
		Floor	Returns the highest integer less than or equal to the number.	1.2=> 1
		Ceil	Returns the lowest integer greater than or equal to the number.	1.2 =>2
		Round	Returns the nearest integer, rounding half way from zero.	7.68 => 8  2.43 => 2  2.5 => 3
		Absolute	Returns the absolute value of the given number.	-2 => 2
		Decimal precision	Truncates the number of digits after the decimal point, according to the by argument.  by: Number of digits to keep after the decimal point, default is 0.	8.6666 by: 2 => 8.66
		Modulus (remainder)	The modular operator (%) returns the division remainder.  by (required): Modulo by, default:0	20 by: 3=> 2
		To percent	Converts a number to a percent.  withsign: Specify true to include %. Default is false	0.22 => 20  0.22 withsign: true =>20%
		Quadratic equation	Returns the result of the Quadratic Formula.b (required): The b number of: ax <sup>2</sup> + bx + c = 0, default is 0.c (required): The c number of: ax <sup>2</sup> + bx + c = 0, default is 0.	1 b: 3 c: 2=> -1.00, -2.00  3 b: 2 c: 4=> (-0.333 +1.106i), (-0.333 -1.106i)

Transformer Category	Description	Built-In Transformers																
Date	Date transformers	<p>Date built-in transformers</p> <table border="1"> <thead> <tr> <th>Name</th><th>Description</th><th>Example</th></tr> </thead> <tbody> <tr> <td>Date to string</td><td> <p>Converts any date to a specified string format. The date input must be in ISO format. For example, <code>2021-10-06T13:44:07</code>. The default output format is RFC822.</p> <p><b>format:</b> The desired string output format. For example, if you want to convert to RFC822 format, enter <code>02 Jan 06 15:04 MST</code>.</p> <p>The following are available output format options:</p> <ul style="list-style-type: none"> <li>Layout = <code>01/02 03:04:05PM '06 -0700</code> // The reference time, in numerical order</li> <li>RFC3339Nano = <code>2006-01-02T15:04:05.99999999Z07:00</code></li> <li>Kitchen = <code>3:04PM</code> // Handy time stamps</li> <li>Stamp = <code>Jan _2 15:04:05</code></li> <li>StampMilli = <code>Jan _2 15:04:05.000</code></li> <li>StampMicro = <code>Jan _2 15:04:05.000000</code></li> <li>StampNano = <code>Jan _2 15:04:05.000000000</code></li> </ul> <p>This transformer is in GO language.</p> </td><td><code>2021-10-06T13:44:07 =&gt; 06 Oct 21 13:44 EDT</code></td></tr> <tr> <td>Date to Unix</td><td>Converts any date to Unix format.</td><td><code>Mon, 02 Jan 2006 15:04:05 MST =&gt; 1136214245</code></td></tr> </tbody> </table>	Name	Description	Example	Date to string	<p>Converts any date to a specified string format. The date input must be in ISO format. For example, <code>2021-10-06T13:44:07</code>. The default output format is RFC822.</p> <p><b>format:</b> The desired string output format. For example, if you want to convert to RFC822 format, enter <code>02 Jan 06 15:04 MST</code>.</p> <p>The following are available output format options:</p> <ul style="list-style-type: none"> <li>Layout = <code>01/02 03:04:05PM '06 -0700</code> // The reference time, in numerical order</li> <li>RFC3339Nano = <code>2006-01-02T15:04:05.99999999Z07:00</code></li> <li>Kitchen = <code>3:04PM</code> // Handy time stamps</li> <li>Stamp = <code>Jan _2 15:04:05</code></li> <li>StampMilli = <code>Jan _2 15:04:05.000</code></li> <li>StampMicro = <code>Jan _2 15:04:05.000000</code></li> <li>StampNano = <code>Jan _2 15:04:05.000000000</code></li> </ul> <p>This transformer is in GO language.</p>	<code>2021-10-06T13:44:07 =&gt; 06 Oct 21 13:44 EDT</code>	Date to Unix	Converts any date to Unix format.	<code>Mon, 02 Jan 2006 15:04:05 MST =&gt; 1136214245</code>							
Name	Description	Example																
Date to string	<p>Converts any date to a specified string format. The date input must be in ISO format. For example, <code>2021-10-06T13:44:07</code>. The default output format is RFC822.</p> <p><b>format:</b> The desired string output format. For example, if you want to convert to RFC822 format, enter <code>02 Jan 06 15:04 MST</code>.</p> <p>The following are available output format options:</p> <ul style="list-style-type: none"> <li>Layout = <code>01/02 03:04:05PM '06 -0700</code> // The reference time, in numerical order</li> <li>RFC3339Nano = <code>2006-01-02T15:04:05.99999999Z07:00</code></li> <li>Kitchen = <code>3:04PM</code> // Handy time stamps</li> <li>Stamp = <code>Jan _2 15:04:05</code></li> <li>StampMilli = <code>Jan _2 15:04:05.000</code></li> <li>StampMicro = <code>Jan _2 15:04:05.000000</code></li> <li>StampNano = <code>Jan _2 15:04:05.000000000</code></li> </ul> <p>This transformer is in GO language.</p>	<code>2021-10-06T13:44:07 =&gt; 06 Oct 21 13:44 EDT</code>																
Date to Unix	Converts any date to Unix format.	<code>Mon, 02 Jan 2006 15:04:05 MST =&gt; 1136214245</code>																
Supported time and date formats																		
		<table border="1"> <thead> <tr> <th>Format</th><th>Example</th></tr> </thead> <tbody> <tr> <td>ANSIC</td><td>Tues Jan _2 15:04:05 2019</td></tr> <tr> <td>UnixDate</td><td>Tues Jan _2 15:04:05 MST 2019</td></tr> <tr> <td>RubyDate</td><td>Tues Jan 02 15:04:05 -0700 2019</td></tr> <tr> <td>RFC822</td><td>02 Jan 19 15:04 MST</td></tr> <tr> <td>RFC822Z</td><td>02 Jan 19 15:04 -0700 // RFC822 with numeric zone</td></tr> <tr> <td>RFC850</td><td>Tuesday, 02-Jan-19 15:04:05 MST</td></tr> <tr> <td>RFC1123</td><td>Tues, 02 Jan 2019 15:04:05 MST</td></tr> </tbody> </table>	Format	Example	ANSIC	Tues Jan _2 15:04:05 2019	UnixDate	Tues Jan _2 15:04:05 MST 2019	RubyDate	Tues Jan 02 15:04:05 -0700 2019	RFC822	02 Jan 19 15:04 MST	RFC822Z	02 Jan 19 15:04 -0700 // RFC822 with numeric zone	RFC850	Tuesday, 02-Jan-19 15:04:05 MST	RFC1123	Tues, 02 Jan 2019 15:04:05 MST
Format	Example																	
ANSIC	Tues Jan _2 15:04:05 2019																	
UnixDate	Tues Jan _2 15:04:05 MST 2019																	
RubyDate	Tues Jan 02 15:04:05 -0700 2019																	
RFC822	02 Jan 19 15:04 MST																	
RFC822Z	02 Jan 19 15:04 -0700 // RFC822 with numeric zone																	
RFC850	Tuesday, 02-Jan-19 15:04:05 MST																	
RFC1123	Tues, 02 Jan 2019 15:04:05 MST																	

Transformer Category	Description	Built-In Transformers	
		Format	Example
	RFC1123Z	Tues, 02 Jan 2019 15:04:05 -0700 // RFC1123 with numeric zone	
	RFC3339	2019-01-02T15:04:05Z07:00	
	RFC3339Nano	2019-01-02T15:04:05.999999999Z07:00	
	Kitchen	3.04PM	
	Stamp	Jan _2 15:04:05	
	StampMilli	Jan _2 15:04:05.000	
	StampMicro	Jan _2 15:04:05.000000	
	StampNano	Jan _2 15:04:05.000000000	

## 10.5.5 | Extract indicators

### Abstract

Extract indicators from Cortex XSOAR incident fields and enrich them with commands and scripts defined for the indicator type.

In Cortex XSOAR, the indicator extraction feature extracts indicators from incident fields and enriches them using commands and scripts defined for the indicator type. If indicator extraction is enabled, indicators are extracted according to the incident type. For more information about indicator extraction, see [Indicator extraction](#).

How to set up indicator extraction in a playbook task

1. Select the playbook where you want to add indicator extraction, and click Edit.
2. In the playbook, click a task to open the task details pane.
3. Click the Advanced tab.
4. For Indicator Extraction mode, select the mode you want to use (default is inline).
5. Click OK.

Example 13.

The following scenario shows how indicator extraction is used in the Process Email - Generic v2 playbook to extract and enrich a very specific group of indicators.

This playbook parses the headers in the original email used in a phishing attack. It is important to parse the original email used in the phishing attack and not the email that was forwarded to ensure that you only extract the email headers from the malicious email and not the one your organization uses to report phishing attacks.

1. Navigate to the Playbooks page and search for the Process Email - Generic v2 playbook.
2. Click either Duplicate Playbook or Detach Playbook.
3. Open the Add original email details to context task, and for the Script drop down, change the script from Set to ParseEmailFilesV2.

Under the Outputs tab, you can see all of the different data that the task extracts.

The screenshot shows the 'Task Details' window for a 'ParseEmailFilesV2' task. The 'Outputs' tab is selected. The task has one output named '#2 Add original email details to context'. The 'Script' dropdown shows 'ParseEmailFilesV2'. Below the script dropdown are several input fields: To, CC, From, Subject, HTML, Text, Depth, Headers, HeadersMap, AttachmentNames, Format, and Email.HeadersMap. The 'From' field under Email.HeadersMap is highlighted with a red box.

4. Click the Advanced tab and set Indicator Extraction mode to **Inline**. This ensures all the outputs are processed before the playbook moves ahead to the next task.
5. Open the Display email information in layout - Email.Headers task. This task receives the data from the saved attachment tasks and sets the various data points to context.
6. Click the Advanced tab and set Indicator Extraction mode to **None**, because the indicators were already extracted earlier in the Extract email artifacts and attachments task and there is no need to extract them again.

#### Indicator extraction modes

Indicator extraction supports the following modes:

- **None:** Indicators are not extracted automatically. Use this option when you do not want to further evaluate the indicators.
- **Inline:** Indicators are extracted within the context that indicator extraction runs (synchronously). The findings are added to the context data. For example, if indicator extraction for the phishing alert type is inline:
  - For incident creation, the playbook you define to run by default does not run until the indicators have been extracted.
  - For an on-field change, extraction occurs before the next playbook tasks run. This option provides the most robust information available per indicator.

**NOTE:**

This configuration may delay playbook execution (incident creation).

While indicator creation is asynchronous, indicator extraction and enrichment are run synchronously. Data is placed into the incident context and is available via the context for subsequent tasks.

- **Out of band:** Indicators are extracted in parallel (asynchronously) to other actions. The extracted data will be available within the incident, however, it is not available for immediate use in task inputs or outputs because the information is not available in real-time.

For incident creation, out of band is used in rare cases where you do not need the indicators extracted for the proceeding flow of the playbook. You still want to extract them and save them in the system as indicators, so that they can be reviewed at a later stage for manual review. System performance may be better as the playbook flow does not stop extracting, but if the alert contains indicators that are needed or expected in the proceeding playbook execution flow, inline should be used, as it will not execute the playbook before all indicators are extracted from the alert.

**NOTE:**

When using Out of band, the extracted indicators do not appear in the context. If you want the extracted indicators to appear select Inline.

- Indicators are extracted according to the following rules:

- Incident creation - inline
- Incident field change - inline
- Tasks - none, can be overridden on a per task basis
- CLI - out of band, but can be overridden on a per-command basis

#### Troubleshoot indicator extraction

If indicators are not extracted, check whether the indicator mode is set to none. Even if you select the relevant incident fields and the indicators to extract, if the mode is set to none, indicators do not extract.

### 10.5.6 | Extend context

#### Abstract

Extend context to retrieve specific information from integrations or commands and map to fields.

By design, integrations do not write all of the data returned from a command to the context. This prevents large context size and enables you to store only the most relevant information.

The Extend Context feature enables you to save additional data from the raw response of the command. For example, when a command runs to retrieve events from a SIEM, only some of the event fields are written to context, according to the integration design. With Extend Context, you can save additional fields specific to your use case.

Extend Context can also be used when the same command runs multiple times in the same playbook, but the outputs need to be saved to different context keys. For example, you can execute the `!ad-get-user` command twice, once to retrieve the user's information and again to retrieve the user's manager's information. By default, an integration command writes the data from the same command to the same context key. By using Extend Context, you can write the command's response to a custom context key of your choice.

You can extend context either in a playbook task or directly from the command line. Whichever method you use, first run your command with the `raw-response=true` flag. This helps you identify the information that you want to add to your extended data.

#### Filter for specific keys from lists of dictionaries

You can use DT to get select keys of interest from a command that returns a list of dictionaries containing many keys. For example, the `findIndicators` automation returns a long list of indicator properties, but you may only be interested in saving the value and the `indicator_type` to minimize the size of the context data. For more information about DT, see Cortex XSOAR Transform Language (DT).

Example 14.

1. Run the command `!findIndicators size=2 query="type:IP" raw-response=true`.

You will see a list of two dictionaries containing 20+ items.

2. Use the following value for extend-context to save only value and indicator\_type into a context key called FoundIndicators:

```
!findIndicators size=2 query="type:IP" extend-context='FoundIndicators=.={"value": val.value, "indicator_type": val.indicator_type}'
```

3. Use the following value for extend-context to save only the incident name, status, and id to a key called FoundIncidents:

```
!SearchIncidentsV2 id=<ANY INCIDENT_ID> extend-context='FoundIncidents=Contents.data={"name": val.name, "status": val.status, "id": val.id}' ignore-outputs=true
```

#### Extend context in a playbook task

1. Go to the **Advanced** tab of the relevant playbook task, such as a Data Collection task.
2. In the Extend Context field, enter the name of the field in which you want the information to appear and the value you want to return. For example, using the `!ad-get-user` command, enter `name="john" attributes=displayName` to place the user's name in the `displayName` key.

The following image shows the result of the `!IPReputation ip=20.8.1.5 raw-response=true` command.

The screenshot shows a JSON viewer interface with the title "#INCIDENT-7898 -". At the top, there is a message from "DBot" dated "Feb 14th 2022 13:26:40" with the command "IPReputation ip='20.8.1.5' raw-response='true'". Below the message is a search bar labeled "Search in JSON entries". The main area displays a hierarchical JSON structure:

- root:** 3 items
  - bucketInfo:** 7 items
    - dailyBucketStart: 2022-02-13 18:57:20
    - dailyPoints: 100000
    - dailyPointsRemaining: 82629
    - minuteBucketStart: 2022-02-14 11:25:48
    - minutePoints: 200
    - minutePointsRemaining: 159
    - waitInSeconds: 0
  - indicator:** 8 items
    - firstSeenTsGlobal: null
    - indicatorType: IPV4\_ADDRESS
    - indicatorValue: 20.8.1.5
    - lastSeenTsGlobal: null
      - latestPanVerdicts:** 1 item
        - PAN\_DB: UNKNOWN
      - seenByDataSourceIds:** 0 items
        - summaryGenerationTs: 1644838000403
      - wildfireRelatedSampleVerdictCounts:** 0 items
    - tags:** 0 items

To include more than one field, separate the fields with a double colon. For example: `attributes=displayName::manager=attributes.manager`

3. To output only the values for Extend context and ignore the standard output for the command, select the Ignore Outputs checkbox.

While this will improve performance, only the values that you request in the Extend Context field are returned. You cannot use Field Mapping as there is no output to which to map the fields.

#### Extend context using the CLI

1. Run your command with the extend-context flag `!<commandName><argumentName> <value>extend-context=contextKey=JsonOutputPath`.

For example, to add the user and manager fields to context use the ad-get-user command, as follows:

```
!ad-get-user=${user.manager.username} extend-context=manager=attributes.manager::attributes=displayName
```

2. To output only the values that you set as Extend context, run the command with the ignore-output flag=true. `!ad-get-user=${user.manager.username} extend-context=manager=attributes.manager::attributes=displayName ignore-output=true`

#### Example 15. Extend context using the CLI with the IBM Qradar v3 integration instance

By default, after adding an IBM Qradar v3 integration instance, incidents pulled from QRadar to Cortex XSOAR return multiple fields, including `event_count`, `device_count`, `offense_type`, `description`. You can use extend context to show which additional information is available. You can also use that information to map it to a field in Cortex XSOAR.

- Run the command `!qradar-offenses-list raw-response="true"`. From the context data, you should see that multiple fields are returned.
- Identify the fields that you want to view and run your command. For example, to retrieve the number of devices affected by a given incident, as well as the domain in which those devices reside, run the following command:

```
!qradar-offences-list extend-context=device-count=device_count::domain_id=domain_id
```

## 10.5.7 | Set and update incident fields

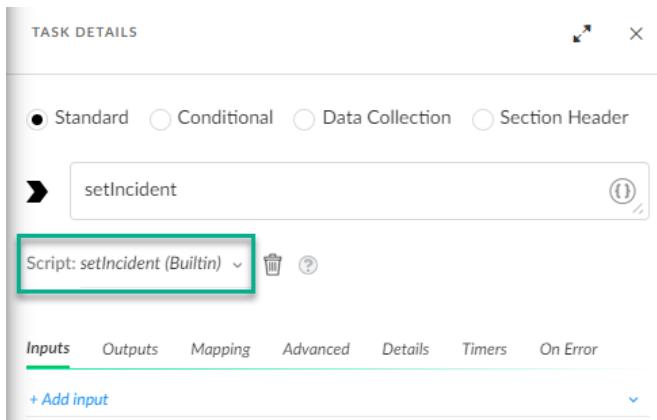
### Abstract

Use the `setIncident` script to set and update all system incident fields.

Using a playbook to create incident fields offers a structured and automated approach to defining and populating fields with relevant data during incident handling. This ensures consistency in data collection, enhances the organization of incident information, and facilitates streamlined analysis and response processes.

Creating incident fields is essential for structuring and storing specific information related to security incidents. These fields enable efficient organization and retrieval of incident data, enhancing analysis, decision-making, and automated response actions. It is an iterative process in which you create fields as you better understand your needs and the information available in the third-party integrations you use. You initially define incident fields after the planning stage, with mapping and classification for how the incidents will be ingested from third-party integrations into Cortex XSOAR.

During the investigation, you can then use the `setIncident` script in a playbook task to set and update incident fields.



### NOTE:

- The `setIncident` script includes all available input fields. Click + Add input and use the scroll bar to see all the fields.
- The name field has a limit of 600 characters. If there are more than 600 characters, you can shorten the name field to under 600 characters and then include the full information in a long text field such as the description field.
- There are many fields already available as part of the Common Type content pack. Before creating a new incident field, check if there is an existing field that matches your needs.

For more information on creating custom incident types and fields, see this video.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

## 10.5.8 | Playbook polling

### Abstract

Generic Polling playbook enables you to periodically poll the status of a process on a remote host.

When working with third-party products (such as detonation, scan, search, and other third-party products) you may need to wait for a process to finish on the remote host before continuing. In these cases, the playbook should stop and wait for the process to complete on the third-party product, and continue when it is done. Integrations or automations may not be able to do this due to hardware limitations.

Generally, polling is used in the following scenarios:

- File detonation in a sandbox
- URL detonation
- Queries that take a long time to complete

To use polling, Cortex XSOAR comes out-of-the-box with the GenericPolling playbook, which periodically polls the status of a process being executed on a remote host, and when the host returns that the process execution is done, the playbook finishes execution. For more information about using this playbook, see Generic Polling.

The GenericPolling playbook is used as a sub-playbook to block the execution of the main playbook until the remote action is complete. There are a number of playbooks that use the GenericPolling playbook that come out-of-the-box or installed from a content pack, such as:

- Context Polling - Generic: Polls a context key to check if a specific value exists.
- Field Polling - Generic: Polls a field to check if a specific value exists.
- QRadarFullSearch: Runs a QRadar query and return its results to the context.
- Scan Assets - Nexpose: Scans according to asset IP addresses or host names from Rapid7 Nexpose, and waits for the scan to finish by polling the scan status in pre-defined intervals.
- Detonate File - JoeSecurity: Detonates one or more files using the Joe Security integration.

See this video for more information on how to use generic polling in Cortex XSOAR.

#### PREREQUISITE:

You need to use the GenericPolling playbook as a sub-playbook in a main playbook, such as Detonate File - JoeSecurity.

The main playbook should follow this structure:

1. **Start Command:** The task contains a command that fetches the initial state of the process and saves it to context. This command starts the process that should be polled. For example:

Detonation: Submits a sample for analysis (detonated as part of the analysis), using the `joe-analysis-submit-sample` command.

Scan: Starts a scan for specified asset IP addresses and host names using the `nexpose-start-assets-scan` command.

Search: Searches in QRadar using AQL using the `qradar-searches` command.

2. **Polling Command:** The task contains the GenericPolling sub-playbook that polls for an answer. For example:

- Detonation: After the file is submitted to Joe Security, the playbook polls for specific information for analysis, such as ID, status, comments, errors, SHA-256 hash details.
- Scan: After the scan runs in Nexpose, using the playbook polls for scan information such as the scan type, the number of assets found, the scan ID, and other information.
- Search: The playbook runs the `qradar-get-search` to poll for the search ID and status.

3. **Results Task:** Returns the results of the operation. The task contains the results that were polled, which are added to context. For example, after polling JoeSecurity, the results are added to context.

For information about the GenericPolling playbook inputs such as `Ids`, `Interval`, and `dt`, see Playbook inputs.

Example 16.

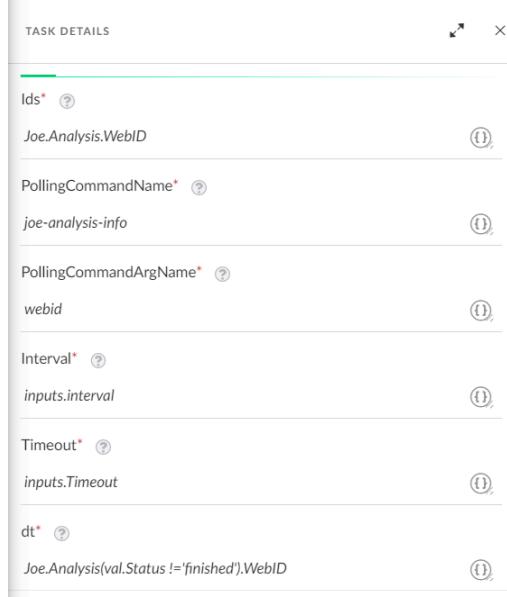
This generic polling example uses the Detonate File - JoeSecurity playbook from the Joe Security content pack.

The Detonate File - JoeSecurity playbook detonates one or more files using the Joe Security integration and returns relevant reports to the War Room and file reputations to the context data.

1. If you have not done so, go to Marketplace and download the Joe Security content pack.
2. Go to Playbooks and search for Detonate File - JoeSecurity.
3. Open the JoeSecurity Upload File task. This task uses the `joe-analysis-submit-sample` command, which starts a new analysis of a file in Joe Security. This is the **Start** command.
4. Open the GenericPolling task. This is the **Polling** command.

- **Ids:** Returns a list of **Joe.Analysis.ID**'s to poll.
  - **PollingCommandName:** The **joe-analysis-info** command returns information for a specified analysis, such as status, MD5, SHA256, vendor.
  - **PollingCommandArgName:** The **webid** argument name of the polling command.
  - **dt:** The filter for polling. This is defined as **Joe.Analysis(val.Status != 'finished').ID**.
- Joe.Analysis:** The object to return.  
`(val.Status != 'finished').ID` Gets the object that has a status other than 'finished', and then gets its ID field. The polling is done only when the result is **finished**. When finished, the **dt** filter returns an empty result, which triggers the playbook to stop running.

You can change the **Status** to: **starting**, **running**, or **finished**.

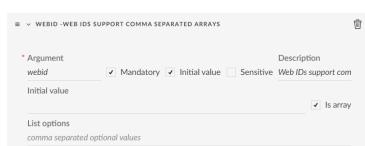


5. Open the JoeSecurity Get Info task. The **joe-analysis-info** command returns details of the IDs that have finished polling. This is the **Results** task.
6. Open the Set Context task. The context path to store the poll results is **Joe.Analysis**.

#### GenericPolling playbook limitations

The GenericPolling playbook has the following limitations.

- Global context is not supported.
- Global context outputs enable receiving information from multiple integrated products when executing playbooks and commands.
- It does not run from the Playground.
  - It uses the ScheduleGenericPolling script, which must support a list argument.



#### Troubleshoot playbook polling

The following are common generic polling issues and the recommended ways to deal with them.

- The playbook is “stuck” on **Waiting for polling to complete**.

As generic polling schedules tasks are outside the context of the playbook (not visible in the playbook run), errors may appear only in the War Room. Go to the War Room for the incident and check for errors or warnings related to GenericPolling tasks.

- The GenericPolling task completes but the status has still not “finished”.

If the timeout is reached, the playbook successfully finishes even if there are items that did not complete. Try increasing the timeout value for the GenericPolling task.

- The integration returns an ID not found error when running from the GenericPolling sub-playbook, but when running manually, it finishes successfully.

Some products cannot handle consecutive requests to query an action status right after the request to perform the action. After you initiate the action, try adding a Sleep task before calling the GenericPolling sub-playbook.

## 10.6 | Scripts

### Abstract

Create and edit a script including detaching and attaching and automation settings.

Scripts perform specific automated actions using commands that are used in playbook tasks and in the War Room.

On the Scripts page, you can view, edit, and create scripts in JavaScript, Python, or PowerShell. When creating a script, you can access all Cortex XSOAR APIs, including access to alerts, investigations, share data to the War Room. Scripts can receive and access arguments and can be password protected.

### Configure existing scripts

When you developing a script, consider editing an out-of-the-box script to leverage existing functionality and save time and effort. On the Scripts page, use free text in the search box to find an existing script. You can search using part or all of the scripts' names or tags. You can also search for an exact match of the script name by putting quotation marks around the search text. For example, searching for **"AddEvidence"** returns the script with that name. You can search for more than one exact match by including the logical operator **"or"** in-between your search texts in quotation marks. For example, searching for **"AddEvidence" or "AddKeyToList"** returns the two scripts with those names. Wildcards are not supported in free text search.

The Script Helper provides a list of available alphabetically ordered commands and scripts.

Start by exploring the Common Scripts.

### Common Scripts

Cortex XSOAR comes out-of-the-box with several common scripts that can be used in playbooks and commands (from the War Room), the majority of which are contained in the Base and Common Scripts content packs.

The Base content pack is a core pack that helps you get started and includes scripts that can be used in other JavaScript, Python, and PowerShell scripts. The Common Scripts content pack includes scripts that are commonly used, such as EmailReputation, RunDockerCommand, and ConvertXMLToJson.

Common Scripts contain code (such as functions and variables) that can be used across scripts and can be embedded when writing your scripts and integrations. Common Scripts are reusable modules or functions that provide additional functionality and capabilities to interact with APIs. Instead of duplicating code across multiple scripts or integrations, developers can create common scripts containing commonly used API interactions, such as authentication, data retrieval, or data manipulation. For example, in the **CommonServer** script, the **tableToMarkdown** function takes a JSON and transforms it into markdown. You can call this function from integrations and scripts that you author.

On the Scripts page, you can view/edit common scripts such as:

- CommonServer

The CommonServer script contains JavaScript functions and variables that can be used when writing your scripts and integrations.

The script contains nearly 200 functions/variables, such as `tabletoMarkdown`, `closeInvestigation`, and `SetSeverity`.

You can copy the script and add new functions/variables or add your functions to the CommonUserServer script. You can also use your scripts to override the existing scripts in the CommonServer script.

- CommonServerPython

The CommonServerPython script contains Python functions that can be used when writing your scripts and integrations.

The script contains over 400 functions, such as `appendContext`, `vtCountPositives` (which counts the number of detected URLs in the War Room entry), and `datetime_to_string`, (which converts a DateTime object into a string).

You can copy the script and add new functions/variables or add your functions to the CommonServerUserPython script. You can also use your scripts to override the existing scripts in the CommonServerPython script.

- CommonServerPowerShell

The CommonServerPowerShell script contains PowerShell arguments/functions that can be used when writing your scripts and integrations.

The script contains many arguments/functions, such as `SetIntegrationContext`, `Write-HostToLog` (which writes to the demisto.log), and `ReturnOutputs` (which returns results to the user more intuitively).

You can copy the script and add new arguments/functions or add your own to the CommonServerUserPowerShell script. You can also use your scripts to override the existing scripts in the CommonServerPowerShell script.

#### 10.6.1 | Create a script

##### Abstract

Create or edit an out-of-the-box script, including detach and attach and automation settings.

Developing scripts in Cortex XSOAR helps to automate repetitive tasks, streamline security operations, and make incident response more efficient. Customizing scripts can improve threat detection, mitigation, and remediation processes specific to your organization's needs.

Rather than creating a script from scratch, you can edit existing scripts. If the script was installed from a content pack, by default, the script is attached, which means that it is not editable. To edit the script, you need to either make a copy or detach it. While the script is detached, it is not updated by the content pack. This may be useful when you want to update the script without breaking customization. If you want to update the script through content pack updates, you need to reattach it, but any changes are overridden by the content pack on upgrade. If you want to keep the changes, make a copy before reattaching.

**NOTE:**

You can enable/disable a script in the Settings, without having to detach or duplicate the script.

1. Select Scripts → New Script.
2. Add an identifying name for the script.
3. Save the script.

##### Basic Script Settings

Define the relevant Basic script parameters.

Parameter	Description
Name	An identifying name for the script.
Language type	Select the script language type.
Description	A meaningful description of the script.

Parameter	Description
Tags	<p>Predefined script identifiers.</p> <p>For example, if a script is intended for phishing, tagging it with the phishing tag helps organize, classify, and manage the script among other scripts.</p> <p>Organizations can also implement policies or restrictions based on tags associated with scripts. For example, they may restrict certain users from accessing or executing a script tagged for phishing.</p>
Enabled	Whether the script is available for playbook tasks and indicator types, or to run in the CLI.

#### Special Script tags

Special script tags enable you to use the script in a specific area of Cortex XSOAR. For example, a script can be tagged for use in post-processing, indicator formatting, field display, and indicator enhancement. The following table includes the commonly used tags:

Tag Value	Description
Condition	<p>Conditional script in a playbook task</p> <p><b>NOTE:</b></p> <p>All custom scripts are available for conditional tasks, including scripts without the condition tag. System scripts are only available for use in conditional tasks if they have the condition tag.</p>
dynamic-indicator-section	General purpose dynamic section script for indicator layout
dynamic-section	General purpose dynamic section script for incident layout
enhancement	Indicator enhancement script
field-change-triggered	Script to run on incident field change
field-display	Field display script
filter	Script used as filter or conditional operator in a playbook task
general-dynamic-section	General purpose dynamic section script for object layouts (excluding incidents and indicator layouts)
incident-action-button	Incident layout action button script
indicator-action-button	Indicator layout action button script
indicator-format	Indicator formatting script
post-processing	Incident post-processing script
preProcessing	Pre-process rule script

Tag Value	Description
reputation	Indicator reputation script
sla	SLA breach script
threatIntelReport-action-button	Threat Intel Report layout action button script
transformer	Script used as transformer in a playbook task
widget	Script that can be used to generate a dashboard/report widget

#### Arguments

You can create, edit, or delete arguments as required.

Parameter	Description
Argument	An identifying name.
Mandatory	Makes the argument mandatory.
Default	Makes the argument the default.
Sensitive	Makes the argument case-sensitive.
Description	A meaningful description of the argument.
Default	The default value for the argument.
Is array	Specifies that the argument is an array.
Type	Select Unknown (default), Key-Value, or Text Area.
List options	A comma-separated list of argument values.

You can create, edit, or delete outputs as required. Define the outputs according to types such as string, number, date, and boolean. For more information, see Context and Outputs.

Parameter	Description
Context Path	A dot-notation representation of the path to access the Context. For example, <code>ThreatStream.Analysis.ReportID</code> .

Parameter	Description
Description	A short description of what the context path represents. For example, the ID of the report submitted to the sandbox.
Type	The value type of the context path, such as string, number, and date Enables Cortex XSOAR to format the data correctly.

#### Script Permissions

Define the script permissions to set who can view and execute the script.

Parameter	Description
Password Protect	Enables you to add a password for the script, which will be required when running the script from the CLI.
Run as	Permissions for scripts in Cortex XSOAR are determined by the Run as and Role fields.  Run as defines the specific role the script is executed with. Role determines who can manually execute the script.  By default, most scripts in Cortex XSOAR are run as a limited user with restricted access. However, some scripts require higher permissions and are delivered out-of-the-box with Run as set to DBotRole, which means the script executes with elevated permissions.  When a user runs a script manually, the Run as permissions are added to the permissions of the user running the script. For example, if user runs a script with Run as set to elevated permissions, its results can be viewed even by users with lower permissions. To mitigate this, you should assign Run as aligned with the users to whom you want to expose the information the script can extract.  When a script runs automatically, for example from a playbook running on fetched incidents, a display script, a trigger script or a pre-processing rule, the Run as permissions define who can view the playbook results.

**NOTE:**

Content packs typically use scripts, and the scripts can have dependencies on each other, so it is important to assign the Run as and Role parameters consistently to ensure scripts run properly.

If you change permissions for a script, the new permissions do not affect playbooks that are already using the script. The playbooks continue using the previous permissions until the next run, even when the playbook is triggered manually.

#### Example 17. Script Permissions

For a script that searches for all incidents in the system, the following scenarios show how the system behaves given the various Run as and Role configurations.

If you implement the following roles on your incidents:

- Incident 1: Viewable to Administrators
- Incident 2: No role assigned
- Incident 3: Viewable by Analysts
- Incident 4: Viewable by nested Analyst (custom role)

Scenario	How The System Behaves	What A Nested Analyst (Custom Role) Sees For Each Incident When Running The Script To Search For Incidents
Run as set to DBotRole Role not set	All users can execute the script, whether they are an analyst or administrator, and they have access to all incidents that are returned.	Incidents 1 - 4
Run as set to DBotRole Role set to Analyst	Any user with at least analyst permissions will have access to execute the script. All other users are not able to implement the script in playbook tasks. They cannot run the script manually from the command line or in a playbook.  All users can see the results of the execution in the War Room.	Nested Analyst cannot execute the script, but if a playbook runs the script, nested Analyst sees incidents 1 through 4.
Run as set to Analyst Role not set	All users can execute the script, but only incidents that are viewable by the analyst role and by the executing user's role are returned.	Incidents 2 - 4
Run as set to Analyst Role set to Analyst	Only users with at least the analyst role are able to execute the script. Only incidents that are viewable by the analyst role and by the executing user's role are returned.	Nested Analyst cannot execute the script, but if a playbook runs the script, nested Analyst sees incidents 2 and 4.

## Advanced

Parameter	Description
Timeout (seconds)	Time (in seconds) before the script times out. Default is 180.
Docker image name	For Python scripts, this is the name of the Docker image to use to run the script.  Cortex XSOAR supports the following Python versions: <ul style="list-style-type: none"><li>• 2.7</li><li>• 3.0 and later</li></ul> You can change the Docker image.  The default Docker image that Cortex XSOAR uses is demisto/python3, but you can use other Docker images from a private image registry. See Change the Docker image in an integration or script for more information.

Parameter	Description
Run on a separate container	Runs the script on a separate container.

#### Depends On Commands

You can set the commands that the script depends on directly from these settings. You still have the option to set the dependencies in the script YAML file.

#### Edit existing code or create new code

Modify parameters, logic, or integrations within a script to adapt it to specific use cases, optimize performance, and address evolving security needs without starting from scratch.

The Script Helper provides a list of available alphabetically ordered commands and scripts.

Example 18.

See this video for an example using Python code to develop a script.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

## 10.7 | Debug your playbook

### Abstract

Set breakpoints, conditional breakpoints, skip tasks, and input and output overrides in the playbook debugger.

The debugger provides a test environment where you can make changes to data and playbook logic and view the results in real-time to test and troubleshoot playbooks. You can see exactly what is written to the context at each step and which indicators are extracted.

To open a detached system playbook, a copy of a system playbook, or a custom playbook in the debugger, select the playbook and click Edit.

To open an attached playbook in the debugger, select the playbook and click View to access the debugger. While editing a playbook, sub-playbooks can be opened directly in the debugger by choosing Open sub-playbook in the task pane.

In some cases, you may have a playbook that includes two or more copies of the same sub-playbook. When you set breakpoints, override inputs or outputs, or skip tasks in sub-playbook A, the same changes apply to the identical sub-playbook B. In addition, if you set a breakpoint, override inputs or outputs, or skip tasks within a loop in a playbook, that setting will be applied every time the loop executes.

#### NOTE:

The debugger runs with the permissions of the logged in user. The user must have permissions for both playbooks and investigations (View/Edit) to run the debugger.

Running the debugger involves the following actions.

#### Choose test data

The debugger uses test data to execute the playbook, so you can see what your expected results would be. The following are options for test data.

- New Mock Incident:** By default, the debugger runs using an empty mock incident. An empty mock incident is useful to test simple functionality, such as a playbook that does simple tasks such as parsing inputs.
- Playground:** You can load the contents of the Playground as test data, enabling you to use uploaded files and custom context data for testing purposes.
- Existing Incident:** You can select an existing incident. For example, when debugging a phishing playbook, you might want to use an existing phishing incident that came from the mail listener integration. Using an existing incident in the debugger does not change the original incident.

If you need to use event data from third-party software that is not yet set up as an integration, you can import a JSON file into Cortex XSOAR through the mapping feature and create an incident that can then be used as test data.

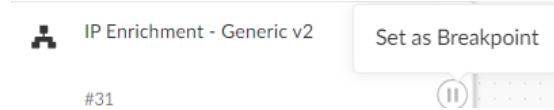
You can use a file attachment for your test data by adding the file to an incident and selecting the incident or by uploading the file to the playground and using the playground as test data.

### Set a breakpoint

At the breakpoint, you can override inputs and outputs to see how changes affect playbook execution. In addition, conditional breakpoints set conditions for the playbook to proceed. The playbook only pauses if your condition is met, letting you manipulate data to see how different scenarios impact how the playbook runs. For example, you can set a conditional breakpoint to pause the playbook when a phishing incident targets a member of a VIP asset list. If there are no VIPs in this incident, the execution does not pause. If there is a VIP in the incident, you can check that the member was properly identified by the playbook task.

Breakpoints do not apply to manual tasks, as a manual task will always pause the playbook run unless you skip the manual task. When the playbook reaches a breakpoint, no new tasks begin, but parallel tasks that have already begun continue. Breakpoints can be set in both the parent playbook and sub-playbooks.

1. To set a breakpoint, go to a task and click on the breakpoint button. When a breakpoint is set, the breakpoint button changes to orange.



2. After a breakpoint is reached, click the task to override inputs and outputs if needed.

3. When you are finished with the task, run the debugger, and in the task, select an option for the playbook to continue.

For an automated task, you have the options Run automation now or Complete Manually. If you choose Complete Manually, click on Mark Completed for the playbook to continue.

For a task that is a sub-playbook, click Run playbook now for the playbook to continue.

For a conditional task, choose which branch the playbook should follow and click Mark Completed for the playbook to continue. The default branch is else.

When the playbook reaches a breakpoint, the task has an orange line at the top to indicate the breakpoint.



Breakpoint alerts are also displayed at the top of the playbook, enabling you to navigate between multiple breakpoints that have been reached in the playbook or sub-playbooks.

### Set a conditional breakpoint

Conditional breakpoints enable you to debug loops and tasks with multiple values. The playbook only pauses if your condition is met, letting you manipulate data to see how different scenarios impact the playbook run.

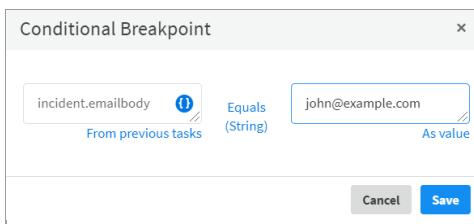
1. Click on the breakpoint button for a task.

After a breakpoint is set and the breakpoint icon is orange, a tooltip appears enabling you to add a condition to the breakpoint.



2. On both sides of the condition statement, you can choose available playbook data From previous tasks or use As value to set any other value.

Clicking on the curly brackets enables you to use data from the current playbook and from sub-playbooks.



3. Click on the Equals (String) to select from a set of conditions (such as: contains, ends with, greater than.)

#### NOTE:

If the breakpoint condition as defined does not exist when the debugger runs, the condition will default to false. For example, if you choose IP address and there is no IP address available, the playbook will not pause.

4. Click Save to save your conditional breakpoint.

### Start and stop the debugger

The debugger runs the playbook with the permissions of the logged in user. If a user runs potentially harmful commands, they are logged to the audit trail with the user's username. When the user sets breakpoints, skips tasks, or overrides inputs or outputs, those changes only apply to the individual user's session and do not permanently change the playbook. Using an existing incident as test data does not affect the original incident or change the original context data. When tasks run, however, they execute the same as they would without the debugger. For example, if you run the debugger and a task adds an item to a list, that item will be in the real list, accessible across for all users with permission to view that list.

Breakpoints pause playbook execution before a specific task. When the playbook is paused, the Debugger Panel displays the current state of context data, indicators, and task information.

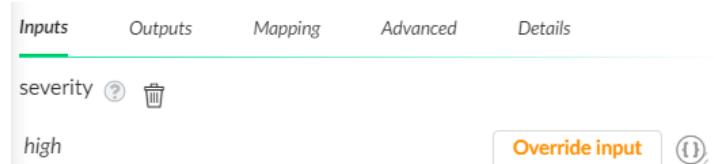
To start the debugger, click Run. When you click Stop, the debugger stops, and the context data is reset to the original incident data. In the case of a new mock incident, the context data is cleared and the context is empty. Any breakpoints, skips, or overrides you applied are still available.

### Override inputs and outputs

The debugger enables you to temporarily override inputs and outputs for a playbook run and to view the results in real time. When you override an input or output in the debugger, the change is saved only in the debugger view and only for the user who made the change. If after testing you decide to keep the temporary changes you made and apply them permanently to the playbook for all users, you need to cancel the override and edit the task. Tasks can be edited directly in the debugger or outside of the debugger using the standard playbook editing options.

You can override task inputs or outputs before or during a playbook run to troubleshoot tasks that fail or to try different input and outputs as part of playbook development. If you override an input or output during a playbook run, the override is applied to the run if the playbook has not yet reached that task. If you edit (permanently change) inputs during a playbook run, the changes only take effect the next time you run the playbook. You cannot use filters or transformers for overrides.

1. To override an input or output, open the task and hover over any existing input or output. Click Override Input.



2. Enter a new input or output that will be used only in the debugger. For output overrides, you can enter a value, an array of values, or JSON. For input overrides, you can only enter plain text.

3. Click OK to save your changes.

The playbook task card displays a label indicating that the task input or output has been overridden.

### Skip tasks

For testing purposes, you may want to skip a task that for example closes a port in a firewall, deletes an email, or sends a notification to a manager. Or you might skip a task where the integration has not yet been configured. By skipping a task and overriding the output, you can provide the data necessary to complete the playbook run. When you skip a conditional task, you can choose which branch runs after the skipped task, enabling you to test different outcomes for multiple branches.

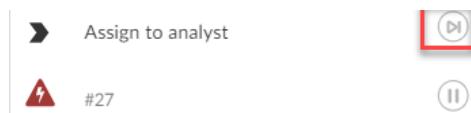
You might need to skip tasks within a playbook:

- To check if a particular task is causing an issue.
- To avoid performing tasks not relevant for your troubleshooting.
- To skip tasks with potentially harmful results such as blocking a user or opening a port in a firewall.
- To skip tasks for integrations that are not yet configured.

### How to skip a task

1. Click the 'skip' button for the task.

When a task is set to skip, the 'skip' button will be orange.

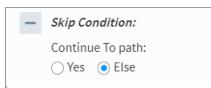


2. If the output is required for the playbook to proceed, click the task and override inputs and outputs.

## Skip conditional tasks

When you skip a conditional task, you can set which branch runs after the skipped task, enabling you to test different outcomes for multiple branches.

1. Choose skip for a conditional task. The skip button will turn orange.
2. Click on the task. Select which branch runs after the skipped task. If you do not choose, the else branch runs by default.



3. Click OK to save your changes.

## View context data, indicators, and task information

Within the debugger panel, you can view the context data during the playbook run as well as the indicators as they are extracted by clicking any completed task in the playbook while the debugger is running.

You can see the results of that task in the debugger panel.

## 10.7.1 | Troubleshoot playbook performance

### Abstract

Obtain playbook metadata to troubleshoot performance issues.

You can analyze playbook metadata such as tasks input and output, the amount of storage each task input/output uses, and the type of task. This is useful when troubleshooting your custom playbook if your system has slowed down and is using high CPU usage, memory, or storage (disk space).

After an incident has been assigned to a playbook you can analyze it to see its tasks inputs/outputs storage. You can filter the data according to the KB used in each task input/output.

- From the Incidents page, in the Incident War Room, run the following command in the CLI.

```
!getInvPlaybookMetaData incidentId=<incident ID> minSize=<size of the data you want to return in KB. Default is 10>
```

Example 19.

To view the playbook metadata that is used in incident number 964, in the CLI type `!getInvPlaybookMetaData incidentid="964" minSize="0"`.

## 10.8 | Manage playbook content

### Abstract

Manage playbook content by either using a remote repository, or by saving versions of your playbook.

### Manage playbook content with a remote repository

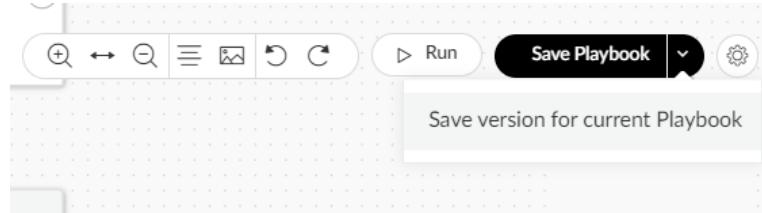
In Cortex XSOAR, you can develop and test your playbook content on development machines before using it in a production environment using the remote repository feature.

For more information about content management in Cortex XSOAR, see Content management in Cortex XSOAR.

#### Save versions of your playbook in Cortex XSOAR

You can save versions of a playbook as you are developing it. When you save a version of a playbook, add a meaningful comment so that you will be able to recognize the changes you made in that version at a later time. The version is saved with the name of the playbook, your commit message, an indication of what the change was (modify, insert), the date the playbook was saved, and the name of the author who last saved it. If necessary, you can access the playbook's version history and revert your playbook to a previous version.

1. In a playbook, after making changes, click the list next to Save Playbook and then click Save version for current Playbook.

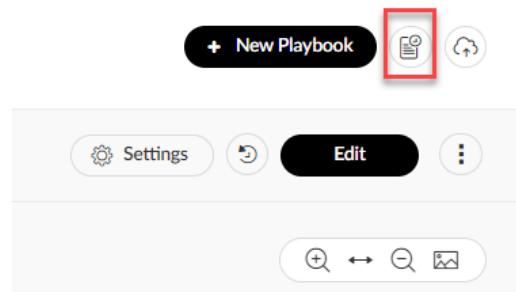


2. Enter a description of the change that was made to the current version.

3. Click Update Playbook.

4. To access a version of a playbook:
  - a. Click the icon next to New Playbook. The tooltip displays Version history for all Playbooks.

a. Click the icon next to New Playbook. The tooltip displays Version history for all Playbooks.



b. Search for the required playbook. The description that was entered when the version was saved should help you locate the version you now require.

c. Click Restore to restore the required version of the playbook.

## 10.9 | Add ad-hoc tasks to a Work Plan as part of your investigation

### Abstract

Add ad-hoc tasks to a Work Plan in Cortex XSOAR for a specific iteration of a playbook.

As part of your incident investigation, within the Work Plan you can create tasks for a specific iteration of a playbook. The task type can be an automation or another playbook. For example, within a manual task, you might need to enrich some data and run an investigation playbook.

When you create a task, add a name, automation, and description. The name and description should be meaningful so that the task corresponds to the data that you are collecting.

1. In the Work Plan, go to the task where you want to add and click the + sign at the bottom right-hand corner of the task.

The ad-hoc task is added after the task on which you clicked.

2. Select the task type.

- Standard: Runs a single automation.
- Playbook: Runs a playbook to enhance the investigation.

The playbook functions as any playbook would and requires you to define the inputs and outputs, as well as any other details.

3. Click Save.

4. To run the Work Plan again, click the Run again icon.

Example 20.

For a phishing investigation, after the initial playbook run parses the email and extracts email addresses, as part of the manual investigation, you could use the Email Address Enrichment - Generic v2.1 playbook as an ad-hoc playbook task to get more information about these email addresses.

## 10.10 | Best practices

### Abstract

Best practices for building and working with playbooks.

The following guidelines are best practices for building playbooks as well as optimizing playbook design and performance. Whether you are just starting or are creating advanced workflows, we recommend reviewing these recommendations carefully so your playbooks have a clear logical flow and run correctly and efficiently.

#### Best practices for building your playbook

##### Use the Use Case Builder to define your use case

The Use Case Builder content pack helps you streamline the use case design process, including building your playbook. It contains tools to help you measure and track use cases through your automation journey and quickly autogenerate OOTB playbooks and custom workflows.

For a detailed example of designing and building a use case, watch this video series.

##### Use clear task names and descriptions

Describe tasks clearly. Tasks should be clear to someone not familiar with the playbook workflow. This applies to task names, task descriptions, and the playbook description. When naming tasks, the guideline should be that users can understand what the playbook does by reading the task names, without having to open individual tasks to view the details.

Clear	Unclear
Task name: Check if the IP is Private	Task name: IP Check

##### Define playbook inputs and outputs properly

- **Group related input fields.**

Grouping inputs organizes the input fields and provides clarity and context to understand which inputs are relevant to which playbook flow.

- **Use camel case for input names.**

Use the CamelCase convention for inputs, keeping in mind that inherently capitalized terms should be kept in upper case. For example, the Entity ID input should be named EntityID and MITRE Technique should be MITRETechnique.

- **Define outputs properly.**

When configuring playbook outputs, configure sub-keys as much as possible, do not limit configuration to only the root keys. For example, instead of outputting File, output File.Name, File.Size, etc. This helps when viewing the outputs of the playbook within another playbook.

##### Configure playbook task inputs correctly

- **Avoid using Cortex XSOAR Transform Language (DT) in the Get input field definition.**

If you need to use DT for complex processing and you think a new filter or transformer would provide a better alternative to your DT solution, you can request the feature or contribute it. Consider using DT only if it can drastically simplify the playbook or improve performance.

##### Define playbook logic carefully

In each task, make sure appropriate logical operations are performed on input data. For example:

- **Avoid race conditions.**

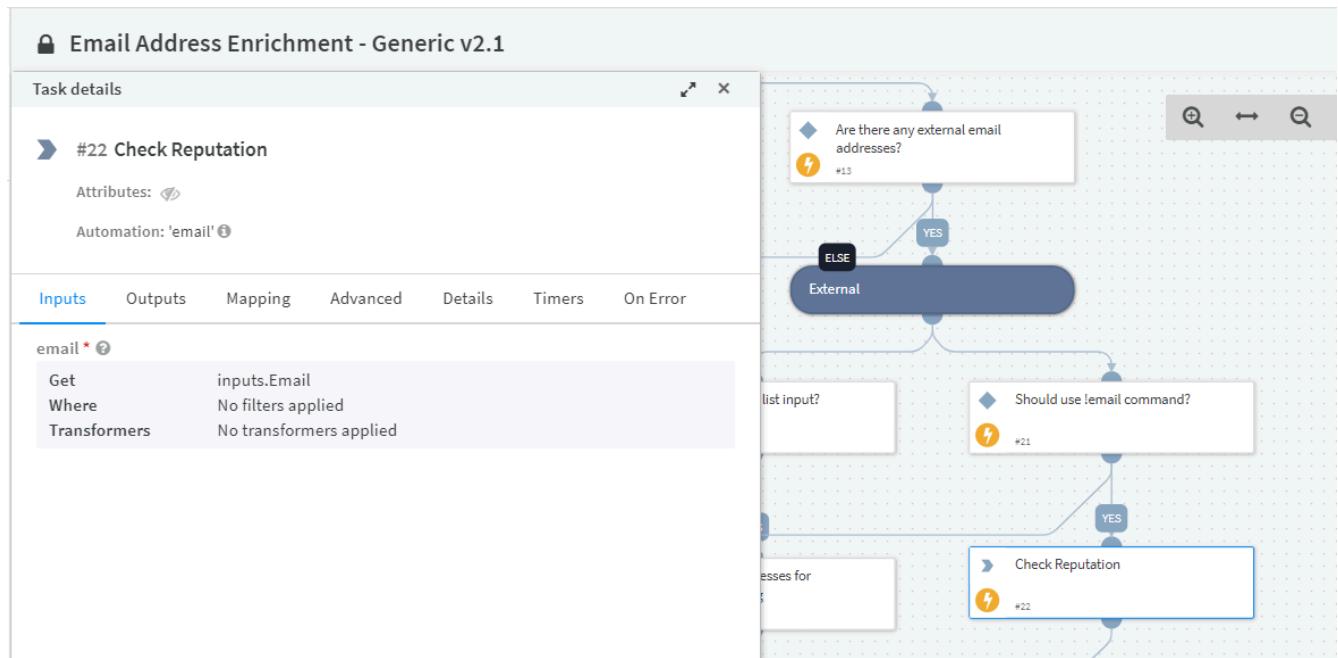
Be aware of potential race conditions. When you want to add multiple values to the same key, do not use multiple tasks that run Set, SetAndHandleEmpty, or any other script that sets data in context at the same time, because a race condition can cause your data to be overwritten by the same tasks. This is especially problematic when trying to append data. Instead, run the tasks one after the other or use scripts to append the data instead of setting a new value to the key.

- **Determine where inputs are coming from.**

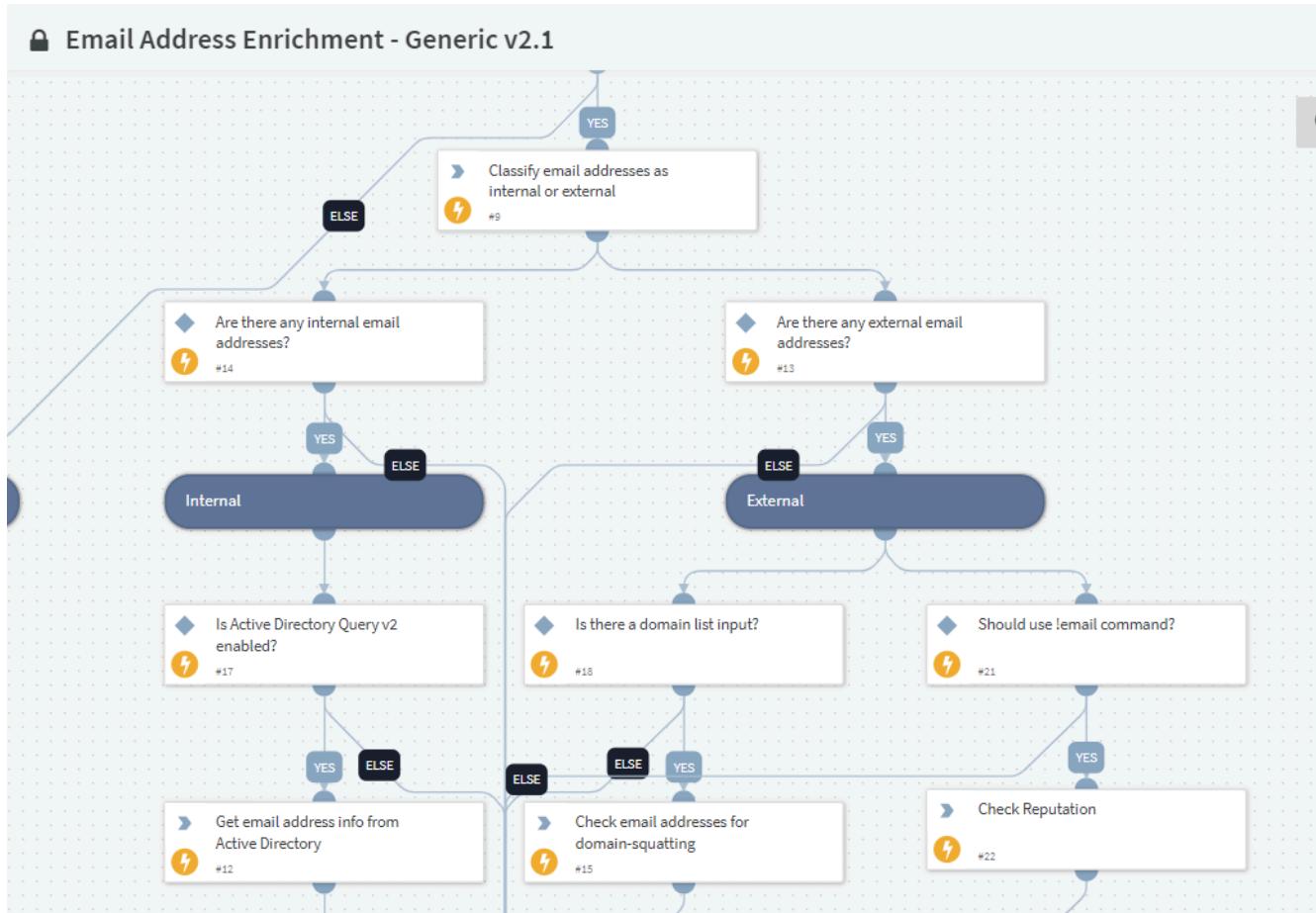
Verify whether the data you're getting is As value (simple value) or From Previous Tasks (from context).

- **Filter your inputs correctly so the task runs efficiently.**

Tasks take their inputs from the context, not directly from the previous tasks (even if it says from previous tasks). For an example of a task not receiving the right context, see this bug (since fixed) in a playbook:



The playbook begins by classifying the emails as internal or external. It then checks the reputation of external email addresses if any were found. That happens on the right side of the image. We expect that branch to run only if external addresses are found.



However, we did not apply a filter to the last task that gets the reputation on the right side:

This means that if both internal and external email addresses are found, we proceed with both branches (internal and external) of the playbook, and the task that gets the reputation runs without an applied filter, effectively taking all the emails we have in the inputs. The correct task input should have been:

email \* ?

Get	Account.Email.Address
Where	Account.Email.NetworkType Equals External
Transformers	Unique

- Select Ignore case for input names.

Use ignore-case option where possible, especially when checking Boolean playbook inputs such as True which users may end up configuring as true with a lowercase t:

Get IsolateEndpoints Equals (String) True

Search...

General	String	Number
Contains	Doesn't end with	Doesn't equal
Doesn't Contain	Doesn't equal	Equals
Has length of	Doesn't include	Greater or equal
In	Doesn't start with	Greater than
Is defined	Ends with	InRange
Is empty	Equals	Less or equal
Is not empty	Has length	Less than
Not defined	In list	

Ignore case

- When working with two lists, if you need multiple items from list A, which are also in list B, use the in filter instead of the equals or contains filters.

Correct Method	Incorrect Method
<p>Get the IP addresses that are in the list of inputs.</p>	<p>Get the IP addresses where the addresses contain the list. This is incorrect because they don't contain the list, they contain individual items from it.</p>

- Differentiate between checking if a specific element exists versus checking if an element equals something. This is a common mistake that can lead to tests working in some situations, but not all.

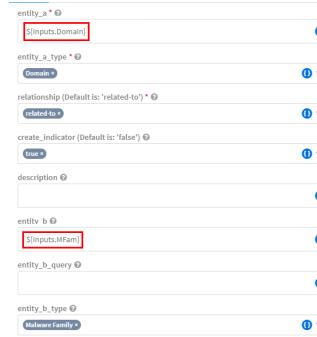
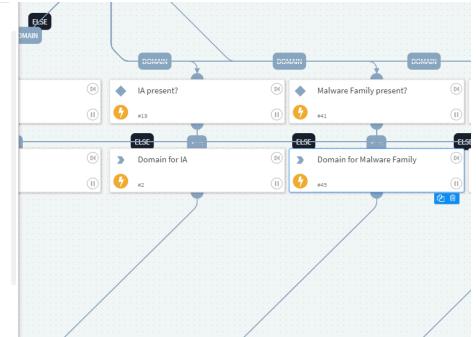
Correct Method	Incorrect Method
<p>Check if any object where the NetworkType is External exists.</p> <p>Condition for: yes</p>	<p>Check if the NetworkType of the IP object is External. This is incorrect because the IP object may contain multiple IPs, some internal and some external.</p>

- Run one or more tasks based on the object types versus running either one task or the other based on the type of one object.

Correct Method	Incorrect Method
<p>Check the existence of both object types and run tasks for the types found.</p>	<p>Check if there is either an internal or an external IP, and take only one path even if both types exist.</p>

Define playbook loops correctly

Use playbook loops only where needed. Loops are needed when certain actions have to be performed on specific pairs of data.

Correct Method Example	Incorrect Method Example
<p>Either use filters and transformers or loop through each separate indicator to verify they're creating the correct relationships.</p> 	<p>A user has a playbook that creates relationships for multiple indicator types. All indicator types and malware families are in their \${inputs.Domain} and \${inputs.MFam} playbook inputs.</p> <p>The user wrongly assumes that when creating the relationships, the correct malware families in \${inputs.MFam} correspond to the correct domains in \${inputs.Domain}.</p> 

Add a task to check that integrations are enabled

Use the `IsIntegrationEnabled` script in your playbook to make sure any integrations you need to run are enabled.

#### Best practices for optimizing playbook design and performance

In order to minimize your incident response time and make sure the system runs optimally, it's important to follow design and performance guidelines.

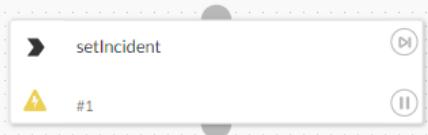
Use latest playbook and script versions

#### Playbooks

When returning to work on a playbook after a break, verify you're working on the latest version. Reattach the playbook if it's detached, and update it to ensure you're not editing an older version and introducing regressions. If you don't want to reattach your playbook, or you're still working on your custom version, we recommend reviewing the release notes to see what changes were made to the out-of-the-box playbook and copying those changes to your version.

#### Scripts

Update scripts and integration commands in playbook tasks to their most current version. Scripts that have updates or are deprecated are designated by a yellow triangle.



Break up large playbooks into sub-playbooks

If a playbook has more than thirty tasks, consider breaking the tasks into multiple sub-playbooks. Sub-playbooks can be reused, managed easily when upgrading, and they make it easier to follow the main playbook.

Playbooks that are triggered by an incident/job are considered a parent playbook. Sub-playbooks are playbooks that are used from within a parent playbook, as building blocks. The parent playbook is the main playbook that runs on the investigation, and each sub-playbook has a specific goal/responsibility.

- Parent playbooks usually have a `closeInvestigation` task at the end because they are the main playbook for that incident.
- Parent playbooks usually contain inputs that are passed down to sub-playbooks. Certain True/False flags may come from the parent playbook inputs.

Remove unused playbook tasks

For production playbooks, remove playbook tasks that are not connected to the playbook workflow.

Set the playbook to run in quiet mode

Run playbooks in quiet mode to reduce the incident size and execute playbooks faster. For playbooks running in jobs, indicator enrichment should be done in quiet mode.

Only extract indicators when needed

When indicator extraction is enabled for a playbook task, the task by default tries to extract all indicator types from the task Results. (The Results entry is the information printed to the War Room, not the outputs of the task). Extracting all indicator types can slow down the playbook, so it is important to only extract indicators as needed. For example, for the ParseEmailFilesV2 script which prints email information to the War Room, extraction should be enabled in order to extract email addresses, URLs, and other indicators. However, if your task runs the Sleep script, there is no point in extracting indicators.

Set the Indicator Extraction mode to None in the playbook task Advanced tab.

Minimize disk usage, CPU usage, and API calls

Consider the following:

- Do I need to do this action in multiple tasks?
- Can these tasks run in parallel instead of synchronously?
- Where applicable, am I setting realistic timeouts, search windows, intervals?
- Can I consolidate the API calls into one call? If not, can an integration enhancement solve this by accepting arrays as input instead of running multiple times for each input?
- Am I unnecessarily storing the same data twice? Do I have the data I need already stored?
- Where applicable, can I run this playbook without a loop?
- What extractions are running in my incident?
  - If your task requires extracted indicators, change the indicator extraction mode to inline. Use this mode carefully because it can affect performance. In addition, it is important to customize and limit the indicators extracted from incident fields of the incident type you are ingesting in the incident type settings Indicator Extraction Rules.
  - When creating new incident fields that do not need to be searched, double check whether they should be searchable under the relevant checkbox. Example of fields that should be searchable: Endpoint ID, Is Admin. Example of fields that should not be searchable: Additional Notes, Alert Summary.

## 11 | Lists

Abstract

Create and manage lists and add them to your playbook or script.

Create reusable data lists in Markdown, HTML, CSS, or JSON, and add them to your playbooks and scripts. Add data to your lists and leverage them across various automations for maximum efficiency.

### 11.1 | What is a list?

Abstract

Create and manage lists in Cortex XSOAR.

A list is a data container for storing data and is mainly used in playbooks and scripts, but can be accessed anywhere the context button appears (double-curly brackets). For example, in a playbook task, access the data in a list via the context button under Lists, or by using the path \${lists.<list\_name>}. Different types of data can be stored in a list, for example, text, string, numbers, Markdown, HTML, CSS, and JSON objects.

#### NOTE:

The maximum size of a list is 209715 characters.

#### Use cases

The following are use cases for lists:

- **Defining HTML templates:** An HTML template can be defined as part of a communication task.
- **Organizing Network Security:** Use lists to keep track of internal networks and IP addresses. Compare them to a set list to ensure only allowed connections get through.
- **Store Data Objects:** For example, a list of URLs, which you can call as an input for scripts and playbooks.
- **Prioritizing Incident Response:** Create lists to identify critical assets like important users or servers. This helps manage incidents better by focusing on the most important things first.

## 11.2 | Create a list

### Abstract

Create a list that can be accessed later such as in a playbook script or managed in the CLI.

To create a Text, Markdown, HTML, CSS, or JSON list type:

1. Go to Settings & Info Settings Advanced Lists Add a List.
2. Enter a name for the list.
3. From the list, select the Content Type.
4. Add content as required. For an example of a JSON list and how to use it, see Use cases: JSON lists.
5. To save, do one of the following:
  - Click Save.
  - Click Save Version to save your changes in Version history for all Lists. This allows you to revisit and restore previous versions.

### NOTE:

If you want to edit a list from a content pack, you need to duplicate or detach a list. Detached lists do not receive updated content in subsequent Cortex XSOAR content releases. To retain an updated list, reattach it.

## 11.3 | List commands

### Abstract

Use list commands in the CLI, playbooks, and scripts

Use the following list commands in the CLI, scripts, and playbook tasks:

Command	Description	Arguments
getList	Retrieves the contents of the specified list.	listName: The name of the list for which to retrieve the contents.
createList	Creates a list with the supplied data.	<ul style="list-style-type: none"> <li>• listName: The name of the list to which to add items.</li> <li>• listData: The data to add to the new list.</li> </ul>
addToList	Appends the supplied items to the specified list. If you add multiple items, make sure you use the same list separator that the list currently uses, for example, a comma or a semicolon.	<ul style="list-style-type: none"> <li>• listName: The name of the list to which to append items.</li> <li>• listData: The data to add to the specified list. The data will be appended to the existing data in the list.</li> </ul>
setList	Adds the supplied data to the specified list and overwrites existing list data.	<ul style="list-style-type: none"> <li>• listName: The name of the list to which to add items.</li> <li>• listData: The data to add to the specified list. The data overwrites the existing data in the list.</li> </ul>
removeFromList	Removes a single item from the specified list.	<ul style="list-style-type: none"> <li>• listName: The name of the list from which to remove an item.</li> <li>• listData: The item to remove from the specified list.</li> </ul>

### Example

In this example, a manageOOUsers script uses the `getList`, `createList`, and `setList` commands.

```

register_module_line('Manage000Users', 'start', __line__())

def _get_current_user():
    current_username = demisto.executeCommand("getUsers", {"current": True})
    if isError(current_username):
        demisto.debug(f"Failed to get current username - {get_error(current_username)}")
        return
    else:
        return current_username[0]["Contents"][0]['username']

def main():
    # get current time
    now = datetime.now()

    # args
    list_name = demisto.getArg("listname")
    username = demisto.getArg("username")

    option = demisto.getArg("option")
    days_off = now + timedelta(days=int(demisto.getArg("daysoff")))
    off_until = days_off.strftime("%Y-%m-%d")

    # update list name to start with '000', so we can't overwrite other lists with this
    if not list_name.startswith("000"):
        list_name = f"000 {list_name}"

    current_user = _get_current_user()
    if not current_user and not username:
        return_error('Failed to get current user. Please set the username argument in the script.')

    if not username:
        # Current user was found, running script on it.
        username = current_user
    else:
        # check if provided username is a valid xsoar user
        users = demisto.executeCommand("getUsers", {})
        if isrror(users):
            return_error(f'Failed to get users: {str(get_error(users))}')
        users = users[0]['Contents']

        users = [x['username'] for x in users]
        if username not in users:
            return_error(message=f'{username} is not a valid user')

    # get the out of office list, check if the list exists, if not create it:
    ooo_list = demisto.executeCommand("getList", {"listName": list_name})[0]["Contents"]
    if isError(ooo_list):
        return_error(f'Failed to get users out of office: {str(get_error(ooo_list))}')

    if "Item not found" in ooo_list:
        demisto.results(demisto.executeCommand("createList", {"listName": list_name, "listData": []}))
        ooo_list = demisto.executeCommand("getList", {"listName": list_name})[0]["Contents"]

    # check status of the list, and add/remove the user from it.
    if not ooo_list:
        list_data = []
    else:
        list_data = json.loads(ooo_list)
    if option == "add":
        # check if user is already in the list, and remove, to allow updating
        list_data = [i for i in list_data if not (i['user'] == username)]
        list_data.append({"user": username,
                          "offuntil": off_until,
                          "addedby": current_user if current_user else 'DBot'})
    else:
        # remove the user from the list.
        list_data = [i for i in list_data if not (i['user'] == username)]

    set_list_res = demisto.executeCommand("setList", {"listName": list_name, "listData": json.dumps(list_data)})
    if isError(set_list_res):
        return_error(f'Failed to update the list {list_name}: {str(get_error(set_list_res))}')

    # welcome back, or see ya later!
    if option == "add":
        demisto.results(f"Vacation mode engaged until {off_until}, enjoy the time off {username}")
    else:
        demisto.results(f"Welcome back {username}, it's like you never left!")

if __name__ in ('__builtin__', 'builtins', '__main__'):
    main()

register_module_line('Manage000Users', 'end', __line__())

```

## 11.4 | Use cases: JSON lists

### Abstract

Manage JSON lists in Cortex XSOAR that can be accessed by automations, playbooks, etc. List commands, lists arrays separators delimiters

List data can be stored in various structures, including JSON format. When accessing a valid JSON file from within a playbook, it is automatically parsed as a JSON object (list). Depending on how you store the data, you may need to Transform a List into an Array. For example, if using non-built-in commands in a script or you want to loop over list items, you should transform a list into an array. Working with a JSON file list in a playbook typically involves the following actions:

- Extract the data from a JSON object
- Extract a subset of the data
- Filter extracted data
- Apply transformers to extracted data. See [Filter and transform data](#) for more details.

#### Extract data from a JSON object

Create a JSON list and use the **Set** automation to create a new context key that can extract the data from the list.

##### 1. Create a List:

- a. In the Name field, type **Test1**.
- b. Select **Settings & Info** **Settings** **Advanced Lists** **Add a List**.
- c. In the Content Type field, select **JSON** and add the following content:

```
{
  "domain": {
    "name": "mwidomain",
    "prod_mode": "prod",
    "user": "weblogic",
    "admin": {
      "servername": "AdminServer",
      "listenport": "8001"
    },
    "machines": [
      {
        "refname": "Machine1",
        "name": "MWINODE01"
      },
      {
        "refname": "Machine2",
        "name": "MWINODE02"
      }
    ],
    "clusters": [
      {
        "refname": "Cluster1",
        "name": "App1Cluster",
        "machine": "Box1"
      },
      {
        "refname": "Cluster1",
        "name": "App2Cluster",
        "machine": "Box2"
      }
    ],
    "servers": [
      {
        "name": "ms1",
        "port": 9001,
        "machine": "Box1",
        "clusterrefname": "Cluster1"
      },
      {
        "name": "ms2",
        "port": 9002,
        "machine": "Box2",
        "clusterrefname": "Cluster2"
      }
    ]
  }
}
```

- d. Save the list.

##### 2. Create a playbook task with the Set automation:

- a. Select **Playbooks** → **New Playbook**.
- b. Name the playbook, and click **Save**.
- c. Click **Create Task** and provide a task name.
- d. In the **Choose Script** field, select **Set**.

The Set script sets a value in context under the key entered.

e. In the key field, define a context key name for the data. For example, JSONData.

The screenshot shows the configuration interface for a 'Set' action. At the top, there are four radio button options: 'Standard' (selected), 'Conditional', 'Data Collection', and 'Section Header'. Below this is a list item labeled '#1 Set' with a delete icon. Underneath is a script dropdown set to 'Set' with a trash icon and a help icon. A horizontal navigation bar includes 'Inputs' (underlined), 'Outputs', 'Mapping', 'Advanced', 'Details', and 'On Error'. The 'Inputs' tab is active. The 'key\*' field is filled with 'JSONData' and has a help icon. The 'value\*' field is empty and has a help icon.

f. In the value field, set the list you want to extract by clicking the curly brackets.

g. Click Filters And Transformers.

h. In the Get field, click the curly brackets, and in the Select source for value section, select the list you created in step 1: Test1.

i. In the Fetch data field, select an incident to test the data.

j. Click Test.

In this example, the test results have found the list data.

### Test result

```

root: {} 1 item
  domain: {} 7 items
    admin: {} 2 items
      listenport: 8001
      servername: AdminServer
    clusters: [] 2 items
      0: {} 3 items
        machine: Box1
        name: App1Cluster
        refname: Cluster1
      1: {} 3 items
        machine: Box2
        name: App2Cluster
        refname: Cluster1
  
```

k. When the test completes, click Save.

l. Save the task and playbook.

3. Check all the data is stored in the context key you defined by testing the playbook using the debugger:

a. Click Run.

b. Open the Debugger Panel.

The key you defined, JSONData, holds the data in context from the JSON object.

## DEBUGGER PANEL

**Context**      *Indicators*

Test data: New Mock Alert

Search in JSON context data...

**Alert**      Incident

```
1 ~   JSONData: {  
2 ~     domain: {  
3 ~       admin: {  
4         listenport: "8001"  
5         servername: "AdminServer"  
6       }  
7 ~       clusters: [  
8 ~         0: {  
9           machine: "Box1"  
10          name: "App1Cluster"  
11          refname: "Cluster1"  
12        }  
13 ~         1: {  
14           machine: "Box2"  
15           name: "App2Cluster"  
16           refname: "Cluster1"
```

Extract a subset of the data

In general, you can extract subsets of context data in a playbook to analyze a specific information set. This also applies to working with lists, for example extracting a subset of the data from a JSON object. In this example, we want to extract server information from the list created above.

1. In a playbook, create a task.

a. In the Choose Script field, select Set .

- b. In the key field, define a context key name for the data; for example, JSONDataSubset.
- c. In the value field, set the list you want to extract by clicking the curly brackets.
- d. Click Filters And Transformers.
- e. In the Get field, enter `lists.Test1.domain.servers`.
- f. In the Fetch data field, select an incident to test the data.
- g. Click Test.
- h. When the test completes, click Save.
- i. Save the task and the playbook.

2. Check that all the data is stored in the context key you defined by testing the playbook using the debugger.

- a. Click Run Debugger Panel.

- b. The key you defined (JSONDataSubset) holds the subset of the data in context from the JSON object.

The screenshot shows the 'Context' tab of the Run Debugger Panel. It displays a hierarchical JSON structure under 'Test data: New mock incident'. The structure includes a 'v0' node with 'clusterrefname: Cluster1', 'machine: Box1', 'name: ms1', and 'port: 9001'. A 'v1' node follows with 'clusterrefname: Cluster2', 'machine: Box2', 'name: ms2', and 'port: 9002'. A search bar at the top allows filtering of the JSON data.

```

{
  "v0": {
    "clusterrefname": "Cluster1",
    "machine": "Box1",
    "name": "ms1",
    "port": "9001"
  },
  "v1": {
    "clusterrefname": "Cluster2",
    "machine": "Box2",
    "name": "ms2",
    "port": "9002"
  }
}

```

#### Filter extracted data

You can filter the data subset you extracted and analyze this information on a more granular level. In this example, you want to filter Box1 information from the list created in Extract the data from a JSON Object above.

1. Re-open the task you created above.
2. Click the value field.
3. Under Filter, click Add Filter.
4. Set the condition you want to filter.

In this example, retrieve the list of machines named `Box1` from `Test1` list by setting the filter `lists.Test1.domain.servers.machine Equals Box1`.

The screenshot shows the 'FILTERS & TRANSFORMERS FOR value' section. Step 1, 'Get', contains the expression `lists.Test1.domain.servers`. Step 2, 'Filter', has a dropdown set to `lists.Test1.domain.servers` and a condition `File.Type is PDF`. Step 3, 'Apply transformers on the field (Optional)', contains the filter condition `lists.Test1.domain.servers.machine Equals Box1`.

#### 5. Click Test.

6. Check whether the data subset was accessed successfully by selecting the data source from an incident. You can see the results returned **machine: Box1**.

The screenshot shows the Cortex XSOAR interface. At the top, there's a header with 'TEST' on the left and a close button 'X' on the right. Below this is a list of extracted items:

- > **DBotScore**: 6 items
- > **File**: 3 items
- > **IP**: 5 items
- > **IPInfo**: 1 item
- > **VirusTotal**: 2 items
- > **alert**: 107 items

Below this list is a black 'Test' button. To the right of the list is a vertical scroll bar.

Underneath the list is a section titled 'Test result' with a collapsible tree view:

- ✓ **root**: 2 items
  - ✓ **0**: 4 items
    - clusterrefname: Cluster1
    - machine: Box1
    - name: ms1
    - port: 9001

At the bottom of this section is a 'Done testing' button.

#### Apply transformers to extracted data

In general, in a playbook task, you can transform (apply changes) to the data you extracted. This also applies to working with lists, for example, to transform extracted data from a JSON object. In this example, we extract the first element in the list and transform the data to upper case from the list created in Extract data from a JSON object above.

1. Re-open the task, click the contents of the value field, and keep the current filters.

2. In the Apply transformers on the field, click Add transformer.

3. Add the following transformers to the extracted data:

1. Add the **Get index (General)** transformer to extract a specific machine element.

Set **index: 0** to extract the first element from the list.

2. Add the **To upper case (String)** transformer.

The **To upper case (String)** transformer does not work on lists, only on individual elements. Therefore, the **Get index (General)** transformer should be applied before adding the **To upper case (String)** transformer.

## 1 Get

lists.Test1.domain.servers.machine 

## 2 Filter (Get subset of the data, e.g.

File.Type is PDF)

Where all of the following are true for  
lists.Test1.domain.servers

[lists.Test1.domain.servers.machine Equals Box1](#) 

[+ Add filter](#)

## 3 Apply transformers on the field (Optional)

[Get index \(index: 0\)](#) 

[To upper case](#) 

4. In the Fetch Data field, select an incident to test and click Test.

## 11.5 | Transform a list into an array

### Abstract

Create a transformer to split a list into an array when adding or editing a task in a playbook or when mapping an integration instance in Cortex XSOAR.

Create a transformer to split a list into an array, add or edit a task in a playbook, or map an instance.

1. Go to Playbooks and create or edit a playbook.
2. Select Create Task.
3. In the Choose script field, select the Set automation.
4. In the Key field, enter the key name.
5. In the value field, click {}.
6. Add a transformer.
  - a. Click Filters And Transformers.
  - b. In the Get field, click {}.
  - c. Expand the Lists node and select a list to transform.
  - d. In Apply transformers on the field, click Add transformer.
  - e. Search for and select Split.
  - f. (Optional) In the delimiter field, type the delimiter used to separate the items in the string (default is ",").
  - g. Click Save.
7. Save the task and playbook.

For an example of using a transformer in a list, see [Apply Transformers to Extracted Data](#).

## 12 | Jobs

### Abstract

Create a time-triggered job or event-triggered job to run a playbook

Schedule playbooks to run automatically by defining a job based on events or specific times. For instance, process indicators automatically upon ingestion and then add them to your SIEM.

## 12.1 | Manage jobs

### Abstract

Jobs run playbooks and are either time-triggered (run at specific times) or event triggered (run when there are changes to a feed).

A job is an automated playbook task or set of playbook tasks that are scheduled to run at predefined intervals or under specific conditions. Jobs can be used for data enrichment, periodic reporting, threat intelligence gathering, or any repetitive operational tasks that need to be performed regularly without manual intervention. There are two types of jobs:

- Time triggered jobs that run at specific times: For example, you can schedule a time triggered job that runs nightly and removes expired indicators.
- Jobs triggered by a delta or change in a feed: For example, you can define an event triggered job to run a playbook when a specified TIM feed finishes a fetch operation for new indicators.

On the Jobs page, you can:

Action	Details
Create a new job	Click + New Job.
Edit an existing job	In the table, select a job and click Edit.
Perform additional job management	<p>In the table, select a job and click one of the following:</p> <ul style="list-style-type: none"> <li>• Run now</li> <li>• Disable</li> <li>• Enable</li> <li>• Pause</li> <li>• Resume</li> <li>• Abort</li> <li>• Delete</li> </ul>
View job status	<p>The chart panel at the top of the Jobs page shows various status buttons. Click one of the following buttons to filter the list of jobs for that status:</p> <ul style="list-style-type: none"> <li>• Running</li> <li>• Waiting</li> <li>• Error</li> <li>• Disabled</li> <li>• Time Triggered</li> <li>• Event Triggered</li> </ul> <p>You can hide this panel by clicking Hide Chart Panel.</p>
Search for a specific job	Enter a search query in the filter field. You can also save a filter.

Action	Details
View job details in the table	<p>By default, the displayed table columns are:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Job Status</li> <li>• Last Incident's Status</li> <li>• Last Run</li> <li>• Next Run</li> <li>• Details</li> </ul> <p>Click  to change the displayed columns. You can also select to show:</p> <ul style="list-style-type: none"> <li>• Owner</li> <li>• Playbook</li> <li>• SLA</li> <li>• Labels</li> <li>• Attachments</li> <li>• Job Schedule: This column shows a human readable description of a cron schedule for a job.</li> </ul>

## 12.2 | Create a time triggered job

### Abstract

Create a time triggered or feed triggered job in Cortex XSOAR to run a playbook.

Time triggered jobs run at predetermined times. You can schedule the job to run at a recurring time or one time at a specific date and time. For an example, see the Create jobs to process indicators example.

1. Select Jobs → New Job.
2. Select Time triggered.
3. If you want the job to repeat at regular intervals, select Recurring and select the desired interval.

You can choose to run the job every X number of days, on specific days of the week, at a specific time and also choose a start date and an expiration date.

You can configure the recurring job using a cron expression. To do so, after selecting the Recurring checkbox, click Switch to Cron view and enter the expression. For help defining the cron expression, click Show cron examples after switching to cron view.

#### NOTE:

To view a human readable description of a cron schedule for an existing job, click  and select Job Schedule from the available columns.

4. If you do not want the job to repeat, Select date and time for the job to run.
5. Add or create any relevant tags to use as a search parameter in the system.
6. In the BASIC INFORMATION, section, add relevant time triggered job parameters from the following:

Name	Description
Name	Enter a meaningful name for the job.

Name	Description
Owner	Assign an owner to the incident.
Role	Select the role who can access the incident.
Type	Determine the incident type created by this job.
Severity	Determine the severity of the incident that is created.
Playbook	Determine which playbook to run when this job is triggered.
Labels	Select the labels that are available in the incident type.
Phase	Select the phase of the investigation in which this incident is opened.
Details	Add details that should appear within the incident.
Attachments	Add attachments to the job.

7. Enter any relevant custom field parameters.

All fields that have the Add to all incident types checkbox selected appear in incident and indicator fields.

8. In the QUEUE HANDLING section, select one of the following response options to use if the job is triggered while a previous run of the job is active:

- Notify the owner
- Don't trigger a new job run
- Cancel the previous job run and trigger a new job run
- Trigger a new job run and execute concurrently with the previous run

**IMPORTANT:**

We recommend to avoid triggering a job while a previous run of the job is active by configuring the playbook a job triggers to close the investigation before running a new job.

9. Select Create new job.

## 12.3 | Create a job triggered by a delta in a feed

### Abstract

Create a job that is triggered when a feed has complete an operation and there is a change in the content.

Jobs triggered by a delta in a feed (event triggered jobs) run when a feed completes an operation and there is a change in the content. For the job to trigger, there must be a delta between the incoming feed and the previous one. You can define a job to trigger a playbook when the specified feed or feeds finish a fetch operation that includes a modification to the feed. The modification can be a new indicator, a modified indicator, or a removed indicator. For example, you may want to update your firewall every time a URL is added, modified, or removed from the Office 365 feed. You can configure a job that triggers the firewall update playbook to run whenever a modification is made to the feed.

For an example of using a job triggered by a delta in a feed, see the Create jobs to process indicators example.

**NOTE:**

A job triggered by a delta in a feed runs only if there is a change in the feed, and does not run on a feed's initial fetch. For the initial fetch, you can run the playbook manually and then set up an event triggered job for subsequent fetches.

If you want to trigger a job after a feed completes a fetch operation and the feed does not change frequently, you can select the Reset last seen option in the feed integration instance. The next time the feed fetches indicators, it will process them as new indicators in the system.

1. Select Jobs → New Job.
2. Select Triggered by delta in feed.
3. Add or create any relevant tags to use as a search parameter in the system.
4. In the Trigger section, select one of the following:
  - Any feed: The playbook runs when a modification is made to any feed.
  - Specific feeds: Select the feed instances that will trigger the playbook to run when a modification is made to them.
5. In the BASIC INFORMATION section:
  - Add a meaningful name for the job.
  - Select the playbook you want to run when the conditions for the job are met.
6. Create new job.

## 12.4 | Create jobs to process indicators example

### Abstract

Provides an example of a job triggered by a delta in a feed to process incoming indicators and a time triggered job to push indicators to a SIEM.

In this example, when indicators are fetched from a threat intel feed, a job triggers a playbook to enrich the indicators to determine which indicators should be investigated. A time triggered job then pushes the relevant indicators to your SIEM.

Use the following integration and playbooks to ingest and process the indicators.

Content Item	Description
Unit 42 Intel Objects Feed integration	This integration fetches a list of threat intel objects, including Campaigns, Threat Actors, Malware, and Attack Patterns, provided by Palo Alto Network's Unit 42 threat researchers.
TIM - Process Indicators - Manual Review playbook	<p>This playbook tags indicators ingested by feeds that require manual approval. To enable this playbook, the indicator query needs to be configured. The playbook uses the Indicator Auto Processing sub-playbook, which identifies indicators that should not be added to a blocked list, such as IP indicators that belong to business partners or important hashes.</p> <p>For the TIM - Process Indicators - Manual Review playbook to run, it needs to be triggered by a job. The job concludes by creating a new incident that includes all the indicators that the analyst must review.</p>
TIM - Add All TIM - Add All Indicators Types to SIEM playbook	<p>This playbook sends to the SIEM only indicators (IP, bad hash, domains, and URLs) that have been processed and tagged accordingly after an automatic or manual review process.</p> <p>By default, the playbook is configured to work with ArcSight and QRadar, but change this to match the SIEM in your system.</p>

### Task 1. Configure the Unit 42 Intel Objects Feed to ingest indicators

If you have a TIM license, this feed is preconfigured.

1. Go to Settings & Info → Settings → Integrations → Instance and search for Unit 42 Intel Objects Feed.
2. Click Add instance.
3. In the Collect section, select Fetches indicators.
4. Test the Feed to ensure that it is working correctly.
5. Save and Exit.

### Task 2. Create a list of indicators to exclude

Before customizing the playbook, we recommend creating a list of indicators that you want to exclude from the manual review process. In this example, we will create a list of business partner IP addresses.

1. Select Settings & Info → Settings → Advanced → Lists → Add a List.
2. Enter a meaningful name for the list. For example, BusinessPartnersIPAddresses.
3. In the Content Type field, select Text.
4. Select who can view or edit the list in the PERMISSIONS section.
5. In the list enter a comma-separated list of IP addresses of your business partners.
6. Save the list.

#### Task 3. Customize the TIM - Process Indicators - Manual Review playbook to process the indicators

1. Go to Playbooks and search for TIM - Process Indicators - Manual Review and either detach or duplicate the playbook.

**NOTE:**

If you detach a playbook, it does not receive content pack updates until it is reattached, but then your changes are discarded. Duplicate the playbook if you want to receive content pack updates and keep your changes.

2. Click the Playbook Triggered task at the top of the playbook.
  - a. Change From Context data → Inputs → General (Inputs group) → OpenIncidentToReviewIndicatorsManually the value to Yes, so an incident with the indicators for review is created.
  - b. Select the From indicators radio button.
  - c. Under Query, enter a query to process the specific indicators that you want. For example, `sourceBrands:"Unit42IntelObjectsFeed"`.
  - d. Save the task and then save the playbook.
3. Update the TIM - Indicator Auto Processing sub-playbook and either detach or duplicate the playbook.
  - a. To exclude business partner IP addresses that you defined in Task 2, locate and edit the TIM - Process Indicators Against Business Partners IP List task.
  - b. From the Inputs tab, under BusinessPartnersIPListName, select the source, and under LISTS, add the created list.
  - c. Save the task.
4. Save the playbook.

#### Task 4. Define a job to trigger the playbook when indicators are fetched

1. Go to Jobs → New Job → Triggered by delta in feed.
2. Go to Incident Response → Jobs → New Job → Triggered by delta in feed.
3. From the TRIGGERS section, select Specific feeds and add the feed configured in Task 1.
4. Add the name of the job.
5. In the Playbook field, add the playbook customized in Task 3.
6. Create the job.

Whenever indicators are ingested from Unit 42, the playbook runs and creates an incident if an incident needs to be reviewed. You can track the status of the job in the table on the Jobs page.

You can now add indicators to a SIEM.

#### Task 5. Customize the TIM - Add All TIM - Add All Indicators Types to SIEM playbook

1. Go to Playbooks and search for TIM - Add All Indicator Types to SIEM and either detach or duplicate the playbook.

**NOTE:**

If you detach a playbook, it does not receive content pack updates until it is reattached, but then your changes are discarded. Duplicate the playbook if you want to receive content pack updates and keep your changes.

2. Click the Playbook Triggered task at the top of the playbook.
  1. Select From indicators and set the query for the indicators to add. For example `tags:approved_black, approved_white`.

The purpose of the playbook is to send to the SIEM only indicators that have been processed and tagged accordingly after an automatic or manual review process. The playbook comes out-of-the-box with queries that you can update if required.

## 2. Save the playbook.

Ensure the playbook includes a task that closes the investigation once it is completed.

### Task 6. Define a time triggered job to push the indicators to the SIEM

1. Select Jobs → New Job.
2. Select Time Triggered.
3. (Optional) Select **Recurring** and determine how often you want the job to run. For example, run once a day at midnight.
4. Enter a name for the job.
5. In the Playbook field, select the TIM - Add All Indicator Types To SIEM playbook to run.
6. Create new job.

Whenever an indicator is ingested that has a relevant tag such as approved\_list, the job pushes that indicator to the SIEM.

### Task 7. Test the workflow

1. Open the job that you created to process indicators from Task 3.  
You can tag any indicator with the tags that you want to push. It does not have to be this job.
2. In the Work Plan, open the Create Process Indicators Manually incident task.
3. In the Outputs tab, copy the incident ID for the incident that was created.
4. Go to Incidents and search for the incident ID that was created.
5. Review the indicators and update the indicators with tags that you want to push to the SIEM.
6. When finished with the review, in the Work Plan, click the Manually review the incident task, select Yes, and Mark Completed.
7. Select the job you defined in Task 6 and click Run now.
8. Go to Indicators and run the query **tags:SIEM**.

This tag is appended to every indicator that has been processed and pushed to the SIEM.

## 13 | SLAs

### Abstract

SLAs enable you to define specific goals and responsibilities and improve quality and availability in your investigations.

Service Level Agreements (SLAs) empower you to define clear expectations, prioritize incidents effectively, and ensure efficient resolution. Configure SLAs within incident types and fields, and set automated timers directly in your playbooks or scripts for enforcement. Additionally, you can manage SLA/timers through the CLI.

### 13.1 | SLAs in Cortex XSOAR

#### Abstract

SLA fields count down the time remaining. SLAs fields can be incorporated in cases. You can trigger actions in the event the SLA passes.

SLAs are an important aspect of case management in Cortex XSOAR. SLAs enable you to define specific goals and responsibilities and improve quality and availability. Analysts can prioritize incidents and ensure that those incidents are handled efficiently. Managers can see an overview of those incidents, improve reaction time, and measure success.

You can do the following:

Action	Description
Define SLAs in incident types and fields	<p>Incorporate SLAs into your incidents to set how long an action should take. SLAs are not enforced inherently, but can be configured to be acted upon by the user. You can view how much time is left before the SLA becomes due, as well as configure actions to take if the SLA passes its due date.</p> <p>You can define an SLA in an incident type, which occurs when the incident is created. These global settings apply when the incident opens until closed. Some out-of-the-box incident types have the SLA defined by default. For more information, see Configure an SLA in an incident type.</p> <p>You can also define an SLA in an incident field for more granular control, such as setting the time to assign an incident. For more information, see Configure Timer/SLA fields.</p> <p>When set up, you can see the SLAs for the incident type and incident fields in the incident table and incident layout.</p>
Set up Timers	<p>Timer incident fields can be started, stopped, or paused in a playbook, script, or manually in the CLI. These fields give you granular control when tracking the response to a given incident. For example, the Time to Assignment incident field tracks the time to assign an incident that can be started, stopped, or paused.</p> <p><b>NOTE:</b> Timers measure how much time has passed since the event. SLAs measure how much time is left until the event.</p>

#### SLA scripts

You can use SLA scripts to act on breaches, such as sending an email when a breach occurs, or specific changes to an incident field, such as a change of incident owner. Cortex XSOAR includes out-of-the-box scripts or you can create your own script. For more information, see Automate changes to incident fields using SLA scripts.

#### Using the CLI

If you want to set or change the SLA for an incident type or field you can use the `setIncident` command in the CLI. For timers, you can use commands such as `startTimer`, `stopTimer`, and `pauseTimer`. For more information, see Use SLA and Timer field commands manually in the CLI.

#### Incident layouts

When you configure the Timer/SLA fields, you can add them to your incident layout to view the status of the SLA, if any of the SLAs are overdue, and if so, by how much. You can also view the number of cases that are at risk of passing the SLA or are already late. You can set the risk threshold for each incident field or rely on the default setting, which is 72 hours. You can change the default threshold by adding a server configuration. See Configure the Global Risk Threshold.

#### Dashboards

Cortex XSOAR comes out-of-the-box with an SLA dashboard, where you can view SLA information, such as within SLA by type, late SLA by type, mean time to resolution, etc. You can also generate reports such as late incidents, open incidents, etc.

#### Further resources

Watch the following video to see how to set up SLA/Timers in your use case.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

## 13.2 | Configure an SLA in an incident type

#### Abstract

Add SLA time/date to an incident type.

On the Incidents page, in the incident table, you can view the SLA (due date) by default. You can also search using the dueDate parameter, such as dueDate:>="now" to search for incidents that are either due now or overdue. If it has not been set, you need to configure the incident type.

1. Go to Settings → Settings & Info → Object Setup → Incidents.

2. Select the incident type to add the SLA.

Some out-of-the-box incident types have a default SLA date. To update out-of-the-box incident types, you need to either duplicate or detach them.

3. In the SLA field, add the weeks, days, and hours required.

Estimate how long the incident should take from being ingested into Cortex XSOAR until it is closed. For example, if you expect your incident type to be closed within 36 hours, select 1 day and 12 hours.

4. (Optional) Set a reminder before the SLA expires.

The owner of the incident will receive an email that the SLA expiration date is approaching.

5. Save the incident type.

6. (Optional) To test the SLA, go to the integration instance where you ingest incidents.

**NOTE:**

Any previous incident types that were ingested will not have the SLA set. You need to ingest the incidents again.

- a. Open the instance settings and select Fetches incidents (if not already set).

- b. Save the instance.

After the instance fetches incidents, you may want to turn off the fetched incidents setting.

- c. Go to the incidents page and search for the incident type.

You should see the SLA date.

### 13.3 | Configure Timer/SLA fields

#### Abstract

Create a new SLA or timer and add an SLA script to trigger when SLA time has passed.

By default, Cortex XSOAR comes out-of-the-box with several Timer/SLA fields, such as Remediation SLA and Time to Assignment, or create your own Timer/SLA fields. You can use the fields as an SLA, an SLA and timer, or a timer.

Action	Description
SLAs	<p>Set the date in the incident field, which counts the completion time. Use it to create widgets in a dashboard/report and to the incident layout, which is useful to see when an SLA is breached or at risk.</p> <p>You can also add an SLA script, so when an SLA is breached certain actions can occur, such as sending an email. For more information, see Automate changes to incident fields using SLA scripts.</p> <p><b>NOTE:</b></p> <p>Incidents sorted using an SLA/Timer field are sorted by the due date of the SLA field.</p>
SLA Timers	Counts the time elapsed since the incident field started. You can add it to a playbook task or script. It does not run automatically. You need to start/stop/pause it in a playbook, script, or manually in the CLI.

In the following example, configure the SLA information in the Time to Assignment field.

1. Navigate to Settings & Info → Settings → Object Setup → Incidents → Incident Fields.

2. Edit the Time to Assignment field.

**NOTE:**

If creating a new SLA field, in the field type field, select Timer/SLA

3. Set the SLA time.

By default, the SLA field shows hours and minutes. You can change this to days and hours, by clicking Hours.

For example, if you set the SLA for one day and the Time to Assignment has started but not stopped within one day, the analyst will be in breach of the SLA.

#### 4. Set the Risk Threshold.

Useful for dashboards and reports. When the timer falls below this threshold, it is considered at risk. By default, the threshold is 3 days. You can change this by adding a server configuration. See Configure the Global Risk Threshold.

#### 5. Under Run on SLA Breach, select the script to run when the SLA time has passed. For example, the sendEmailOnSLABreach script sends an email when the SLA is breached. For more information, see Automate changes to incident fields using SLA scripts.

**NOTE:**

Only scripts to which you have added the SLA tag appear in the list of scripts you can select.

When you hover over the machine name (below the Field Name) note the name which is used in the command line or script.

#### 6. Save the field.

#### 7. Add the field to the incident layout.

Ensure that the incident layout is used in the incident type you want to view the SLA information.

#### 8. If you want to automate SLA timers, add or configure a playbook to run the timer fields.

In this example, you want to create a new field that notifies a user when it reaches a particular stage in the investigation with an SLA of three days and the risk set to one day.

**New Incident Field**

**Field Type**: Timer/SLA

**\* Field Name**: *TimeToNotify*  
Machine name: timetonotify (use in search and command line)

**Tooltip**: Informs you of the stage of the investigation

<b>Basic Settings</b>		<b>Attributes</b>	
<b>SLA</b>			
3	Days	00	Hours
<b>Risk Threshold</b> ⓘ			
1	Days	03	Hours
Run on SLA Breach: SendEmailOnSLABreach			

## 13.4 | Configure a playbook to run Timers/SLAs

### Abstract

Add or configure a playbook to run SLA timers.

To run a timer, it must be run in a playbook task, a script, or manually in the CLI.

You can set a Timer/SLA field to start running by doing the following:

- In a Timer/SLA field such as the `Time To Assignment` field, you can control all incidents that use the field regardless of the playbook configured for them by configuring a script to run when the `Owner` field changes.. This method automatically stops the timer when an analyst is assigned. See [Automate changes to incident fields using SLA scripts](#). The advantages of using this option are scalability and consistency.
- Stop the field through a playbook. The Timer/SLA field can be triggered to start, pause, or stop when a certain task occurs. For example, a timer can be triggered to stop for the `Time to Assign` field when the incident is assigned an owner, and to immediately start the timer for the `Time to Remediation` field.

In a playbook, you add timers to specific tasks to manage SLAs.

When defining a Timer in a task or section header, in the Timers tab, select the action that you want the timer to perform for the task.

**NOTE:**

If creating tasks for SLAs they do not have to execute anything. You can also use section headers.

Valid options are:

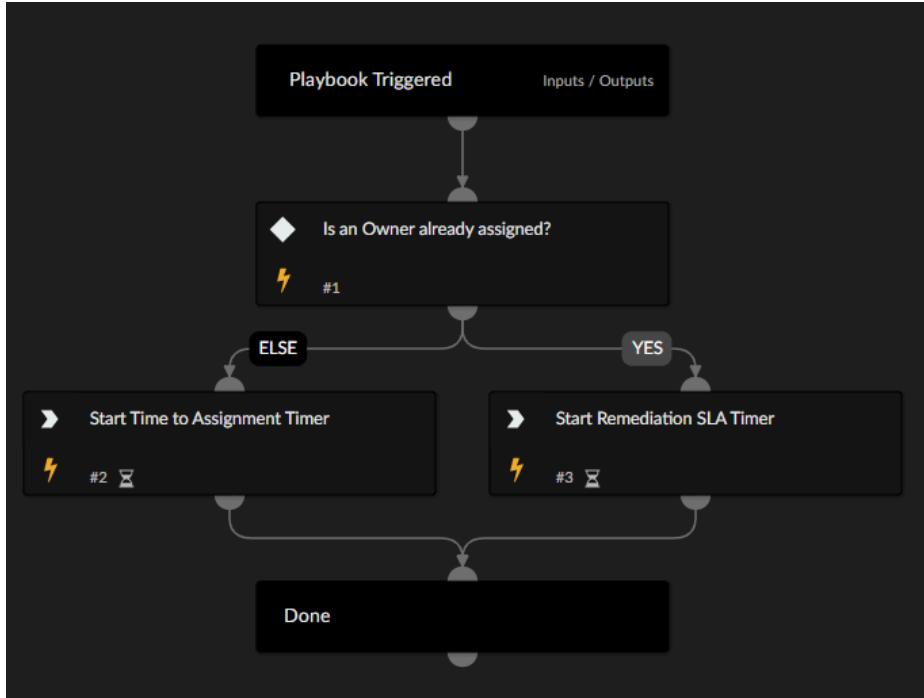
Option	Description
<code>Timer.start</code>	<p>Starts the timer.</p> <p><b>NOTE:</b></p> <p>Timers are not started automatically when an incident is created.</p>
<code>Timer.pause</code>	Pauses the timer.
<code>Timer.stop</code>	<p>Stops the timer.</p> <p><b>NOTE:</b></p> <p>Timers are automatically stopped when an incident is closed. After a timer is stopped, you can only reset a timer using the <code>resetTimer</code> command in the CLI.</p>

Some playbooks, such as Phishing - Generic v3, come out-of-the-box with SLA timer tasks included. If you need the same timers across use cases, create a sub-playbook based on your use case or conditions such as incident severity.

Although you can create your own SLA sub-playbooks, the CaseManagement - Generic content pack includes several SLA playbooks, which you can configure. For more information, see the CaseManagement - Generic content pack.

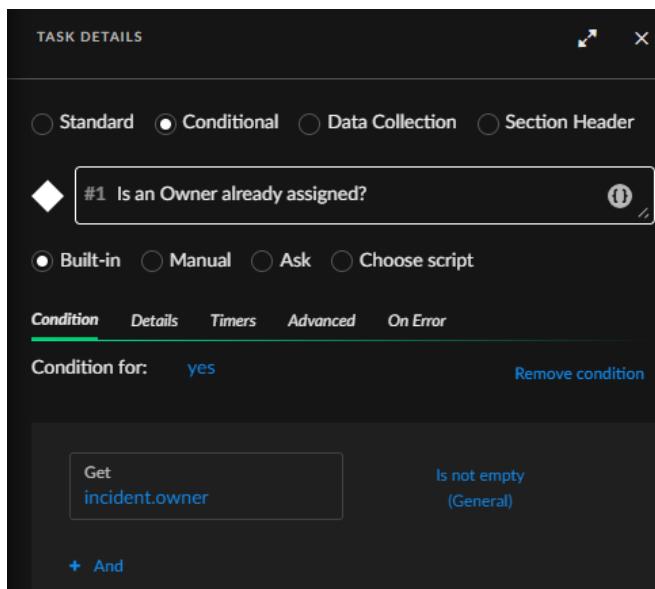
#### The Case Management - Generic - Start SLA Timers playbook

The Case Management - Generic - Start SLA Timers playbook starts the `Time to Assignment` or `Remediation` SLA timers field based on whether an owner is assigned to the Incident. You can add this as a sub-playbook to your use case.

**NOTE:**

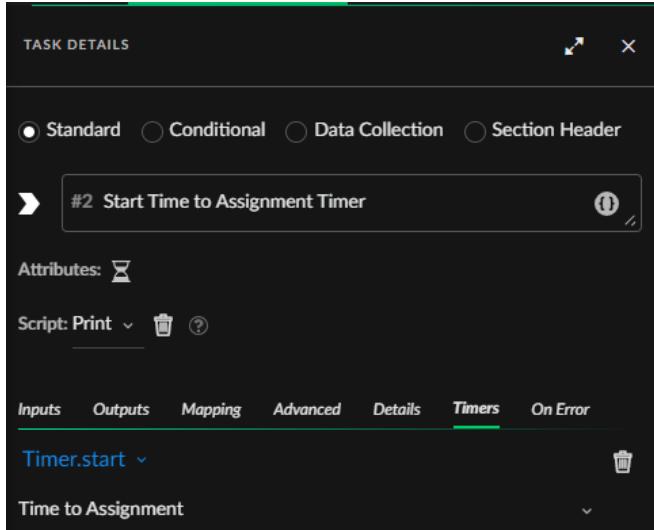
When a task or section has a Timer/SLA action configured, it displays the hourglass icon.

1. The first task is a conditional task which determines whether an `incident.owner` has been assigned.

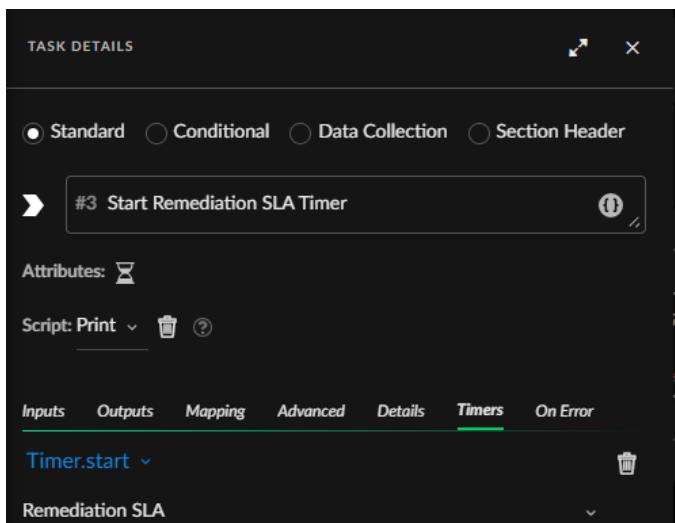


2. On the left-hand side task, if no owner is assigned the Time to Assignment timer starts.

The Print script returns details to the War Room confirming that the script has started to run.



3. On the right-hand side task, if an owner is assigned the Remediation SLA timer starts.



#### NOTE:

If you want to stop or pause a timer in a playbook, you can use an existing or create a section header/task. When you select Timer.stop, the run is considered finished and cannot be restarted without setting it to zero. If you want to restart the timer, select Timer.pause so you do not lose the accumulated time. By default, all timers stop when the incident closes.

Add the sub-playbook to the main playbook, as required.

#### Case Management - Generic - Set SLAs based on Severity playbook

This playbook sets the SLAs for incidents, the Time to Assignment Timer, and the Remediation SLA Timer based on the incident severity using playbook inputs. For example, set the number of minutes for incident and remediation SLAs for critical incidents. For more information, see Case Management - Generic - Set SLAs based on Severity. Add this as a sub-playbook to your use case.

Alternatively, create a playbook or script to modify SLA fields based on certain conditions. For example, in the Set Severity to Medium task, you can add an SLA such as Time to Assignment 15 minutes where there is high severity (3).

## 13.5 | Automate changes to incident fields using SLA scripts

### Abstract

Create scripts to perform specific actions in Cortex XSOAR when the SLA is breached. Properties in the SLA timer field value.

Scripts in Cortex XSOAR enable you to automate processes. In the context of SLA, you can create scripts that will perform specific actions when the SLA is breached. Each SLA script must include the SLA tag.

You can use an out-of-the-box script and attach it to an incident field.

#### Send an email when the SLA is overdue

Cortex XSOAR comes with an out-of-the-box script, called `SendEmailOnSLABreach`, that sends an email to specific users when the script is triggered. You can add this to any incident field as required. For example, add the script to the Remediation SLA incident, so that when an SLA/Timer is breached, an email is sent automatically. By default, the script sends an email to the incident assignee, but you can manually edit the script to add additional recipients..

#### Stop and start different timers in an incident field

In the following example, you want to stop the Time To Assignment timer when an owner is assigned and start the Remediation SLA timer.

If you have not done so already, download the CaseManagement-Generic content pack. This content pack includes the `TimersOnOwnerChange` script.

1. Go to Settings & Info → Settings → Object Setup → Incidents → Incident Fields
2. Edit the Owner field.
3. In the Script to run when field value changes field, add the `TimersOnOwnerChange` script.
4. Save the field.
5. (Optional) Test the field change.
  - a. Open an unassigned incident and in the CLI type `!startTimer timeField=timetoassignment`.

In the War Room, the field returns the new value from idle to running.

- b. Go to the Incident Info tab and add an owner.
- c. In the War Room, you should see that the Time to Assignment has ended and the Remediation SLA has started:

Incident was modified.		
Field changes		
Field	Old Value	New Value
Owner		xsoar-admin@panw.com
Remediation SLA	Status: idle	Status: running (2024-01-26T11:10:16Z)
Time to Assignment	Status: running	Status: ended

## 13.6 | Create SLA scripts

### Abstract

Create scripts that perform specific actions in Cortex XSOAR when the SLA is breached. Properties in the SLA timer field value.

When you create your scripts, the following arguments are automatically added, in addition to the basic elements provided with every script (for example, current investigation and current incident):

- **field**: The current triggered SLA breach field object (contains: name, cliName, threshold and more.).
- **fieldValue**: The current triggered SLA field's value. For example the `startDate`.

The following table lists the different properties in the SLA timer field value:

Property	Type	Description
dueDate	Date	The date by which the SLA for this timer is due.
breachTriggered	Boolean	Whether the timer was already in breach of the SLA.
sla	INT (in minutes)	The period is defined as the SLA for this timer. This is the value that you defined in the Timer field.
endDate	Date	The date at which the SLA timer is completed.
lastPauseDate	Date	The last date at which the SLA timer was paused.
startDate	Date	The date at which the SLA timer was started.
accumulatedPause	INT (in seconds)	The total number of seconds that the timer was in a paused state.
totalDuration	INT (in seconds)	The total number of seconds that the timer was running. This property is populated after the timer is stopped.
slaStatus	INT	Represents the Cortex XSOAR SLA status. Values are: <ul style="list-style-type: none"> <li>◦ -1: The SLA has not been set.</li> <li>◦ 0: The SLA is within the allotted range.</li> <li>◦ 1: The SLA is below the defined risk threshold.</li> <li>◦ 2: The SLA is in breach.</li> </ul>
runStatus	String	Represents the current status of the timer. Values are: <ul style="list-style-type: none"> <li>◦ idle</li> <li>◦ running</li> <li>◦ paused</li> <li>◦ ended</li> </ul>

See the following video for a practical example of creating an SLA script and how to use it in a playbook.

Terjadi error.

---

Cobalah menonton video ini di [www.youtube.com](https://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

## 13.7 | Use SLA and Timer field commands manually in the CLI

### Abstract

Use timers and SLA commands for a specific incident, such as decreasing the required response time for a high-priority incident.

You can manage the timers and SLA for a specific incident manually in the CLI, which enables you to manage SLAs on a global level and a more granular level within specific incidents when the need arises. For example, if the severity of the incident dictates that you decrease the response time for the given incident.

### Set Timer/SLA fields

Use the `setIncident` command to set the SLA incident due to date or to set a specific SLA field in an incident. When adding the `sla` parameter to the command, it sets the time for the incident's due date. If you also add the `slaField` you set the SLA for the incident field.

For example, to change the Time to Assignment field to 30 minutes in the current incident:

```
!setIncident sla=30 slaField=timetoassignment
```

To change the SLA time to February 1, 2024, at 11.12 am:

```
!setIncident sla=2024-02-01T11:12
```

#### NOTE:

When defining the values for the `slaField` use the machine name for the field, which is lowercase and without spaces. You can check the machine name by editing the incident field. For example, the Remediation SLA field is `remediationsla`.

### Start/stop Timer/SLA fields

Use the following commands in the CLI:

Command	Description
<code>startTimer</code>	Starts the timer in a Timer/SLA field. For example, <code>!startTimer timerField=timetoassginment</code> . This command can also be used to restart a paused timer.  NOTE: Timer/SLA fields are not started automatically when an incident is created unless run in a playbook.
<code>pauseTimer</code>	Pauses the timer in a Timer/SLA field. For example, <code>!pauseTimer timerField=timetoassignment</code> . Use this command when a Timer/SLA field has started.
<code>stopTimer</code>	Stops the timer in a Timer/SLA field. For example, <code>!stopTimer timerField=timetoassignment</code> . After a Timer/SLA field is stopped, you can only reset a timer using the <code>resetTimer</code> command.  NOTE: Timers are automatically stopped when an incident is closed.
<code>resetTimer</code>	Resets a timer in a Timer/SLA field, which resets the elapsed time, and the status of the timer for the incident. This command should be used to enable a timer that was stopped. For example, <code>!pauseTimer timerField=timetoassignment</code> .

#### NOTE:

When running the commands, you can specify the `incidentID` to change the timer for a different incident.

## 13.8 | Configure the Global Risk Threshold

### Abstract

Add server configuration in Cortex XSOAR to change the SLA Risk threshold from the default 72 hours.

By default, the risk threshold is 72 hours. You can change the threshold by adding a parameter to the system settings.

**NOTE:**

When changing the server configuration, the new value does not affect existing fields retroactively. It affects new fields that you create.

1. Navigate to Settings & Info → Settings → System → Server Settings → Server Configuration → + Add Server Configuration.
2. Add the following server configuration.
  - a. In the key field, enter `sla.risk.threshold`.
  - b. In the value field, enter the number, in hours, to which to set the risk threshold.
3. Click Save.

## 13.9 | Search incidents for Timer/SLAs

### Abstract

Search incidents based on their SLA status, a SLA field, or a timer field.

You can search for incidents based on their SLA in several ways:

- Based on the SLA status.

**NOTE:**

The SLA status is not defined unless the timer is in a stopped mode, meaning either paused or ended.

- Based on an SLA field.
- Based on a timer field.

For example, you can search for all of the timer fields that are currently running, or you can search for all incidents with a specific SLA status.

1. Navigate to the Incidents page.
2. To search for an incident whose Timer/SLA is still active, enter the following:
  - The name of the field
  - The run status
  - The due date

This parameter is required for queries whose run status is neither ended nor paused, to improve query performance.
3. To search for an incident whose timer is no longer active, enter the SLA Status.

### Examples

In the following example, search for all incidents using the Remediation SLA field that fulfill the following criteria:

- The Remediation SLA run status has not ended or paused AND the due date is later than now OR the SLA status is within time.  
`(-remediationsla.runStatus:(ended paused) and remediationsla.dueDate:>now) or (remediationsla.slaStatus:"within")`
- The Remediation SLA run status has not ended or paused AND the due date is earlier than now OR the SLA status is late.  
`(-remediationsla.runStatus:(ended paused) and remediationsla.dueDate:<now) or (remediationsla.slaStatus:late)`
- The Remediation SLA run status has not ended or paused AND the due date is between now and five hours (the five hours represent our risk threshold) OR the SLA status is Risk.  
`(-remediationsla.runStatus:(ended paused) and remediationsla.dueDate:>now and remediationsla.dueDate:<"in 300 minutes") or (remediationsla.slaStatus:risk)`

## 14 | Dashboards and Reports

### Abstract

Create, edit, and share dashboards and reports in Cortex XSOAR. Add widgets to a dashboard and configure a default dashboard

Create or modify dashboards and reports, schedule automated reports for recurring needs, and design custom widgets to suit your visualization goals. Leverage fully customizable widgets from different sources and display them in clear formats like graphs, pie charts, and text.

## 14.1 | Dashboards

### Abstract

Create, edit, and share dashboards in Cortex XSOAR. Add widgets to a dashboard and configure a default dashboard.

Dashboards offer graphical overviews of your tenant's activities, enabling you to effectively monitor incidents and overall activity in your environment. Each dashboard comprises widgets that summarize information about your endpoint in graphical or tabular format.

### Default dashboards

Cortex XSOAR provides several out-of-the-box dashboards, including the following:

**NOTE:**

If you install a content pack which contain dashboards, these can be added from the More Dashboards dropdown. To change the order of the dashboards, hover over the six block icon next to a dashboard name. When the cursor turns into a hand, drag and drop the dashboard into the required location.

Dashboard	Description
My Dashboard	A personalized dashboard showing your incidents, tasks, etc.
My Threat Landscape	Information about malicious/suspicious indicators in incidents, top 10 indicators in related incidents, Unit 42 feed (if enabled).
SLA	Information about your Service Level Agreements.
Troubleshooting Playbooks	Information about playbook run and execution errors.
Incidents	Information about incidents, such as severity type, active incidents, unassigned incidents, etc.
API Execution Metrics	Information about API calls. You can use the API Execution Metrics for Enrichment Command widget for troubleshooting and to make decisions about indicator enrichment.
Cost Optimization Playbooks	Information about playbooks including task executions, average runtime, etc.
Troubleshooting Instances	Information about integration instance errors.
Threat Intelligence Feeds	Information about TIM feeds that are being ingested into Cortex XSOAR.
Cost Optimization Instances	Information about commands that have been executed in Cortex XSOAR.
MITRE ATT&CK	Information about MITRE ATT&CK techniques. Part of the MITRE ATT&CK content pack.  <b>NOTE:</b> You can add this to your displayed dashboards when clicking More dashboards.

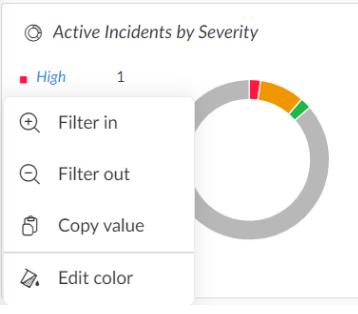
Dashboard	Description
Threat Intel Management	<p>Information about active indicators by reputation, type, expired indicators, etc.</p> <p><b>NOTE:</b></p> <p>You can add this to your displayed dashboards when clicking More dashboards.</p>
VirusTotal API Execution Metrics	<p>Information about VirusTotal API commands. Part of the VirusTotal content pack.</p> <p><b>NOTE:</b></p> <p>You can add this to your displayed dashboards when clicking More dashboards.</p>

#### 14.1.1 | Dashboard actions

##### Abstract

Cortex XSOAR dashboards provide visual data from customizable widgets. Create, edit, import, share and delete Cortex XSOAR dashboards.

In the Dashboards tab, you can set the date range from which to return data and the refresh rate. In each dashboard you can also do the following.

Dashboard Actions	Description
Filter dashboard data	<p>You can filter dashboard data by either typing the query in the query bar, or in the relevant widget, by clicking Filter In. When clicking Filter In the query is added to the query bar. To filter out, delete the query. For example, if you only want to see active incidents that are high severity, in the Active Incidents by Severity widget, hover over High and click Filter In.</p>  <p>To remove the filter, delete the query.</p> <p><b>NOTE:</b></p> <p>If you want to see more information about the data, click the data to take you to the relevant page. For example, in the Active Incidents by Severity widget, to see only high incidents, click High. This takes you to the Incidents page, where you can see all the active critical incidents.</p> <p>After creating the filter, you can send the URL of the filtered dashboard to other users.</p>
Change the color of legend items in graphs	<p>You can change the color of items (such as indicator types and incident types) in some widgets, depending on the widget type and the chart/graph type. When editing a widget, click the item within the legend in the preview window on the right. The Edit color option appears and you can select the color for the item.</p> <p>If you edit the color after a widget has been added to a dashboard or report, the change only applies to the widget within that dashboard or report. If you edit the widget directly in the Widgets Library before adding it to a dashboard or report, the change is applied every time you add the widget to a dashboard or report. Changes to an item within a widget only apply within that widget. For example, changing the color for the Phishing incident type within the Active Incidents widget only applies to Active Incidents, and not other widgets that contain incident types.</p>

Dashboard Actions	Description
Copy values from graphs	In the Quick chart definitions window, click an item in the legend and select Copy value. This enables copying the value from the widget for commands in the War Room.
Create or edit a dashboard	Design a new interface for specific security investigation needs, or edit an existing dashboard. To edit out-of-the-box dashboards, you first need to duplicate them.  For more information, see <a href="#">Manage dashboards</a> .
Import and export a dashboard	The dashboard is exported as a JSON file. You can make any changes you require and then import the file, for example between test and production environments.
(Admin only) Define dashboard access	In a production environment, an administrator defines the default dashboard for each user and selects the default dashboards that the user sees when logging into the tenant, depending on a user's role. If a user has not modified their dashboard, these dashboards are added automatically, otherwise users can add these dashboards to their existing dashboards. These default dashboards can be removed but not deleted, and can be added again if required.  <b>NOTE:</b> You cannot add default dashboards to out-of-the-box roles.  For more information, see <a href="#">Manage roles in the Cortex XSOAR tenant</a> .
Share a dashboard	Sharing dashboards enables collaboration and alignment among security teams by providing real-time visibility into key metrics and insights, facilitating informed decision-making and coordinated response efforts. Out-of-the-box dashboards and dashboards from content packs cannot be shared unless you duplicate them.  For more information, see <a href="#">Manage dashboards</a> .
Create a report	You can generate a report from the dashboard as is, or configure report settings, for example add new widgets to the report, change the report format, and schedule running a report. To create a report from a dashboard, click  and select Create report. Click Run Now to generate the report.  For more information, see <a href="#">Manage reports</a> .

#### 14.1.2 | Manage dashboards

##### Abstract

Create and customize a dashboard in Cortex XSOAR, including adding widgets to a dashboard. Share a dashboard.

You can create or edit dashboards according to your organization's requirements.

Creating a new dashboard enables designing a personalized interface for specific security investigation needs. This facilitates quick access to critical information and enhances operational effectiveness.

Editing an existing dashboard enables security teams to focus on the most relevant information. By presenting only the most relevant data and metrics, investigation and response is more efficient and streamlined.

Once you create or edit a dashboard, you can share it with relevant roles to facilitate collaboration and enhance visibility into relevant data and insights across teams.

##### Create a dashboard

1. To create a new dashboard, select Dashboards & Reports → Dashboards → More Dashboards → New Dashboard.
2. Enter a name for the dashboard.
3. Select the date range for a dashboard.

From the Date Range dropdown list, set the date range for the dashboard.

By default, a widget inherits the date range that you specify when creating the widget. If the date range for the report or dashboard does not include the widget date range, the data is blank. To change the widget's date range, click  and select Use Widget's date range or Use Dashboard's date range. By default, the dashboard's date range is used and the option in the dropdown shows as Use Widget's date range. If you change this to use the widget's date range, the dropdown then shows the option to Use Dashboard's date range.

**NOTE:**

Each widget can have its own date range, which can be different from the dashboard's date range.

#### 4. Add a widget to a dashboard.

1. Click  to create a custom widget or in the Widgets Library, find a relevant existing widget and click Add.

2. To edit a widget in the dashboard, select  then Edit widget.

The edits to the widget in the dashboard apply only for the report. If you want to make changes that are available for other users, dashboards, or reports, edit the widget directly in the Widgets Library by clicking the pencil edit icon.

3. To add a new widget from the Widgets Library, follow the procedure in Create a widget using the widget builder.

4. Click Save.

#### 5. Save the dashboard. If you select Save Version, you can view a history of the changes made to your dashboard and you can revert to previous versions.

#### Edit a dashboard

##### 1. Go to Dashboards & Reports → Dashboards.

##### 2. Select the dashboard you want to edit.

##### 3. Click and select Edit, or for an out-of-the-box dashboard click Duplicate.

##### 4. Enter a name for the dashboard.

##### 5. Select the date range for a dashboard.

From the Date Range dropdown list, set the date range for the dashboard.

By default, a widget inherits the date range that you specify when creating the widget. If the date range for the report or dashboard does not include the widget date range, the data is blank. To change the widget's date range, click  and select Use widget's date range or Use dashboard's date range. By default, the dashboard's date range is used and the option in the dropdown shows as Use widget's date range. If you change this to use the widget's date range, the dropdown then shows the option to Use dashboard's date range.

**NOTE:**

Each widget can have its own date range, which can be different from the dashboard's date range.

#### 6. Add a widget to a dashboard.

1. Click  to create a custom widget or in the Widgets Library, find a relevant existing widget and click Add.

2. To edit a widget in the dashboard, select  then Edit widget.

The edits to the widget in the dashboard apply only for the report. If you want to make changes that are available for other users, dashboards, or reports, edit the widget directly in the Widgets Library by clicking the pencil edit icon.

3. To add a new widget from the Widgets Library, follow the procedure in Create a widget using the widget builder .

4. Click Save.

#### 7. Save the dashboard. If you select Save Version, you can view a history of the changes made to your dashboard and you can revert to previous versions.

#### Share a dashboard

##### 1. Go to the dashboard to share.

##### 2. Click and select Share.

To share an out-of-the-box dashboard, you need to duplicate it and share the copy.

##### 3. In the Share dashboard dialog box, select the roles with whom you would like to share this dashboard and their permission levels.

Dashboards can be shared for all roles or for specific roles, with the following permission levels.

- Read only: Copy, export, import and remove the dashboard (cannot share or edit).

Once shared, regardless of who creates the dashboard, any user who has read and write permissions can change the sharing options including to stop sharing. If an analyst who created and shared the dashboard deletes the shared dashboard, it is removed from all users.

- Read & Edit: Edit, copy, share, export import, and remove the dashboard. For example, analysts may want to enable and encourage team-based dashboards, so that dashboards can be edited and maintained by more than a single user.

**NOTE:**

Only dashboard owners can delete their dashboards.

4. Click Save.

The dashboard is now shared among other analysts in the specified role or all roles.

5. (Users) To add a shared dashboard, from the home page, select More Dashboards and select the shared dashboard from the drop-down.

Instead of a user adding the dashboard, you can send the URL to the user, after sharing the dashboard.

**NOTE:**

If you are using a remote repository, all dashboards are automatically shared in the development environment. As a result, the Share option can not be selected from the Settings menu.

6. To stop sharing a dashboard:

a. Click  and select Share.

b. In the Share dashboard dialog box, remove all the roles.

c. Click Save.

## 14.2 | Reports

### Abstract

Create, edit, and customize reports in Cortex XSOAR. Schedule reports with Cron expressions.

Reports contain statistical data in the form of widgets, which enable you to analyze data from inside or outside Cortex XSOAR in different formats such as graphs, pie charts, or text.

After generating a report, it also appears in the Reports tab for future reference.

### 14.2.1 | Manage reports

#### Abstract

Create a new report or customize an existing report in Cortex XSOAR, including adding widgets and changing the timezone and time format in a report. Schedule and generate a report.

You can create and edit reports in the Reports tab, including adding widgets, scheduling times, setting incident time range, adding recipients, and changing the format and size. Reports support PDF and CSV.

#### Report actions

You can do the following with reports.

Report Action	Description
Create or edit a report	<p>When creating a report, what you see is what you get. How you configure the report is how it generates. You can add widgets to a report, change the format and paper size, and insert page breaks by adding the Page Break widget. If you have a table widget that contains many rows, you can select the number of rows on each page or print the whole table (in the table widget, right click and select Force Print full Chart).</p> <p>You can add your own logo by going to Settings &amp; Info → Settings → System → Server Settings → Logo Configuration and uploading your logo in the Full-size logo field. Reports are generated in PDF or CSV formats.</p>

Report Action	Description
Create a report from a dashboard	<p>You can create a report from the dashboard as is, or add new widgets as required. You have the same functionality as custom reports, such as format, when to run, and orientation. To create a report from the dashboard, on the Dashboards page click  and select Create report.</p>
Schedule a report	<p>You can schedule a report to run specific times, or run the report immediately. You can also send the report to specific recipients, and restrict the report according to roles.</p>
Generate an out-of-the-box report	<p>Cortex XSOAR comes with out-of-the-box reports, such as critical and high incidents, daily incidents, and last 7 days incidents. You can change the time range for the incidents, the scheduled time and who can receive the report. If you want to make more comprehensive changes to out-of-the-box reports, copy or download (and then upload) the report.</p>
Schedule a report from an incident	<p>Captures investigation-specific data and shares it with team members. You can customize how the information is displayed for existing incidents.</p>

#### Create a report

1. In the Dashboards & Reports page Reports tab, select New Report.

2. Enter a name for the report.

3. Add a widget to the report.

1. Click  to add a custom widget or select an existing widget from the Widgets Library.

2. To edit the widget in the report, select  then Edit widget.

The edits to the widget in the report apply only for the report. If you want to make changes that are available for other users, dashboards, or reports, edit the widget directly in the Widgets Library by clicking the pencil edit icon.

3. To add a new widget from the Widgets Library, follow the procedure in Create a widget using the widget builder.

4. Click Save.

4. Define the report settings.

Report Setting	Description
Date Range	<p>The Date Range for the report. Default is Last 7 days.</p> <p>By default, a widget in a report inherits the date range that you specify when creating the report. If the date range for the report does not include the widget date range, the data is blank. To change the widget's date range, click  and select Use widget's date range or Use dashboard's date range. By default, the dashboard's date range is used and the option in the dropdown shows as Use widget's date range. If you change this to use the widget's date range, the dropdown then shows the option to Use dashboard's date range.</p> <p><b>NOTE:</b></p> <p>Each widget can have its own date range, which can be different from the report's date range.</p>

Report Setting	Description
Schedule	<p>You can schedule a report to run at specific times with start and end dates. You can also add restrictions on the report content and the number of recipients.</p> <p>If you want to send the report to users by email, you need to add an email integration instance, such as EWS, Gmail, or Mail Sender. Default is Disabled.</p> <p>To schedule a report:</p> <ol style="list-style-type: none"> <li>Under the Schedule field, click Disabled or the date it was last run.</li> <li>In the dialog box, add the following information: <ul style="list-style-type: none"> <li>A comma-separated email list of report recipients.</li> <li>Select the Scheduled checkbox.</li> </ul> </li> <li>If you want to restrict the content of the report according to roles, in the Run as Roles field, from the dropdown list, select one of the roles.</li> <li>Schedule a report according to one of the following methods: <ul style="list-style-type: none"> <li><b>Human view:</b> Schedules a report according to the set number of hours. You can add days of the week with start and end times. When scheduling a report in the Human view the Next Run date may be incorrect. You may need to change the number of hours field when scheduling the report.</li> <li><b>Cron view:</b> Schedules a report according to a Cron time string format, which consists of five fields that Cron converts into a time interval. Use this view to schedule a report on certain hours, days, months, years, and so on. For examples of Cron strings, see Report scheduling examples.</li> </ul> </li> </ol> <p><b>NOTE:</b> When using the <b>Cron view</b>, the Start at and Ends fields may conflict with Cron string expressions. For example, when using frequencies (i.e. '/') if you type the expression <b>0 */6 * * *</b> (runs every 6 hours), with a start time of 15:00, the next run time is not 21:00. The run time depends on Cron run times, which are 00:00, 06:00, 12:00, and 18:00 per day. In this example, the report runs at 15:00, 18:00, and then 00:00, etc. For examples using Cron generally, see Cron examples.</p> <ol style="list-style-type: none"> <li>Select the Start at date and time.</li> <li>Select the Ends condition. Options are: <ul style="list-style-type: none"> <li>Never (default)</li> <li>by (specify the date)</li> <li>after (specify the date)</li> </ul> </li> <li>Select Run Now to run the report immediately. If you click Save, the report appears in the main Reports tab with the scheduled run date in the Next Run field.</li> </ol>
Recipients	<p>A comma-separated email list of report recipients. Default is None.</p> <p>If you want to send the report to users by email, you need to add an email integration instance, such as EWS, Gmail, or Mail Sender.</p>
Format	<p>The report file format. Options are PDF (default) or CSV.</p> <p><b>NOTE:</b> Only tables and text based widgets are exported in CSV. Other widgets are ignored.</p>
Orientation	<p>Sets the report display orientation. Options are Portrait (default) or Landscape.</p> <p><b>TIP:</b> We recommend using landscape orientation to ensure that all information displays in the report.</p>

Report Setting	Description
Paper Size	Sets the report paper size. Options are: <ul style="list-style-type: none"> <li>• A4 (default)</li> <li>• A3</li> <li>• Letter</li> </ul>

5. Save the report. If you select Save Version, you can view a history of the changes made to your report and you can revert to previous versions.

#### Edit a report

1. On the Dashboards & Reports page Reports tab, select the Duplicate Report icon for the report you want to edit.

2. Enter a name for the report.

3. Add a widget to the report.

1. Click  to add a custom widget or select an existing widget from the Widgets Library.

2. To edit the widget in the report, select  then Edit widget.

The edits to the widget in the report apply only for the report. If you want to make changes that are available for other users, dashboards, or reports, edit the widget directly in the Widgets Library by clicking the pencil edit icon.

3. To add a new widget from the Widgets Library, follow the procedure in Create a widget using the widget builder.

4. Click Save.

4. Define the report settings.

Report Setting	Description
Date Range	<p>The Date Range for the report. Default is Last 7 days.</p> <p>By default, a widget in a report inherits the date range that you specify when creating the report. If the date range for the report does not include the widget date range, the data is blank. To change the widget's date range, click  and select Use widget's date range or Use dashboard's date range. By default, the dashboard's date range is used and the option in the dropdown shows as Use widget's date range. If you change this to use the widget's date range, the dropdown then shows the option to Use dashboard's date range.</p> <p><b>NOTE:</b></p> <p>Each widget can have its own date range, which can be different from the report's date range.</p>

Report Setting	Description
Schedule	<p>You can schedule a report to run at specific times with start and end dates. You can also add restrictions on the report content and the number of recipients.</p> <p>If you want to send the report to users by email, you need to add an email integration instance, such as EWS, Gmail, or Mail Sender. Default is Disabled.</p> <p>To schedule a report:</p> <ol style="list-style-type: none"> <li>Under the Schedule field, click Disabled or the date it was last run.</li> <li>In the dialog box, add the following information: <ul style="list-style-type: none"> <li>A comma-separated email list of report recipients.</li> <li>Select the Scheduled checkbox.</li> </ul> </li> <li>If you want to restrict the content of the report according to roles, in the Run as Roles field, from the dropdown list, select one of the roles.</li> <li>Schedule a report according to one of the following methods: <ul style="list-style-type: none"> <li><b>Human view:</b> Schedules a report according to the set number of hours. You can add days of the week with start and end times. When scheduling a report in the Human view the Next Run date may be incorrect. You may need to change the number of hours field when scheduling the report.</li> <li><b>Cron view:</b> Schedules a report according to a Cron time string format, which consists of five fields that Cron converts into a time interval. Use this view to schedule a report on certain hours, days, months, years, and so on. For examples of Cron strings, see Report scheduling examples.</li> </ul> </li> </ol> <p><b>NOTE:</b> When using the <b>Cron view</b>, the Start at and Ends fields may conflict with Cron string expressions. For example, when using frequencies (i.e. '/') if you type the expression <b>0 */6 * * *</b> (runs every 6 hours), with a start time of 15:00, the next run time is not 21:00. The run time depends on Cron run times, which are 00:00, 06:00, 12:00, and 18:00 per day. In this example, the report runs at 15:00, 18:00, and then 00:00, etc. For examples using Cron generally, see Cron examples.</p> <ol style="list-style-type: none"> <li>Select the Start at date and time.</li> <li>Select the Ends condition. Options are: <ul style="list-style-type: none"> <li>Never (default)</li> <li>by (specify the date)</li> <li>after (specify the date)</li> </ul> </li> <li>Select Run Now to run the report immediately. If you click Save, the report appears in the main Reports tab with the scheduled run date in the Next Run field.</li> </ol>
Recipients	<p>A comma-separated email list of report recipients. Default is None.</p> <p>If you want to send the report to users by email, you need to add an email integration instance, such as EWS, Gmail, or Mail Sender.</p>
Format	<p>The report file format. Options are PDF (default) or CSV.</p> <p><b>NOTE:</b> Only tables and text based widgets are exported in CSV. Other widgets are ignored.</p>
Orientation	<p>Sets the report display orientation. Options are Portrait (default) or Landscape.</p> <p><b>TIP:</b> We recommend using landscape orientation to ensure that all information displays in the report.</p>

Report Setting	Description
Paper Size	Sets the report paper size. Options are: <ul style="list-style-type: none"> <li>• A4 (default)</li> <li>• A3</li> <li>• Letter</li> </ul>

5. Save the report. If you select Save Version, you can view a history of the changes made to your report and you can revert to previous versions.

#### Generate a report

To generate a report immediately:

1. In the Reports tab, edit the report settings as relevant, including:

- Date Range
- Recipients
- Next Run

2. Click Run.

3. Click  to download the report.

#### TIP:

Ensure that you enable pop-ups in your browser. If reports do not download after you click Run, add the Cortex XSOAR URL to your browser's pop up blocker exceptions. For more information, see Troubleshoot script timeout for reports.

#### 14.2.1.1 | Report scheduling examples

##### Abstract

Examples of scheduling a Cortex XSOAR report using Cron expressions. Cron scheduler format.

The following examples describe how to schedule a report using the Cron scheduler format. The Cron time string format consists of five fields that Cron converts into a time interval. For example, a Cron string of **0 10 15 \* \*** runs a report on the 15th of each month at 10:00 am.

Schedule a report starting January 1 and then monthly

In this example, you want to schedule a report on January 1, 2020 at 0800 (8:00 am) and thereafter on the 1st of each month.

In the Cron Expression field, type **00 8 1 1/1 \***

Number	Description
00	00 in minutes
8	8am
1	The first of each month
1/1	Starting in January, and every month thereafter. If you want the report to start on a different month, change 1/1 to the relevant month, such as 2/1 for February, 3/1 for March and so on.
*	Any day of the week

The reports run at 8am on January 1, 2020, February 1, 2020, March 1, 2020 and so on.

**NOTE:**

Cron calculates the next relevant date. If you want the report to run next month, provided that date has passed in the current month, you do not need to specify the month. For example, assume the date is December 12. To run the report on January 11 at 8:00 am, type 00 8 11 \*. The report starts running on January 11 (and on 11th of each month thereafter). If the current date is December 10, the next run date would be December 11.

Schedule a report for once a year

In this example you want to schedule a report on January 6, 2020 at 0800 (8:00 am) and every year on January 1 (the current date is Thursday 12 December 2019).

In the Cron Expression field, type **00 8 1 1 \***

Number	Description
00	00 minutes
8	8am
1	1st day of each month
1	Starting every January. For different months change the number
*	Any day of the week

The report runs at 0800 on January 1, 2020, January 1, 2021, January 1, 2022, etc.

Schedule a report every week on a Monday

In this example, you need to schedule a report at midnight every week on a Monday (the current date is Thursday, 12 December 2019)

Type the following expression in Cron: **00 0 \* \* 1**

Number	Description
00	00 in minutes
0	Midnight
*	Any day
*	Any month
1	Monday

The report runs on the first available Monday December 16 at midnight, and on December 23, December 30, January 6, etc.

Schedule a report every weekday from February for 6 months

In this example, you need to schedule a report at 1730 (5:30 pm) every weekday (Monday - Friday) starting in February for 6 months (assume the current date is Thursday December 12, 2019).

In the Cron Expression field, type **30 17 \* 2/6 1-5**

Number	Description
30	30 minutes
17	5pm
*	Any day
2/6	Starting in February for the next 6 months.
1-5	Monday to Friday

The report runs at 1730 (5:30 pm) on February 3, 4, 5, 6, 7, etc.

Schedule a daily report

In this example, you need to schedule a report every day at 0600 (6:00 am) (the current date is Wednesday 12 December).

In the Cron Expression field, type **0 6 \* \* \***

Number	Description
00	00 in minutes
6	6am
*	Any day
*	Any month
*	Any day of the week. If you want to run from Monday to Friday, type 1-5. For Sunday to Thursday, type 0-4.

The report runs at 0600 (6:00 am) on December 13, 14, 15, 16 and so on.

#### 14.2.2 | Configure the timezone in a report

Abstract

Change the timezone and time format in a report for report troubleshooting.

You can set the timezone for widgets in a report by adding a server configuration. When this is not specified, the time/date format and timezone are the local time and location where the report is generated.

**NOTE:**

Most out-of-the-box reports, timezone and time formats in a widget cannot be changed unless you copy the report. Some reports, such as Open Incidents, include a title widget, which include the date the report is generated. The report is generated according to the system default.

1. Go to Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.
2. Add the following key and value:

Key	Value
<code>reports.time.zone</code>	The timezone for your report widgets. For example: <ul style="list-style-type: none"> <li>• Asia/Jerusalem</li> <li>• UTC</li> <li>• America/New York</li> <li>• CET</li> <li>• EST</li> <li>• GMT</li> </ul>

3. Save the configuration.

#### 14.2.3 | Troubleshoot script timeout for reports

##### Abstract

Change default timeout value for Cortex XSOAR reports, using a server configuration.

If you generate a report that runs a script and the report or the section in the report that has the script is blank or empty, increase the script timeout value.

Scripts have a default timeout value of three minutes.

1. Select Settings & Info → Settings → System → Server Settings → Server Configuration → + Add Server Configuration.
2. Add the following key and value.

Key	Value
<code>script.timeout</code>	10

3. Save the configuration.

4. Generate the report.

## 14.3 | Widgets

##### Abstract

Create and edit widgets in Cortex XSOAR for reports and for dashboards.

Widgets are visual components that enable you to analyze data internally or externally from Cortex XSOAR, in different formats such as graphs, pie charts, and text from information.

#### 14.3.1 | Widget customization

##### Abstract

Overview of widgets, including methods for creating and adding widgets. Use widgets to analyze and display data in a dashboard or report in Cortex XSOAR.

Cortex XSOAR provides out-of-the-box system widgets, such as Late Incidents and Saved by Dbot (ROI Widget). You can edit these widgets when creating or editing a dashboard or report.

##### NOTE:

Some content packs include a widget that tracks API rate limit errors. You can use this information for troubleshooting and to make decisions about indicator enrichment. From the Widgets Library, click + and choose SOAR Metrics from the dropdown. From the Operations tab, in the Sum field, select Total API Calls. In the Group by dropdown, select API Response Type. Note that this widget only displays data if there is an installed content pack that supports API rate limit information.

You can create custom widgets as follows and then add them to a dashboard or report as required.

Widget Customization Method	Description
Widgets Library	<p>Create a widget using the widget builder in the Widgets Library which is available for all users.</p> <p>For more information, see <a href="#">Create a widget using the widget builder</a>.</p>
From an incident	<p>Create the widget from the Incidents page and then add it to a dashboard or a report.</p> <p>For more information, see <a href="#">Create a widget from an incident</a>.</p>
From an indicator	<p>Create the widget from the Threat Intel (Indicators) page and then add it to a dashboard or a report.</p> <p>For more information, see <a href="#">Create a widget from an indicator</a>.</p>
In the War Room	<p>In the War Room, view an incident in widget format, for example, as severity in a bar chart.</p> <p>For more information, see <a href="#">Add a widget in the War Room</a>.</p>

**TIP:**

We recommend the following to optimize performance.

- Keep widgets simple (no scripts)
- Set refresh times to greater than one minute
- Limit the number of widgets on a dashboard

#### 14.3.2 | Create a widget using the widget builder

##### Abstract

Create a widget in the Widgets Library in and then add the widget to a dashboard or report.

In the Widgets Library, you create a widget using the widget builder, which enables you to define and configure data, and preview how that widget appears. The widget builder allows you to create complex widgets, eliminating the need to write scripts or upload JSON files (although you have the option to do this). These complex widgets have the same capabilities as if you were creating a script-based widget.

##### Task 1. Create a new widget

In the Widgets Library of the report or dashboard you are creating or editing, click  and select the widget type as follows.

Widget Type	Description
Incidents	Use incident data to create widgets related to incidents, for example timestamps, duration, incident types, and any incident field.
Indicators	Use indicator data to create widgets related to indicators, for example timestamps, indicator types, and any indicator field.
SOAR Metrics	Use SOAR metrics data to create widgets related to scripts, playbooks, and integrations, for example executions, durations, and errors.

Widget Type	Description
Tasks	<p>Use tasks data to create widgets related to investigation tasks, for example assignee, playbook name, and duration (manual or automated).</p> <p><b>NOTE:</b></p> <p>When creating a widget based on the results of an investigation task, only the following task types are supported for widget aggregation:</p> <ul style="list-style-type: none"> <li>• Manual tasks</li> <li>• Tasks that have an assignee</li> <li>• Tasks that have a due date</li> <li>• Tasks that are in an error state</li> <li>• Oversized tasks</li> </ul>
Scripts	<p>Use a script to create a widget. Although you can create complex widgets using the widget builder, you can also create dynamic widgets using scripts, such as calculating the percentage of incidents that DBot closed. The script can also pull information from the Cortex XSOAR API.</p> <p><b>NOTE:</b></p> <p>Before creating a script based widget, you need to create a script in the Scripts page and then select the script in the widget builder. The script must have the <code>widget</code> tag assigned, otherwise it does not appear when selecting the script in the widget builder.</p> <p>In the widget builder, you cannot manipulate data (no data appears in the Operations tab). However, you can define script arguments and change the color, layout, and legends.</p> <p>For more information, see <a href="#">Create a custom widget using a script</a>.</p>
Threat Intel Reports	<p>Use threat intel data to create widgets related to threat intel reports that have been created, for example reports by type and status.</p>
Upload	<p>Upload a JSON file to create a static widget which displays basic information, such as grouping incidents severity by type and active incidents by type.</p>

#### Task 2. Define the widget data

In the Query step, set the following information:

Parameter	Description																		
Widget display format	<p>Select one of the widget format icons. You can see a preview of how the widget appears.</p> <table border="1" data-bbox="747 323 1464 1410"> <thead> <tr> <th data-bbox="747 323 952 406">Widget Format</th><th data-bbox="952 323 1464 406">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="747 406 952 563"></td><td data-bbox="952 406 1464 563">View data in a timer format. For example, mean time to assignment. In the Visuals tab, you can select the threshold color.</td></tr> <tr> <td data-bbox="747 563 952 676"></td><td data-bbox="952 563 1464 676">View data in a number format. In the Visuals tab, you can select the threshold color.</td></tr> <tr> <td data-bbox="747 676 952 788"></td><td data-bbox="952 676 1464 788">View data in a bar format.</td></tr> <tr> <td data-bbox="747 788 952 893"></td><td data-bbox="952 788 1464 893">View data in a column format.</td></tr> <tr> <td data-bbox="747 893 952 1006"></td><td data-bbox="952 893 1464 1006">View data in a pie format.</td></tr> <tr> <td data-bbox="747 1006 952 1089"></td><td data-bbox="952 1006 1464 1089">View data in a line graph format.</td></tr> <tr> <td data-bbox="747 1089 952 1230"></td><td data-bbox="952 1089 1464 1230">View data in a table format. Click the gear icon to edit columns.</td></tr> <tr> <td data-bbox="747 1230 952 1410"></td><td data-bbox="952 1230 1464 1410">View data in a text format, which can be used as a text summary of the displayed data. You can use {0} to display a query value and {date} to display the date. Markdown is supported.</td></tr> </tbody> </table>	Widget Format	Description		View data in a timer format. For example, mean time to assignment. In the Visuals tab, you can select the threshold color.		View data in a number format. In the Visuals tab, you can select the threshold color.		View data in a bar format.		View data in a column format.		View data in a pie format.		View data in a line graph format.		View data in a table format. Click the gear icon to edit columns.		View data in a text format, which can be used as a text summary of the displayed data. You can use {0} to display a query value and {date} to display the date. Markdown is supported.
Widget Format	Description																		
	View data in a timer format. For example, mean time to assignment. In the Visuals tab, you can select the threshold color.																		
	View data in a number format. In the Visuals tab, you can select the threshold color.																		
	View data in a bar format.																		
	View data in a column format.																		
	View data in a pie format.																		
	View data in a line graph format.																		
	View data in a table format. Click the gear icon to edit columns.																		
	View data in a text format, which can be used as a text summary of the displayed data. You can use {0} to display a query value and {date} to display the date. Markdown is supported.																		

Parameter	Description												
Data source	<p>Select the source data to query.</p> <p>Cortex XSOAR retrieves data relevant for that data source. For example, for Incidents, in the Group by field all data relating to incidents is retrieved, such as type, owner, and created by.</p> <table border="1" data-bbox="747 406 1469 1567"> <thead> <tr> <th data-bbox="747 406 969 485">Widget Data Source</th><th data-bbox="969 406 1469 485">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="747 485 969 631">Incidents</td><td data-bbox="969 485 1469 631">Use incident data to create widgets related to incidents, for example timestamps, duration, incident types, and any incident field.</td></tr> <tr> <td data-bbox="747 631 969 777">Indicators</td><td data-bbox="969 631 1469 777">Use indicator data to create widgets related to indicators, for example timestamps, indicator types, and any indicator field.</td></tr> <tr> <td data-bbox="747 777 969 923">SOAR Metrics</td><td data-bbox="969 777 1469 923">Use SOAR metrics data to create widgets related to scripts, playbooks, and integrations, for example executions, durations, and errors.</td></tr> <tr> <td data-bbox="747 923 969 1069">War Room Entries</td><td data-bbox="969 923 1469 1069">Use War Room entry data to create widgets, for example number of entries according to owner.</td></tr> <tr> <td data-bbox="747 1069 969 1567">Tasks</td><td data-bbox="969 1069 1469 1567"> <p>Use tasks data to create widgets related to investigation tasks, for example assignee, playbook name, and duration (manual or automated).</p> <p><b>NOTE:</b></p> <p>When creating a widget based on the results of an investigation task, only the following task types are supported for widget aggregation:</p> <ul style="list-style-type: none"> <li>• Manual tasks</li> <li>• Tasks that have an assignee</li> <li>• Tasks that have a due date</li> <li>• Tasks that are in an error state</li> <li>• Oversized tasks</li> </ul> </td></tr> </tbody> </table>	Widget Data Source	Description	Incidents	Use incident data to create widgets related to incidents, for example timestamps, duration, incident types, and any incident field.	Indicators	Use indicator data to create widgets related to indicators, for example timestamps, indicator types, and any indicator field.	SOAR Metrics	Use SOAR metrics data to create widgets related to scripts, playbooks, and integrations, for example executions, durations, and errors.	War Room Entries	Use War Room entry data to create widgets, for example number of entries according to owner.	Tasks	<p>Use tasks data to create widgets related to investigation tasks, for example assignee, playbook name, and duration (manual or automated).</p> <p><b>NOTE:</b></p> <p>When creating a widget based on the results of an investigation task, only the following task types are supported for widget aggregation:</p> <ul style="list-style-type: none"> <li>• Manual tasks</li> <li>• Tasks that have an assignee</li> <li>• Tasks that have a due date</li> <li>• Tasks that are in an error state</li> <li>• Oversized tasks</li> </ul>
Widget Data Source	Description												
Incidents	Use incident data to create widgets related to incidents, for example timestamps, duration, incident types, and any incident field.												
Indicators	Use indicator data to create widgets related to indicators, for example timestamps, indicator types, and any indicator field.												
SOAR Metrics	Use SOAR metrics data to create widgets related to scripts, playbooks, and integrations, for example executions, durations, and errors.												
War Room Entries	Use War Room entry data to create widgets, for example number of entries according to owner.												
Tasks	<p>Use tasks data to create widgets related to investigation tasks, for example assignee, playbook name, and duration (manual or automated).</p> <p><b>NOTE:</b></p> <p>When creating a widget based on the results of an investigation task, only the following task types are supported for widget aggregation:</p> <ul style="list-style-type: none"> <li>• Manual tasks</li> <li>• Tasks that have an assignee</li> <li>• Tasks that have a due date</li> <li>• Tasks that are in an error state</li> <li>• Oversized tasks</li> </ul>												

Parameter	Description	
	Widget Data Source	Description
	Scripts	<p>Use a script to create a widget. Although you can create complex widgets using the widget builder, you can also create dynamic widgets using scripts, such as calculating the percentage of incidents that DBot closed. The script can also pull information from the Cortex XSOAR API.</p> <p><b>NOTE:</b></p> <p>Before creating a script based widget, you need to create a script in the Scripts page and then select the script in the widget builder. The script must have the <code>widget</code> tag assigned, otherwise it does not appear when selecting the script in the widget builder.</p> <p>In the widget builder, you cannot manipulate data (no data appears in the Operations tab). However, you can define script arguments and change the color, layout, and legends.</p> <p>For more information, see <a href="#">Create a custom widget using a script</a>.</p>
	Threat Intel Reports	<p>Use threat intel data to create widgets related to threat intel reports that have been created, for example reports by type and status.</p>
Query		<p>Queries data in the Lucene query syntax form relating to the data source.</p> <p>For example when the data source is incidents and the query is: - <code>status:closed and owner:""</code>, it queries all incidents that are not closed which do not have an owner.</p> <p>Or to see all incidents that are not closed, not archived, and are not jobs, use the query: <code>-status:closed and -status:archived and -category:job</code>.</p>
Date range		The time frame to retrieve data.
Widget name		Type a meaningful name for the widget.

### Task 3. Configure the widget data

This step enables data manipulation, similar to scripting. You can configure the data according to groups and fields (including custom calculations on fields).

- (Not relevant for tables or text) Click the Operations step, and in the Values section select one of the following calculations to perform on the data (not relevant for Script and War Room Entries data sources).

Calculation	Description
Count	Counts the total value of the field. For example, display the total number of incidents in your system. You can then group by type and severity.

Calculation	Description
Average	Calculates the average value of the field. For example, display the average number of incidents in your system over the selected time frame. You can then group by type and severity.
Sum	Counts the value of the field according to a specific value. For example, when you define a metrics widget type, select the execution count, total duration, errors count, or create your own custom calculations.
Min	Calculates the minimum numeric value of the data. For example, you may want to see the minimum number of fetched events.
Max	Calculates the maximum numeric value of the data. For example, you may want to see the maximum number of fetched events.

2. (Not relevant for Count) Select one of the fields from the dropdown or create your own custom calculations by selecting Custom calculations on fields.

3. If adding custom calculations, type the calculation as required.

The custom calculation modal suggests incident fields based on the widget data type, which are automatically validated. You can add your own fields (provided these fields exist), according to the widget data type, by using the CLI name. These fields are not validated.

You can add mathematical operators (such as +, -, /, \*) between fields. Variables using {} are also supported. For example:

- To see the average time that incidents are late, type `{now}-remediationsla.dueDate`.
- To calculate the average time between detection and remediation for phishing incidents (in the phishing generic playbook we set the time detection and remediation SLA timers), type `remediationsla.startDate-detectionsla.startDate`.
- To see remediations (less 10 minutes), type `remediationsla.dueDate-10`.

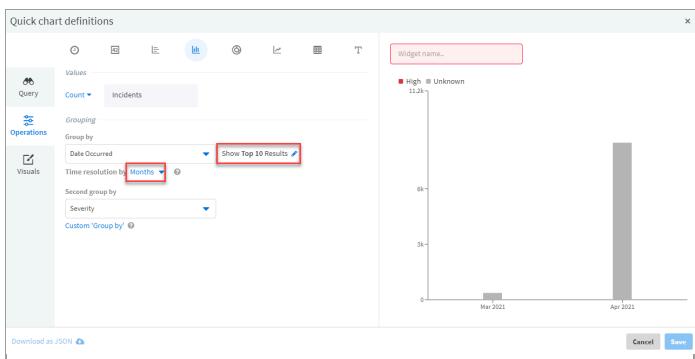
4. In the Axis and grouping section Group by field, from the dropdown, select the group you want to add.

By default, the results are limited to the top 10 most popular results. If you want to change the top most popular to the least popular, change the number, or you want to see the remaining results that are not covered in one group (the Show 'Others' checkbox), click the edit button and update as required.

If you want to add a custom field, ensure the Make data available for search incident type field is checked when editing or creating a new field.

#### Example 21. Limit the number of results

You can limit the amount of results to return, view the most or least popular, and for some fields select the time format. For example, you may want to see the top 10 most popular active incidents active incidents by month.



5. (Optional) Define custom groups (for example, define specific owners in the owner group).

1. Click Custom 'Group by'.
2. In the Create Custom groups window, click Equals (String) to change the operator.
3. Select a value from the dropdown.
4. Change the name as required.
5. If you want to create a second group, click Add custom group.

6. If you want to add a group for all other values that have not been defined, click the Create and display a group for all remaining values checkbox.

#### Example 22. Group data into two teams

You can manipulate data according to one or two groups (two groups are useful for vertical bars and line charts). Within each group, you can group by a bucket. For example, for two teams - Team A and Team B, each one is made up with different team members. You only want to see Team A and Team B and not the individual team members.

6. In the Second group by field, add the group as required. For example, to see data filtered by owner and severity, select Group By Owner and Second Group by Severity.

#### Task 4. Define the widget display

1. Click the Visuals step and define how the widget appears.

Parameter	Description
Axis name	The name of the axis for both horizontal and vertical.
Format	Select the format of the table for both horizontal and vertical axis. For example, hours, minutes, days, weeks, etc.
Reference Line	Whether you want a line showing the average, minimum, maximum, or custom line.
Show Legend	Whether you want to see the legend in your widget.
Show also percentage	Displays the percentage when selecting a pie chart.
Show values on the graph	Add the values on the chart widget.
Display trend	Compares dates for a particular period in a number widget. For example, this week vs. last week, this year vs. last year, and so on. To change the comparison period, in the Time frame field from the dropdown, select the relevant date.
Widget color threshold	Select the Widget color threshold in a number or duration widget to highlight the threshold data and define the threshold by selecting the Widget color threshold checkbox. For example, if less than 150 red, 100 yellow, 50 green. To add more thresholds, click Add new threshold. You can change the colors as required.

2. To change the color, in the preview section, hover next to the legend, click the ellipsis and then click Edit color.

### Task 5. Save and add the widget to a dashboard or report

- Click Save.

The widget is added to the widgets library.

- Add the widget to the dashboard or report.

When you add the widget, it automatically uses the date range of the dashboard or report. You can change it by clicking the settings icon and selecting Use widget's date range. To revert, click the settings icon again and select Use dashboard's date range.

#### 14.3.2.1 | Create a widget using the widget builder examples

##### Abstract

Widget use cases when creating a new widget.

Average time to close incidents

In this example we want to create a bar chart widget that shows the following:

- The average time it takes to close incidents per day
- Classified according to incident types
- Incidents that occurred during the previous seven days

- Click the add + button from the Widgets Library.

- Select Incidents.

- Enter a name in the Widget name field.

- Click the Bar graph icon.

- In the Query tab, define the following:

Data source: Incidents

Query: `-category:job and -status:Closed`

Date range: Last 7 days

- In the Operations tab:

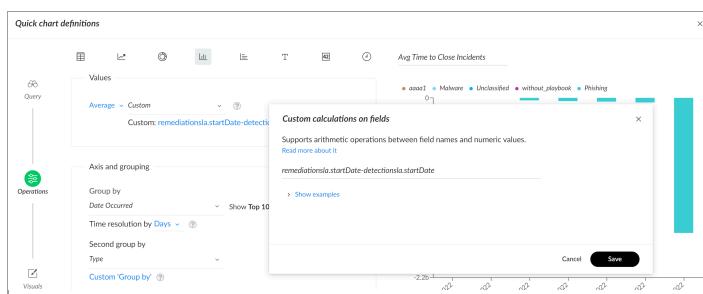
Change Count to Average.

From the dropdown list, select Custom calculations on fields.

Type `remediationsla.startDate-detectionsla.startDate`

Group by: Date Occurred

Second Group by: Type



How many incidents over the last seven days

In this example, we want to view the following data:

- How many incidents occurred in the last 7 days
- Closed vs not closed (pending or active)
- Line chart

- Click the add + button from the Widgets Library.

2. Select Incidents.
3. Enter a name in the Widget name field.
4. Click the Line graph icon.
5. In the Query tab, define the following:

Data source: Incidents

Query: **-category:job**

Date range: Last 30 days

6. In the Operations tab, the first group is **Date Occurred**.
7. In the second group, from the dropdown list, select **status**.
8. Click Custom Group by to add the following data:

**Create custom groups (Status)**

Group name: Not Closed Remove group

[Status] Equals (String) active + -

AND

[Status] Equals (String) pending + -

+ AND

End of group

Group name: Closed Remove group

[Status] Equals (String) closed + -

+ AND

End of group

+ Add custom group

Remaining values

Create and display a group for all remaining values

Cancel Save

Average time for open incidents that are late

In this example, we want to create the following incident type widget:

- The average time for open incidents that are late
- Grouped by 2 groups (group A and group B) and by type
- In a bar chart

1. Click the add + button from the Widgets Library.

2. Select Incidents.

3. Enter a name in the Widget name field.

4. Click the Bar graph icon.

5. In the Query tab, define the following:

Data source: Incidents

Query: **-status:Closed and category:job**

Date range: Last 30 days

6. In the Operations tab, add the following information:

a. In the Values section, select Average.

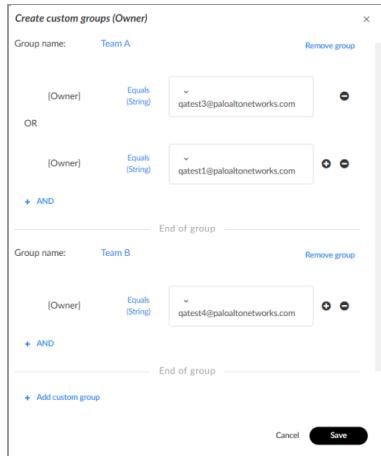
b. From the dropdown list, click Custom calculations on fields.

c. Type **{now}-remediationsla.dueDate**.

We want to see the average time that incidents are late (from today's date). We add a variable **{now}**, so that we do not have to change the date.

d. In the Group by field, select Owner and then click Custom Group by.

e. Add the following, using users from your organization.



f. In the Second group by field, from the dropdown list, select Type.

g. Select the checkbox for Create and display a group for all remaining values and then click Save.

7. In the Visuals tab, select the following:

a. Horizontal options - Axis name: TEAM.

b. Vertical options - Axis name: REMEDIATION TIME.

#### 14.3.3 | Create a custom widget using a JSON file

##### Abstract

Create a custom widget using a JSON file for reports and dashboard in Cortex XSOAR.

You can create a custom widget for your dashboard or report using a JSON file and then add the new widget to a new or edited dashboard or report. If you want to create more complicated widgets using scripts, see Create a custom widget using a script.

1. Create a JSON file, and add the relevant JSON file widget parameters.

List of JSON file widget parameters

Parameter	Description
<b>id</b>	The unique identifier for the widget.
<b>name</b>	The display name of the widget.

Parameter	Description
<code>dataType</code>	<p>The data source of the widget. Must be one of the following:</p> <ul style="list-style-type: none"> <li>• <code>incidents</code></li> <li>• <code>indicators</code></li> <li>• <code>messages</code></li> <li>• <code>entries</code></li> <li>• <code>scripts</code></li> </ul> <p>Relevant only when you are creating a script.</p> <ul style="list-style-type: none"> <li>• <code>tasks</code></li> <li>• <code>generics</code></li> </ul> <p>Relevant when creating Threat Intel reports. When used, the <code>definitionId</code> value must be <code>ThreatIntelReport</code>.</p>
<code>query</code>	<p>Queries query data in the Lucene query syntax form relating to the <code>dataType</code>. For example when <code>dataType</code> is incidents and the query is: <code>-status:closed and owner:""</code>, it queries all incidents that are not closed, which does not have an owner.</p> <p>For script based widgets, the query is the name of the script.</p>
<code>sort</code>	<p>Sorts the data, when displaying the <code>widgetType</code> (applies to table and list widget types) as a list of objects, which consists of the following:</p> <ul style="list-style-type: none"> <li>• <code>field</code>: The field name for which to sort.</li> <li>• <code>asc</code>: Whether to sort data in ascending values. If true, the order is in ascending value.</li> </ul>
<code>widgetType</code>	<p>The type of widget you want to create. Must be one of the following:</p> <ul style="list-style-type: none"> <li>• <code>bar</code></li> <li>• <code>column</code></li> <li>• <code>pie</code></li> <li>• <code>number</code></li> <li>• <code>line</code></li> <li>• <code>table</code></li> <li>• <code>trend</code></li> <li>• <code>list</code></li> <li>• <code>duration</code></li> <li>• <code>image</code></li> </ul>
<code>size</code>	<p>The maximum number of returning elements. Use <code>0</code> for the <code>widgetType</code>'s default.</p> <p><b>NOTE:</b></p> <ul style="list-style-type: none"> <li>• <b>Table/List</b>: Default is up to 13</li> <li>• <b>Chart</b>: Default is up to 10.</li> <li>• <b>Number and Trend</b>: Ignores the size value.</li> </ul>

Parameter	Description
category	Adds a category name. The widget appears under a category instead of being classified by <code>dataType</code> .
dataRange	The time period for which to return data. The time period is overridden by the dashboard or report time period. Default is all times. <ul style="list-style-type: none"> <li>• <code>fromDate</code>: The start date from which to return data in the format: "YYYY-MM-DDTHH:MM:SSZ". For example, "2019-01-01T16:30:00Z".</li> <li>• <code>toDate</code>: The end date for which to return data in the format: "YYYY-MM-DDTHH:MM:SSZ". For example, "2019-01-01T16:30:00Z".</li> <li>• <code>period</code>: An object describing a period of relative time. If using the <code>fromDate/toDate</code> parameters, this parameter is ignored. <ul style="list-style-type: none"> <li>◦ <code>byTo</code>: The to period unit of measurement. Values are 'minutes', 'hours', 'days', 'weeks', 'months'.</li> <li>◦ <code>byFrom</code>: The from period unit of measurement. Values are: 'hours', 'days', 'weeks', 'months'.</li> <li>◦ <code>toValue</code>: The duration of the to period. Integer.</li> <li>◦ <code>fromValue</code>: The duration of the from period. Integer. For example, last 7 days - { <code>byFrom: 'days'</code>, <code>fromValue: 7</code> }.</li> </ul> </li> </ul>
description	The description of the widget in the Widget Library.
params	Enriches the widget with specific parameters, mainly based on the <code>widgetType</code> . Includes the following: <ul style="list-style-type: none"> <li>• <code>groupBy</code>: An array of field names for which to group the returned values. Used when widget type is bar, column, line or pie. For example, [ "type", "owner" ]: Groups results by type and owner, and returns a nested result for each type with statistics according to the owner.</li> </ul> <p><b>NOTE:</b></p> <p>Bar/column charts defined with two groups can become stacked.</p> <ul style="list-style-type: none"> <li>• <code>hideLegend</code>: Shows or hides the legend, if it exists. Default is false.</li> <li>• <code>keys</code>: An array that enables processing the data value and modifies it by the given list of keys. For example, [ "avg openDuration / (3600*24)" ] process for each group found in the result, the average open duration (in days).</li> <li>• <code>text</code>: The markdown text for text widgets or image data for image widgets. For example, if you want the widgets to appear on separate pages in a report, use [ "\\\pagebreak" ].</li> <li>• <code>timeFrame</code>: Supplies the custom time frame for which the widget scales. Values are "years", "months", "days", "hours", "minutes". The default is "days".</li> <li>• <code>tableColumns</code>: Enables you to define the name of the columns in a list or table. For example, "[ { "key": "name" }, { "key": "mycustomfield" } ]": Displays the name and a custom field.</li> </ul>
legend	An array of objects that consists of a name and color. The name must match a group name. The color can be the name of the color, the hexadecimal representation of the color, or the rgb color value.

2. Create or edit a dashboard or report.

3. In the Widget Library section, click Upload.

4. Select the JSON file you created in step 1 and click Open.

5. To add the widget to the dashboard or report, click Add.

## JSON file widget examples

### Display incident severity by type

The JSON file to display incident severity by type contains the following:

- Bar chart
- Incidents from the last 30 days
- Grouped by severity and for each severity display the nested group size (count of incidents displayed by the length of the bar) colored according to type.

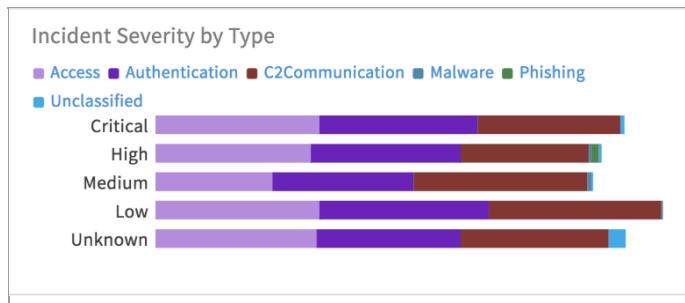
Create the following JSON file:

```
{
  "name": "Incident Severity by Type",
  "dataType": "incidents",
  "widgetType": "bar",
  "query": "-category:job and -status:archived and -status:closed",
  "dateRange": {
    "period": {
      "byFrom": "days",
      "fromValue": 30
    }
  },
  "params": {
    "groupBy": [
      "severity",
      "type"
    ]
  }
}
```

You can see the following parameters:

- The Widget is called **Incident Severity by type**.
- The data type is **incidents**.
- The widget type is **bar**.
- The **query** specifies that you do not want to return incidents that are categorized as job nor incidents that are archived and closed.
- For the date range, the **fromValue** sets the widget to display the last 30 units of time. The **byFrom** sets the units of time to days, which results in the last 30 days.
- The **params** parameter is set with a **groupBy** value marking the first group by severity name and then by type (making the bar chart stacked).

After you import the widget into the **Widget Library** the following widget appears:



You can see the incidents are grouped by severity and the number of incidents are displayed by the length of the bar, which are colored according to type.

### Display incidents by type

The JSON file to display incidents by type contains the following:

- Vertical bar chart
- Incidents from the last 7 days
- Grouped by date and type and sorted by date occurred

```
{
  "dataType": "incidents",
  "widgetType": "column",
  "params": {
    "groupBy": [
      "occurred(d)",
      "type"
    ],
  }
}
```

```

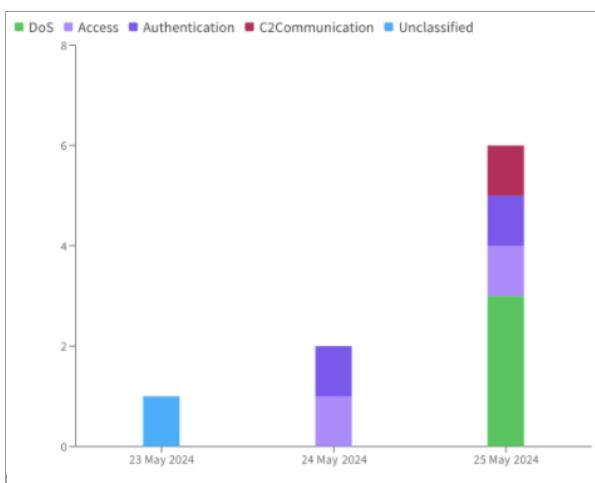
    "valuesFormat": "abbreviated",
    "timeFrame": "days"
},
"dateRange": {
    "period": {
        "byFrom": "days",
        "fromValue": 7
    }
},
"propagationLabels": [
    "all"
],
"customCalculation": {
    "operation": "count",
    "fieldName": "",
    "expression": ""
},
"name": "Change Sort Order In Column Chart - Sort by Date",
"sort": [{ "field": "occurred", "asc": true }]
}
}

```

You can see the following parameters:

- The Widget is called **Change Sort Order In Column Chart - Sort by Date**.
- The data type is **incidents**.
- The widget type is **column**.
- For the date range, the **fromValue** sets the widget to display the last 7 units of time. The **byFrom** sets the units of time to days, which results in the last 7 days.
- The **params** parameter is set with a **groupBy** value marking the first group by occurrence date and then by type (making the column chart stacked).

After you import the widget into the Widget Library the following widget appears:



#### 14.3.4 | Create a custom widget using a script

##### Abstract

Create a custom script based widget in using a script. Use custom widgets in dashboards and reports.

You can use scripts in custom widgets to create dynamic widgets for more complex calculations. For examples of creating widgets using scripts, see Script-based widget examples.

##### **NOTE:**

Cortex XSOAR supports JavaScript, Python and PowerShell.

Before creating a script-based widget in the Widgets Library, you need to create or upload the script to the Scripts page. In the Widgets Library, you can define the script arguments and change the visuals.

##### **NOTE:**

If you upload a script to the Scripts page, the Arguments field is automatically updated. You can then define the arguments in the widget builder. If you create a new script (without uploading) in the Scripts page, you need to add the arguments manually for them to appear in the Widgets Library when creating or editing a widget.

1. In the Scripts page, upload or create a new script for one of the following widget types:

- Text
- Number
- Duration
- Trend
- Chart
- Table or list
- Filter Data for all Widgets (Pivoting)

2. In the Widgets Library, create a widget. For more information about creating a widget, see Create a widget using the widget builder.

3. Select the Scripts data type and then add the script to the widget.

(Upload script only) If you have added arguments, these appear when creating a widget. If you have not uploaded the script, you need to add the arguments manually in the Scripts page.

4. Add the script-based widget where relevant, for example to a report, a dashboard, or an incident.

#### 14.3.4.1 | Script-based widget examples

##### Abstract

Create script based widgets based on automation scripts for reports and dashboards in Cortex XSOAR.

The following are sample arguments/scripts to create a widget. After creating the widget from a script, add the widget to a dashboard or report. For more details, see Create a widget using the widget builder.

##### Script argument examples

To add a time stamp or a use a search query, add the following arguments to a script.

Argument	Description
<code>demisto.args()['from']</code>	The start date of the time-stamp date range of the widget.
<code>demisto.args()['to']</code>	The end date of the time-stamp date range of the widget.
<code>demisto.args()['searchQuery']</code>	The search query entered into the search bar at the top of the dashboard.

##### Text

In this example, create a script that queries and returns current on-line users, and displays the data in a markdown table.

In the script, type one of the following return values:

##### JavaScript

```
return executeCommand("getUsers", {online: true})[0].HumanReadable;
```

##### Python

```
demisto.results(demisto.executeCommand("getUsers", { "online": True })[0]["HumanReadable"])
```

When creating or editing the widget in Cortex XSOAR, to add a page break, type /pagebreak in the text box. When you generate a report, the widgets that follow the page break are on a separate page.

The screenshot shows the 'Quick chart definitions' interface. On the left, there's a sidebar with 'Operations' and 'Visuals' sections. The main area has a 'The query' section with a 'Data source' dropdown set to 'Incidents', a 'Query' input field with placeholder 'Start typing for autocomplete...', and a 'Date range' dropdown set to 'Last 7 days'. To the right is a preview window titled 'Active Incidents' containing a single row with the text 'Pagebreak'. At the bottom, there are 'Download as JSON' and 'Save' buttons.

In the dashboard, the following widget displays the on-line users:

Username	Email	Name	Phone	Roles
@demisto.com	@demisto.com	[REDACTED]		demisto: [Odata-Administrator]
@demisto.com	@demisto.com	[REDACTED]		demisto: [Odata-Administrator]
@demisto.com	@demisto.com	[REDACTED]		demisto: [Odata-Administrator]
@demisto.com	@demisto.com	[REDACTED]		demisto: [Odata-Administrator]

#### NOTE:

(*Multi-tenant*) Script-based text widgets are not supported in the Main Account.

#### Number

This example shows how to create a single item widget with the percentage of incidents that DBot closed.

In the script, type one of the following:

#### JavaScript

```
var res = executeCommand("getIncidents", {
  'query': 'status:closed and investigation.users:""',
  'fromdate': args.from,
  'todate': args.to,
  'size': 0
});
var closedByDbot = res[0].Contents.total;

res = executeCommand("getIncidents", {
  'status': 'closed',
  'fromdate': args.from,
  'todate': args.to,
  'size': 0 });
var overallClosed = res[0].Contents.total;

var result = Math.round(closedByDbot * 100 / overallClosed);
return isNaN(result) ? 0 : result;
```

#### Python

```
res = demisto.executeCommand("getIncidents", {
  "query": "status:closed and investigation.users:\\"\\\"",
  "fromdate": demisto.args()["from"],
  "todate": demisto.args()["to"],
  "size": 0
})
closedByDbot = res[0]["Contents"]["total"]

res = demisto.executeCommand("getIncidents", {
  "status": "closed",
  "fromdate": demisto.args()["from"],
  "todate": demisto.args()["to"],
  "size": 0
})
overallClosed = res[0]["Contents"]["total"]
if overallClosed == 0:
  demisto.results(0)
else:
  result = round(closedByDbot * 100 / overallClosed)
  demisto.results(result)
```

#### Duration

In this example, create a script that queries and returns a time duration (specified in seconds), and displays the data as a countdown clock. If using a JSON file, you must set `widgetType` to duration.

In the script, type one of the following return values:

**JavaScript**

```
return JSON.stringify([{"name": "", "data": [120]}]);
```

**Python**

```
demisto.results('[{"name": "", "data": [120]}]')
```

The return type should be a string (any name) and an integer. The time is displayed in seconds.

After you have uploaded the script and created the widget, you can add the widget to the dashboard or report. The  widget displays the time duration:

**Chart**

A valid result for a chart widget is a list of groups. Each group points to a single entity, for example, in bar charts each group is a bar. A group consists of the following:

- **Name** - A string.
- **Data** - An array of integers.
- **Color** - A string representing a color that will be used as a default color for that group. It can be the name of the color, a hexadecimal representation of the color, or an rgb color value (optional).

**NOTE:**

A widget legend color will override a group color if it exists.

- **Groups** - A nested list of groups (optional).

In this example, we show how to create a script that will query and return the trend between two sums in a pie chart.

- Pie
- Line
- Bar
- Column

**Simple pie/chart**

In the script, type the following return value:

**JavaScript**

```
var data = [
  {"name": "2018-04-12", "data": [10], "color": "blue"},
  {"name": "2018-04-10", "data": [3], "color": "#029be5"},
  {"name": "2018-04-17", "data": [1], "color": "rgb(174, 20, 87)"}, 
  {"name": "2018-04-16", "data": [34], "color": "grey"}, 
  {"name": "2018-04-15", "data": [17], "color": "purple"}];
return JSON.stringify(data);
```

**Python**

```
data = [
  {"name": "2018-04-12", "data": [10], "color": "blue"},
  {"name": "2018-04-10", "data": [3], "color": "#029be5"},
  {"name": "2018-04-17", "data": [1], "color": "rgb(174, 20, 87)"}, 
  {"name": "2018-04-16", "data": [34], "color": "grey"}, 
  {"name": "2018-04-15", "data": [17], "color": "purple"}]
demisto.results(json.dumps(data))
```

After you have uploaded the script and created the widget you can add the widget to a dashboard or report.

**Two group chart****JavaScript**

```
var data = [
  {"name": "2018-04-12", "data": [10], "groups": [{"name": "Unclassified", "data": [10]}]}, 
  {"name": "2018-04-10", "data": [3], "groups": [{"name": "Unclassified", "data": [2]}, {"name": "Access", "data": [1]}]}, 
  {"name": "2018-04-17", "data": [1], "groups": [{"name": "Unclassified", "data": [1]}]}, 
  {"name": "2018-04-16", "data": [34], "groups": [{"name": "Unclassified", "data": [18]}, {"name": "Phishing", "data": [14]}]}, 
  {"name": "2018-04-15", "data": [17], "groups": [{"name": "Access", "data": [17]}]}]
```

```
];
return JSON.stringify(data);
```

**Python**

```
data = [
    {"name": "2018-04-12", "data": [10], "groups": [{"name": "Unclassified", "data": [10]}]},
    {"name": "2018-04-10", "data": [3], "groups": [{"name": "Unclassified", "data": [2]}, {"name": "Access", "data": [1]}]},
    {"name": "2018-04-17", "data": [1], "groups": [{"name": "Unclassified", "data": [1]}]},
    {"name": "2018-04-16", "data": [34], "groups": [{"name": "Unclassified", "data": [18]}, {"name": "Phishing", "data": [14]}]},
    {"name": "2018-04-15", "data": [17], "groups": [{"name": "Access", "data": [17]}]}
]
demisto.results(json.dumps(data))
```

**Trend**

In this example, create a script that queries and returns the trend between two sums.

In the script, type one of the following return values:

**JavaScript**

```
return JSON.stringify({currSum: 48, prevSum: 32});
```

**Python**

```
demisto.results({ "currSum": 48, "prevSum": 32 })
```

The return displays an object which compares the current sum with the previous sum.

**Table or list**

In this example, you need to create a script that queries and returns employee information in a table. For Table or List, if creating a JSON file, set the widgetType to table or list. When using lists, a maximum of two columns displays, the rest are ignored (do not display).

In the script, type one of the following return values:

**JavaScript**

```
return JSON.stringify({total: 3, data:[
    {Employee: 'David D', Phone: '+14081234567', Email: 'David@org.com'},
    {Employee: 'James J', Phone: '+14087654321', Email: 'James@org.com'},
    {Employee: 'Alex A', Phone: '+14087777777', Email: 'Alex@org.com'}
]});
```

**Python**

```
demisto.results({ "total": 3, "data": [{"Employee": "David D",
    "Phone": "+14081234567", "Email": "David@org.com"}, {"Employee": "James J",
    "Phone": "+14087654321", "Email": "James@org.com"}, {"Employee": "Alex A",
    "Phone": "+14087777777", "Email": "Alex@org.com"}]})
```

After you have uploaded the script and created a widget you can add the widget to a dashboard or report. The following widget displays the employee information:

Employee	Email	Phone
David D	David@org.com	+14081234567
James J	James@org.com	+14087654321
Alex A	Alex@org.com	+14087777777

Filter data for all widgets (pivoting)

Example 23. Display filtered incident and indicator data in a widget with a bar graph

In this example, you create a filter according to type (phishing, access and IP) and then pivot to the relevant incident/indicators page. You need to add the following to the JSON or python script.

- **dataType**: Pivots to the relevant page, such as Incidents page.
- **query**: Filters according to the value in the relevant page. For example, for phishing, if you define ‘type:Phishing’ and the **dataType:incidents**, you are taken to the Incident page with the ‘type:Phishing’ filter.
- **pivot**: Filters the dashboard according to data set. For example, **pivot: “type:Phishing”** enables you to filter data that relates to phishing in the dashboard.

In the script, type one of the following return values:

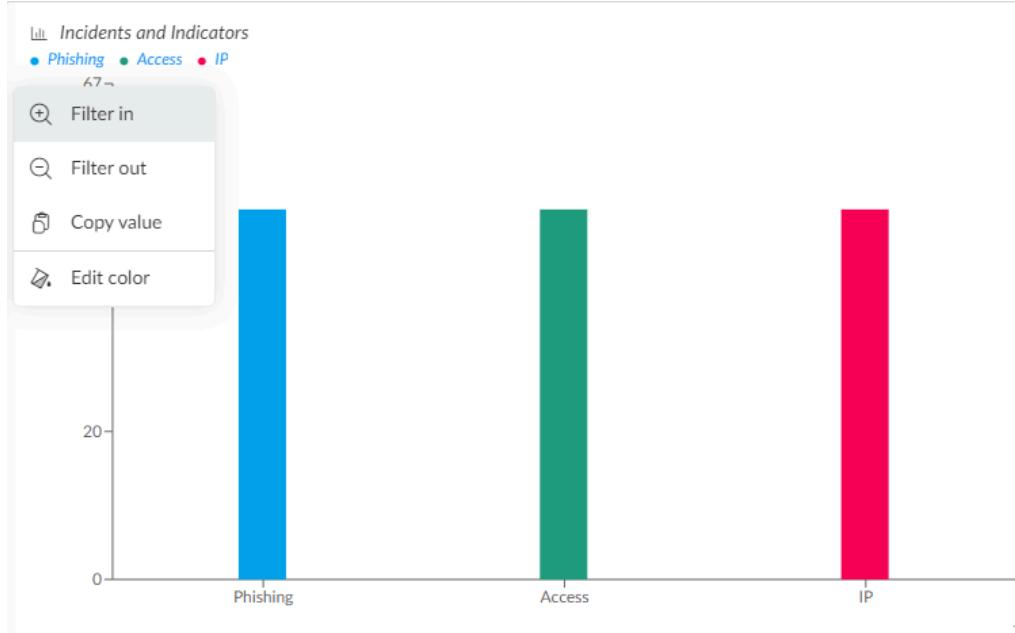
**JavaScript**

```
return JSON.stringify([{"name": "Phishing", "dataType": "incidents", "query": "type:Phishing", "data": [50], "pivot": "type:Phishing"}, {"name": "Access", "dataType": "incidents", "query": "type:Access", "data": [50], "pivot": "type:Access"}, {"name": "IP", "data": [50], "dataType": "indicators", "query": "type:IP", "pivot": "type:IP"}]);
```

**Python**

```
data = [
    {"name": "Phishing", "data": [50], "dataType": "incidents", "Query": "type:Phishing", "pivot": "type:Phishing"}, 
    {"name": "Access", "data": [50], "dataType": "incidents", "query": "type:Access", "pivot": "type:Access"}, 
    {"name": "IP", "data": [50], "dataType": "indicators", "query": "type:IP", "pivot": "type:IP"}
]
demisto.results(json.dumps(data))
```

After you upload the script and created a widget, add the widget to a dashboard or report page.



Example 24. Display filtered incident and indicator data in a widget with a line graph

In this example, you create a filter according to type (phishing, access and IP) and then pivot to the relevant incident/indicators page. You need to add the following to the JSON or python automation script.

**JavaScript**

```
return JSON.stringify([
    {
        "name": "Jan 1, 2024",
        "data": [6],
        "groups": [
            { "name": "Phishing", "data": [1], "pivot": "type:Phishing", "query": "type:Phishing" },
            { "name": "Access", "data": [2], "pivot": "type:Access", "query": "type:Access" },
            { "name": "IP", "data": [3], "pivot": "type:IP", "query": "type:IP" }
        ]
    },
    {
        "name": "Jan 2, 2024",
        "data": [7],
        "groups": [
            { "name": "Phishing", "data": [2], "pivot": "type:Phishing", "query": "type:Phishing" },
            { "name": "Access", "data": [1], "pivot": "type:Access", "query": "type:Access" },
            { "name": "IP", "data": [4], "pivot": "type:IP", "query": "type:IP" }
        ]
    },
    {
        "name": "Jan 3, 2024",
        "data": [8],
        "groups": [
            { "name": "Phishing", "data": [3], "pivot": "type:Phishing", "query": "type:Phishing" },
            { "name": "Access", "data": [4], "pivot": "type:Access", "query": "type:Access" },
            { "name": "IP", "data": [1], "pivot": "type:IP", "query": "type:IP" }
        ]
    }
]);
```

**Python**

```
data = [
    {
        "name": "Jan 1, 2024",
        "data": [6],
        "groups": [
            { "name": "Phishing", "data": [1], "pivot": "type:Phishing", "query": "type:Phishing" },
            { "name": "Access", "data": [2], "pivot": "type:Access", "query": "type:Access" },
            { "name": "IP", "data": [3], "pivot": "type:IP", "query": "type:IP" }
        ]
    }
];
```

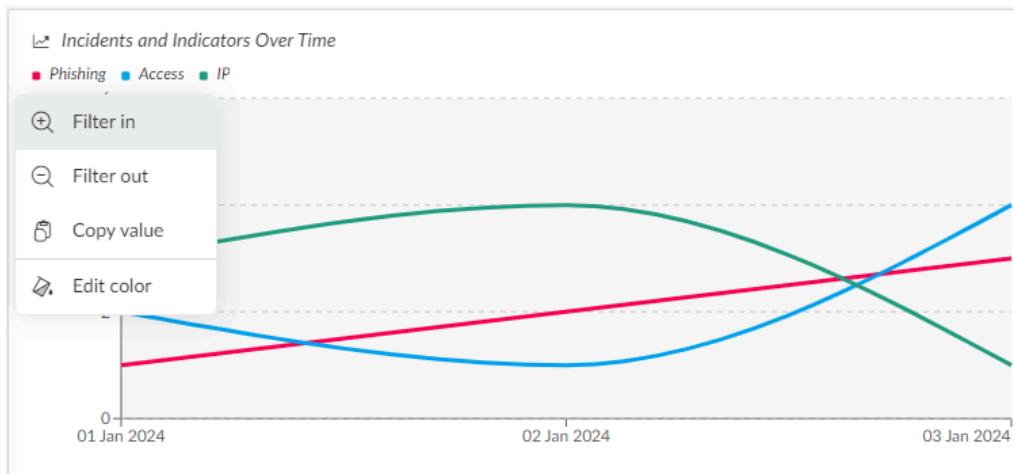
```

"groups": [
  { "name": "Phishing", "data": [1], "pivot": "type:Phishing", "query": "type:Phishing" },
  { "name": "Access", "data": [2], "pivot": "type:Access", "query": "type:Access" },
  { "name": "IP", "data": [3], "pivot": "type:IP", "query": "type:IP" }
]
},
{
  "name": "Jan 2, 2024",
  "data": [7],
  "groups": [
    { "name": "Phishing", "data": [2], "pivot": "type:Phishing", "query": "type:Phishing" },
    { "name": "Access", "data": [1], "pivot": "type:Access", "query": "type:Access" },
    { "name": "IP", "data": [4], "pivot": "type:IP", "query": "type:IP" }
  ]
},
{
  "name": "Jan 3, 2024",
  "data": [8],
  "groups": [
    { "name": "Phishing", "data": [3], "pivot": "type:Phishing", "query": "type:Phishing" },
    { "name": "Access", "data": [4], "pivot": "type:Access", "query": "type:Access" },
    { "name": "IP", "data": [1], "pivot": "type:IP", "query": "type:IP" }
  ]
}
]

demisto.results(json.dumps(data));

```

After you upload the script and create a widget, add the widget to a dashboard or report page.



#### 14.3.5 | Create a widget from an incident

##### Abstract

Create a custom widget from an incident search in Cortex XSOAR.

Although there are various out-of-the-box system widgets available, you can create custom widgets from incidents and then add them to a dashboard or report.

To create a widget from an incident, you need to run a query from the Incidents page and then save the visual results as a widget.

1. In the Incidents page, from the dropdown list select the date range.
2. In the Query field, type the query criteria as required and run the query.
3. Click .
4. Follow the procedure from Task 2. Define the widget data in Create a widget using the widget builder.
5. Click Save.

The widget is added to the Widgets Library.

##### **NOTE:**

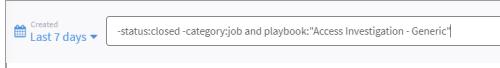
By default, the widget inherits the date range that you specify when creating the widget, but you can modify the date range when you create the dashboard or report. If the date range for the report or dashboard does not include the widget date range, the data is blank. To override the dashboard or report's date range, click Use Widget's date range.

#### Example 25. Create a widget from an incident example

In the following example, create a widget that contains:

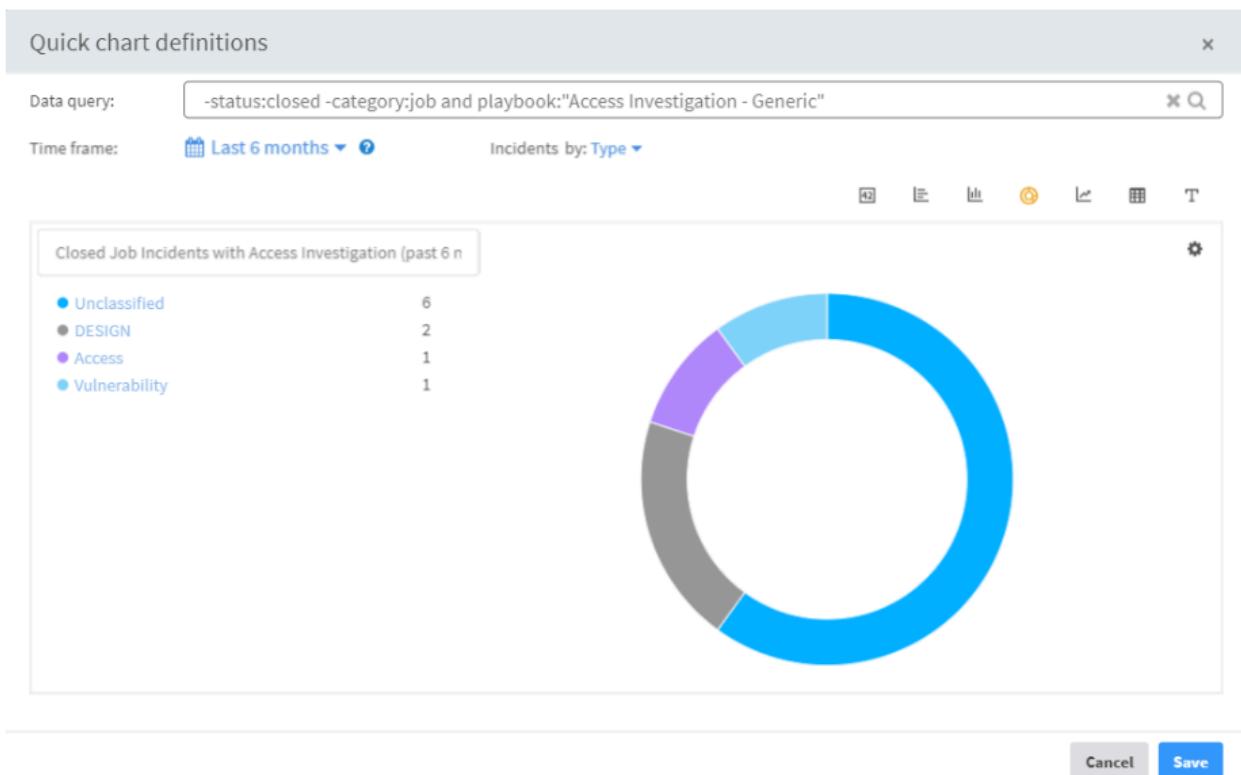
- Incidents created in the last 6 months
- Status: Every status other than closed
- Category: All categories other than jobs
- Use Access Investigation - Generic playbook

1. In the Incidents page, run the following query:



2. Click .

3. Type the name (Closed Job Incidents (past 6 months)) and save the query results as a widget:



4. Add/Edit a dashboard and locate the widget:

The screenshot shows the 'Widgets Library' interface with a search bar at the top containing the text 'closed'. Below the search bar, there is a section titled 'Incidents (3)'. This section contains a brief description: 'Stats about your Soc - IOCs, Incident types, general SLA etc.' and sorting options: 'Sort by ABC' and 'Show All'. There are three widgets listed:

- Closed Incidents By Role**: An icon of a person, an 'Add' button, and a small 'x' icon.
- Closed Job Incidents (Past 6 Months)**: An icon of a person with a checkmark, edit, delete, and 'Add' buttons. This widget is highlighted with a red box.
- Unassigned Closed Incidents**: An icon of a person, an 'Add' button, and a small 'x' icon.

- Add the widget to the dashboard. If no data is returned, click Use widget's date range.

A context menu is open for the 'Closed Job Incidents' widget. The menu items are:

- Use widget's date range** (highlighted with a red box)
- Edit widget
- Remove Widget

#### 14.3.6 | Create a widget from an indicator

##### Abstract

Create a custom widget from an indicator and add it a dashboard or report in Cortex XSOAR.

To create a widget from an indicator, you need to run a query from the Threat Intel page, and then save the visual results as a widget. If you do not have a TIM license, the page is called Indicators.

- In the Threat Intel (Indicators) page, select the date rage from the dropdown list.
- In the query field, type the query criteria as required and run the query.

3. Click .

4. Follow the procedure in Create a widget using the widget builder.

5. Click Save.

The widget is added to the Widgets Library.

**NOTE:**

By default, the widget inherits the date range that you specify when creating the widget, but you can modify the date range when you create the dashboard or report. If the date range for the report or dashboard does not include the widget date range, the data is blank. To override the dashboard or report's date range, click Use Widget's date range.

- Add the widget to a report or dashboard, as required.

#### 14.3.7 | Edit a widget

##### Abstract

Edit a widget in the Widgets Library or in a dashboard or report in Cortex XSOAR.

You can edit an existing widget in the dashboard or report, or in the Widgets Library. If editing a widget in the Widgets Library, it is available to all users. If editing a widget in a dashboard or a report directly, the original widget in the Widgets Library is unaffected.

#### **NOTE:**

Not all widgets are editable, such as system widgets.

1. Create or edit a dashboard or a report.

2. From the widget, select  → Edit widget .

The edits to the widget in the dashboard or report, appear only for the dashboard or report. If you want to make changes that are available for other users, dashboards, or reports, edit the widget directly in the Widgets Library by clicking the pencil edit icon. You can then adjust the size and move the widget as required.

If the widget is not in a dashboard or report, you need to add the widget.

3. Edit the widget in the Widgets Library by following the procedure in Create a widget using the widget builder.

4. Click Save.

#### 14.3.8 | Add a widget in the War Room

##### Abstract

Add a script-based widget in the War Room in Cortex XSOAR.

You can add a script-based widget in the War Room by running a command. After creating a script in the Scripts page, to add the widget you need to run a command in the War Room.

1. Create a custom widget using a script.

2. Go to the War Room and run the command: !<scriptName>

where <scriptName> is the name of the script you created in step 1.

##### Add a custom widget in the War Room example

Example 26. Add a custom widget that returns indicator severity in an incident as a bar chart

1. Use the following script.

```
commonfields:
  id: ee3b9604-324b-4ab5-8164-15ddf6e428ab
  version: 49
name: IndicatorWidgetBar
script: |-
  # Constants
  HIGH = 3
  SUSPICIOUS = 2
  LOW = 1
  NONE = 0

  indicators = []
  scores = {HIGH: 0, SUSPICIOUS: 0, LOW: 0, NONE: 0}
  incident_id = demisto.incidents()[0].get('id')

  foundIndicators = demisto.executeCommand("findIndicators", {"query": "investigationIDs:{}".format(incident_id), 'size':999999})[0]['Contents']

  for indicator in foundIndicators:
    scores[indicator['score']] += 1

  data = {
    "Type": 17,
    "ContentsFormat": "bar",
    "Contents": {
      "stats": [
        {
          "data": [
            scores[HIGH]
          ],
          "groups": None,
          "name": "high",
          "label": "incident.severity.high",
          "color": "rgb(255, 23, 68)"
        },
        {
          "data": [
            scores[SUSPICIOUS]
          ],
          "groups": None,
          "name": "medium",
        }
      ]
    }
  }
```

```

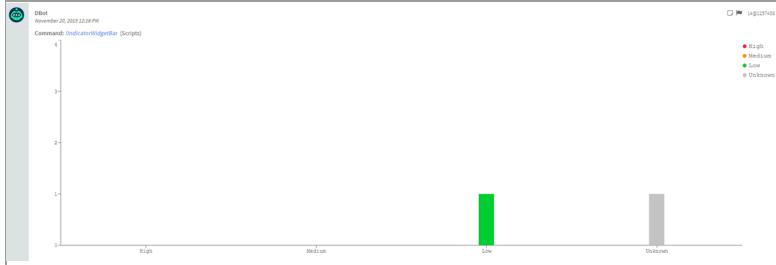
        "label": "incident.severity.medium",
        "color": "rgb(255, 144, 0)"
    },
    {
        "data": [
            scores[LOW]
        ],
        "groups": None,
        "name": "low",
        "label": "incident.severity.low",
        "color": "rgb(0, 205, 51)"
    },
    {
        "data": [
            scores[NONE]
        ],
        "groups": None,
        "name": "unknown",
        "label": "incident.severity.unknown",
        "color": "rgb(197, 197, 197)"
    }
],
"params": {
    "layout": "horizontal"
}
}
}

demisto.results(data)
type: python
tags:
- dynamic-section
enabled: true
scripttarget: 0
subtype: python3
runonce: false
dockerrimage: demisto/python3:3.7.3.286
runas: DBotWeakRole

```

2. Add the script in the War Room by running the !IndicatorWidgetBar command.

The custom widget appears in the War Room.



#### 14.3.9 | Saved By Dbot (ROI) Widget

##### Abstract

Customize the Saved by Dbot widget that calculates the amount saved by Cortex XSOAR. Return on Investment (ROI) widget.

In the Dashboard, Incidents tab, Cortex XSOAR comes with a number of pre-installed widgets, such as Saved by DBot (ROI) Widget.

The Saved by Dbot widget calculates the amount saved in US dollars according to actions carried out by all users in Cortex XSOAR across all incidents.

Although the widget comes out-of-the-box with Cortex XSOAR, you can add the Return on Investment (ROI) widget in the Widgets Library, which is identical to the Saved by Dbot (ROI) Widget.

##### NOTE:

The widget is disabled by default, as enabling it may affect performance when running large amounts of automations. To enable the widget, contact Customer Support.

The following parameters are used to calculate the amount saved by Dbot (ROI):

Parameter	Description
Man Hour	The amount in dollars of each hour for an analyst.

Parameter	Description
Roundtrip	The time it takes in minutes to run an integration task with any of the integrated products. This can be a command within a script or inside the War Room.
Report	The times it takes to write an incident report.
Script	The time it takes to undertake an action that a script would do.

The ROI is calculated as follows:

```
(# of times roundtrip completed * time taken to do roundtrip) + (# of times report generated * time taken to generate report)
+ (# of times script runs * time taken to run script)] * cost of 1 Man hour in dollars.
```

The time to run an integration task with any integrated product, the time to generate a report, and the time to run an automated action are set to 5 minutes.

'Number of times' is how many times an automated procedure has run since Cortex XSOAR was first used. The Saved by dBOT widget is based on absolute time and does not support date ranges.

You can change the way ROI is calculated based on your own statistics of time taken to perform the tasks for the actions when done manually. To change the statistics, select Settings & Info → Settings → System → Server Settings → Server Configuration → + Add Server Configuration and add the following :

Keys	Values
<code>roi.cost.manhour</code>	Amount in Dollars. Default: 60

You can also change the currency symbol from US dollars to a currency of your choice.

Customize the currency symbol in the saved by Dbot (ROI) Widget

The default currency symbol in the Saved by Dbot widget is the Dollar sign (\$). To change the currency symbol, you need to create a widget using a JSON file. For more details, see create a widget using a JSON file.

In this example, which you can use as a template, we changed the value for the `currencySign` argument to Euro (€).

```
{
  "size":5,
  "dataType":"roi",
  "params":{
    "currencySign":"€"
  },
  "query":"",
  "modified":"2019-01-12T15:13:09.872797+02:00",
  "shouldCommit":false,
  "name":"Return On Investment (ROI)",
  "shouldPush":false,
  "dateRange":{
    "fromDate":"2001-01-01T00:00:00Z",
    "toDate":"2001-01-01T00:00:00Z",
    "period":{
      "by":"",
      "byTo":"",
      "byFrom":"days",
      "toValue":null,
      "fromValue":30,
      "field":""
    }
  },
  "commitMessage":"",
  "isPredefined":true,
  "version":13,
  "id":"roi",
  "shouldPublish":false,
  "category":"others",
  "sort":null,
  "prevName":"Return On Investment (ROI)",
  "widgetType":"number"
}
```

## 15 | Incidents and indicators investigation

### Abstract

Investigate incidents and indicators that have been ingested into Cortex XSOAR.

Cortex XSOAR enables you to centralize and manage every aspect of your investigations. Consolidate evidence, assign and review tasks, and leverage the Workplan to orchestrate your response. Deduplicate incidents and create and close them efficiently. For indicators, create, extract and enrich them, and explore their relationships to gain deeper insights. If you have a TIM license, see the Indicator investigation section for more features, such as Unit 42 Intel data and creating a Threat Intel Report.

### 15.1 | Incidents

#### Abstract

Incidents are potential security data threats that are ingested or created in Cortex XSOAR for investigation and remediation.

Incidents are potential security data threats that SOC analysts identify and remediate. There are several incident triggers, including:

- SIEM alerts
- Mail alerts
- Security alerts

These alerts are generated from third-party services, such as SIEMs, mailboxes, and data.

Cortex XSOAR includes several out-of-the-box incident types, fields, and layouts, which can be customized to suit your use case. Incidents can also be created manually, from a JSON file, the Cortex XSOAR RESTful API, or an integration feed.

When incidents have been created, you can start managing and investigating incidents in Cortex XSOAR.

### 15.2 | Incident management

#### Abstract

View and manage incidents in Cortex XSOAR.

On the Incidents page, you can view all of the incidents in Cortex XSOAR and do the following:

#### NOTE:

If you are unable to perform a specific action or view data, you may not have sufficient user role permissions. Contact your Cortex XSOAR administrator for more details.

Action	Description
Search for incidents	<p>You can search for incidents by doing the following:</p> <ul style="list-style-type: none"> <li>• Search query: The Incidents page displays all open incidents from the last 7 days by default. For more information about search queries and to create a query and save it for future use, see Search for incidents.</li> <li>• Search incidents globally using the search box. For more information, see Use the search box.</li> </ul>
Filter incidents using the Bar Charts	<p>Bar charts display important incident information, such as the incident type, severity, and owner. You can change the criteria in each bar chart.</p> <p><b>NOTE:</b> Incidents sorted using an SLA/Timer field are sorted by the due date of the SLA field.</p>
Create a new incident	Create an incident manually. For more information, see Create an incident.
Create a widget	Create a widget based on the search criteria and add it to a dashboard or report. For more information, see Create a widget from an incident.

#### NOTE:

You can change how the top half of the incident page appears, by hiding the chart panel, and query panel, and switching to a detailed view.

#### Manage incidents from the incidents table

In the incidents table, view general information about each incident, such as the type, the severity, and when it occurred. The status of the incident is classified as follows:

Status	Description
Active	The investigation has started. The War Room is activated and the playbook starts, if assigned. Users can be assigned to this incident.
Pending	The investigation has not started and no War Room has been activated. As soon as you open the incident, it becomes active.
Closed	The investigation has been closed.

Incidents can be assigned a severity at incident creation when running a playbook, or after creation through the CLI or in the incident layout. Incident severity levels are:

- Critical (4)
- High (3)
- Medium (2)
- Low (1)
- Informational (0.5)
- Unknown (0)

You can do the following actions:

Action	Description
Investigate an incident	View, investigate, and take remedial action on the incident. For more information, see Investigate an incident.
Assign	Assign incidents to any user who has been added to Cortex XSOAR, including users who are marked as away. You can assign users to many incidents at one time.
Edit	Edit the incident parameters and then rerun a playbook on the incident, which is useful while developing playbooks. You can process an incident multiple times during playbook development, without creating new incidents every time.  <b>NOTE:</b> When batch editing multiple incidents, uploading files is currently not supported.
Mark as Duplicate	Deduplicate an incident. Closing an incident as a duplicate enables you to investigate one rather than multiple incidents. When selected, you need to add the ID you want to retain. When validated, and the other is closed as a duplicate, the duplicated incident is removed from the table.  If you want to link an incident with or without closing you can use the !linkIncidents command. For more information, see Link incidents.
Run Command	You can select multiple incidents and run a command across all of them.
Export	Export incidents to an Excel or a CSV file. For more information, see Export incidents.

Action	Description
Close	<p>You can select multiple incidents and close all of them. If required, add the close reason and details. The investigation will be closed.</p> <p>When you close an incident, the close reason is set to whatever value you last entered. For example, when closing an incident, if you initially selected False Positive as the Close Reason, reopened, and closed it again, leaving the Close Reason empty, the empty Close Reason will overwrite the previous Close Reason. To keep the close reason that was entered previously on the incident, add the previous value in the Close Reason argument.</p> <p><b>NOTE:</b></p> <p>The close reasons are customizable by server configurations. Provided you have administrator permission, you can change the reasons. For more information, see <a href="#">Customize incident close reasons</a>.</p> <p>You can also close the incident when investigating the incident.</p>
Delete	<p>You can select multiple incidents and delete all of them.</p> <p>You can also delete the incident when investigating the incident.</p>
Star an incident	<p>To help you focus on the most important incidents, you can mark an incident as a favorite. Starring incidents enables you to narrow down the scope of incidents on the Incidents page.</p>

**TIP:**

Any incidents assigned to yourself, starred incidents, and incidents you are participating in, can easily be accessed in the My Incidents section.

**Further information**

To see how to manage incidents, watch the following video in Live Community:

Working an Incident

**15.2.1 | Search for incidents****Abstract**

Create a search query for incidents and save search queries.

Cortex XSOAR comes with powerful search capabilities. You can search for data by:

- Using the Search Query: Cortex XSOAR searches for information using the Bleve query syntax. The search query appears on several pages such as Incidents, Indicators, and Playbooks. To search for all incidents that have the status as pending and are critical, type `status:Pending severity:Critical`. You can save and share queries, as required.
- Using the search box: Cortex XSOAR searches for incidents, entries, evidence, investigations, and indicators. The search box appears in the top right-hand corner of every page.

**Use the search query**

By default, the Incidents page displays all open incidents from the last seven days. You can customize which incidents are displayed by creating and saving queries.

When you start typing your search, Cortex XSOAR lists all the indexed fields, such as type and severity, including custom and out-of-the-box fields. The search follows the Bleve query syntax, which is similar to the Lucene query syntax but with some differences, such as query syntax for numeric ranges and date ranges. For more information, see [Bleve Query String Query](#).

The search is performed on certain pages such as incidents, indicators, or the entire data (such as titles, entries, chats).

**NOTE:**

To explicitly use the following characters in a search query, place them within double quotes. An escape character \ is not required.

`&& || ! {} [] () ~ * ?`

To explicitly use the following characters in a search query, place them within double quotes and use an escape character \.

`\, \n \t \r " ^ : and space`

For information about using special characters, see [Run commands in the CLI](#).

**NOTE:**

For precise results when searching for all long text, phase, name, reason, details or type, set the Server Configuration, `incident.search.exact.match.only` to true. For example, when doing a search for `type:Phish Mail`, if the server configuration is set to true, the results returned include the exact text `Phish Mail` and not each word separately. Another option to return exact text, just for name, type and phase, is to add the term "raw" preceding the query in your search. For example, rather than just entering `type:Phish Mail`, type `rawType:"Phish Mail"`.

You can add inputs when searching for data, such as:

Input	Description
Add text	Type any text. The results show all data where one of the words appears. For example, the search <code>low virus</code> returns all data where either the string <code>low</code> or the string <code>virus</code> appears.
and	Searches for data where all conditions are met. For example, <code>status:Active and severity:High</code> finds all incidents with an active status that have a high severity.
or	Searches for data where either conditions are met. For example, <code>status:Pending or severity:High or severity:Critical</code> finds all incidents with a pending status and with severity high or critical.
*	Wildcard search: * and ? should be used when searching for partial strings. For example, when searching for all scripts that start with AD, use <code>AD**</code> . If you need to search for a script which contains "get", search for <code>*get*</code> .
""	An empty value.
-	Excludes from any search. For example in the Incidents page the <code>-status:closed -category:job</code> searches for all incidents that are not closed and for categories other than jobs.
"me"	Filters incidents by a user's account. For example, <code>owner:{me}</code> will display all incidents where I am the owner. It can also be used for other fields such as <code>createdBy:{me}</code> which will display all incidents I created.
Relative time. For example, "today", "half an hour ago", "1 hour ago", "5 minutes ago", "10 days ago", "5 seconds ago", "five days ago", "a month ago", "in 1 year".	<p>Relative time in natural language can be used in search queries. Time filters - &lt; and &gt; can be used when referring to a specified time, such as <code>dueDate:&gt;="2018-03-05T00:00:00 +0200"</code>, or when searching for high severity incidents: <code>Severity:High and created:&gt;= "1 hour ago"</code></p> <p><b>NOTE:</b></p> <p>The timezone for searches is UTC. The system timezone is not used.</p> <p>When adding some fields, such as <code>Occurred</code> you can enter the date from the calendar. You can also filter the date when the results are displayed.</p>
Search using Regex	To use Regex, you need to use the value <code>//</code> . For example, to search for indicator values that contain <code>www</code> and end with <code>.com</code> , type: <code>value: "/w{3}...*.com/"</code> . This returns values such as <code>www.namecheap.com</code> , <code>www.kloshpro.com</code> .
Search for indicator values	To search for indicator values that contain lower-upper a-z letters and 0-9 numbers with a length of 32, type: <code>value: "[a-zA-Z0-9]{32}/"</code> . This returns values such as <code>775A0631FB8229B2AA3D7621427085AD</code> , <code>87798e30ca72f77abe624073b7038b4e</code> .
Timer/SLA fields	To search for Timer/SLA fields in incidents, see Search incidents for Timer/SLAs.

Save a search query

After defining the search query, you can save it for future use. The search query and the bar charts are saved.

**TIP:**

To edit an existing saved query, create a new query and save it with the exact name of the query you want to replace.

1. Select the date range to search (next to the Created field).

By default, the date is set to the last 7 days.

2. In the query bar, type your search criteria.

By default, the query is `-status:closed -category:job`, which searches for categories other than jobs and not those that have been closed. You can add fields like severity or type to narrow your search to critical issues or issues of a certain type.

#### **NOTE:**

If you change drill down in the bar chart fields, the query also changes. For example, in the Severity bar chart, if you click High, `severity:High` is added to the query.

3. Save the query.

a. Click .

b. Type a name for the query.

c. Save the query.

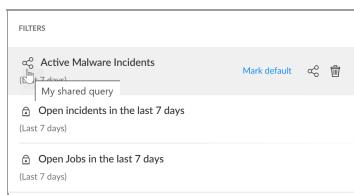
To view all saved queries, click . The list of saved queries appears. You can mark a saved query as a default, or delete a query.

#### Share saved queries

Shared queries enable you to share your customized configurations with all users. For example, you can define queries for security analysts to help focus them on incidents relevant for them to analyze.

Once you create and save a query, to share it with all users click  and then click  for that query.

The icon next to the name of the query changes to . Hovering over this icon in the list of saved queries shows that the query is shared. To remove sharing, click  and remove the users.



The shared query appears in the users' Saved queries list. Users see the query with a  icon and the name of the shared query owner.

#### **NOTE:**

- Edits made to shared queries are not saved. To save an edited version of the shared query, make a copy and then edit and save it.
- Copying the shared query or clicking Mark Default (to make the query the page default) keeps the shared query in the user's Saved queries list even if the shared query owner removes the share. Otherwise, the query will disappear from the users' Saved queries list if the query owner removes the share.

#### Use the search box

The search box searches for incidents, investigations, and indicators. The search box appears in the top right-hand corner on most pages. You can either type free text or search using the search query format (use the arrow keys to assist you in the search). For example, `incident.severity:Low` searches for all incidents that have `Low` in the severity category.

If using the search box during an investigation, you can select whether to search across all incidents or limit the search to the current incident.

#### **NOTE:**

When searching in the current incident, Cortex XSOAR searches only the War Room entries. If a value exists in the incident but is not a War Room entry, no results are returned.

#### Further information

For more information about how to search for incidents and indicators, see the following video in Live Community:

#### Searching in XSOAR

## 15.2.2 | Create an incident

#### Abstract

Create a new incident manually, through the API, ingest incidents, or import a JSON file.

You can create incidents in Cortex XSOAR from:

- The Incidents page
- An indicator
- A JSON file (primarily used for playbook testing)
- The API

To create a single incident using the API, use `/incident`. If you create an incident via the API and do not set `createInvestigation: true`, the incident is created but an investigation will not be opened and a playbook will not automatically run. For more information, see Create or update an incident.

To view the full API documentation, go to Cortex XSOAR 8 API Reference guide.

- Integration feeds

Incidents can be created from an integration instance. For more information about how to fetch incidents, see Fetch incidents from an integration instance.

#### NOTE:

If you can't create an incident from any of these options, you may not have sufficient user role permissions. Contact your Cortex XSOAR administrator for more details.

[Create an incident on the Incidents page](#)

To manually create an incident:

1. Select Incidents → New Incident.
2. Add the relevant data as required.
3. Create new incident.

The incident is added to the incidents table.

#### NOTE:

If any fields are missing, these fields can be added when configuring a layout.

You need administrator permission to configure a layout. For more information, see Incident layout customization.

[Create an incident from an indicator](#)

1. In the Indicators tab, select the indicator.
2. Click Create incident.

The incident appears in the incidents table on the Incidents page.

[Create an incident from a JSON file](#)

The import JSON feature enables you to import event data from third-party software and use it to create new incidents in Cortex XSOAR. These incidents can be used to build and troubleshoot playbooks for integrations that have not yet been installed or configured.

1. Go to Settings & Info → Settings → Object Setup → Incidents → Classification & Mapping and click the mapper you want to use.
2. From the Get Data drop-down, choose Upload JSON and then select the JSON file you want to upload.
3. Map the fields as required. For more information, see Classification and mapping.
4. Click  and select Create Incident from JSON.
5. Select the incident type and Create Incident.

### 15.2.3 | Export incidents

Abstract

Export incidents to an Excel or CSV file.

You can export one or more incidents to an Excel or CSV file.

If you want to export an incident as a JSON file, run the `!js script="return ${.}"` command in the War Room.

**NOTE:**

When exporting an incident to a CSV format, Cortex XSOAR generates the report in UTF8 format. If you want to export an incident that contains Cyrillic characters, such as Russian and Greek, you need to change the format to UTF8-BOM.

Administrator permission is required to change server configurations, including the format. For more information, see Export an incident to CSV using the UTF8-BOM format.

The data does not include files, attachments, and artifacts. All text is plain text (there is no formatting).

Before you begin

Enable pop-ups from your Cortex XSOAR tenant.

Select which data appears in your exported file by adding columns to the incidents table. If a column is hidden, the data is not exported. You can hide, show, or reorder the columns in the table by using the settings icon on the Incidents page.

1. On the Incidents page, at the top of the incidents table, click the settings wheel to configure the columns to include for export.
2. Select the incidents to export, and click Export.
3. Select one of the following:
  - Summary Report (CSV file)
  - Detailed Report (Excel file)

You can export up to 1,000 incidents at a time. If the incidents you select contain more than 10,000 combined entries, an error appears and the file is not generated. The maximum file size for download is 100 MB.

**NOTE:**

The date format displayed in the Excel/CSV file matches the timestamp format set by the user on the Server Settings page. If you haven't set a timestamp format, the default timestamp format is used.

## 15.3 | Investigate an incident

Abstract

Investigate and take remediation steps in Cortex XSOAR.

You can open an incident investigation:

- Automatically: If associated with a playbook, incidents open automatically for investigation and run the associated playbook.
- Manually: Open an incident manually by selecting the incident in the Incidents table.

**NOTE:**

After an incident is created, it is assigned a Pending status. When you start to investigate an incident the status changes automatically to Active, which starts the remediation process.

- In the CLI: If you want to open an incident in the CLI, type `/investigate id=<incidentID#>`.

You can limit access to investigations and restrict investigations according to your requirements, as described in Limit access to investigations using access control.

**NOTE:**

If you are unable to perform a specific action or view data, you may not have sufficient user role permissions. Contact your Cortex XSOAR for more details.

Start the investigation

When you open an incident, you can see various tabs that assist you in the investigation. The following tabs are common to most incident types:

**NOTE:**

Tabs, tab names, sections, and fields vary according to the incident layout.

In an investigation, images from external links don't appear, as they are restricted due to security issues. To use an image, either upload the image using base64 or upload it using markdown in the War Room.

Tab	Description
Case Info	<p>A summary of the incident, such as case details, outstanding tasks, linked incidents, and evidence. Some fields are informational and some are editable. Includes the following sections (depending on the layout):</p> <ul style="list-style-type: none"> <li>• <b>CASE DETAILS:</b> A summary of the incident, such as type, severity, and when the incident occurred. Update these fields as required.</li> <li>• <b>WORK PLAN</b> When you click on the section, you can view or take action on the following: <ul style="list-style-type: none"> <li>◦ Playbook tasks: When a playbook runs, any outstanding tasks appear. You can take various actions here or in the Work Plan tab.</li> <li>◦ To-Do Tasks View or create To-Do tasks.</li> </ul> <p>You can also create To-Do Tasks from the Actions tab. See Incident Tasks.</p> </li> <li>• <b>NOTES:</b> If added to the layout, notes help you understand specific actions taken, and allow you to view conversations between analysts to see how they arrived at a certain decision. You can see the thought process behind identifying key evidence and identifying similar incidents.</li> <p>You can add notes in this section or in the War Room. Notes are searchable when using the incidents search bar.</p> <li>• <b>EVIDENCE:</b> A summary of data marked as evidence. You can add evidence in this tab, the NOTES field, or the Evidence Board tab.</li> <li>• <b>LINKED INCIDENTS:</b> Add or remove linked incidents. For more information, see Link incidents.</li> </ul>
Investigation	<p>Provides an overview of the information collected about the investigation, such as indicators, email information, and URL screenshots.</p>
War Room	<p>A comprehensive collection of all investigation actions, artifacts, and collaboration. It is a chronological journal of the incident investigation. Each incident has a unique War Room. For information, see Use the War Room in an investigation</p>
Work Plan	<p>A visual representation of the running playbook that is assigned to the incident. For more information, see Use the Work Plan in an investigation.</p>
Evidence Board	<p>View any entity that has been designated as evidence. The Evidence board stores key artifacts for current and future analysis. You can reconstruct attack chains and piece together key pieces of verification for root cause discovery. For more information, see Evidence Handling.</p>

#### Incident actions

You can do several actions when investigating an incident, such as adding members, creating a report, and restricting incidents.

When viewing an incident, from the Side panels dropdown, you can do the following:

Action	Description
Quick View	<p>A summary of the incident, timeline information, labels, and indicators.</p>
Incident tasks	<p>Add tasks for users to complete as part of an investigation. For more information, see Incident Tasks.</p>

Action	Description
Team	<p>Add or delete incident team members.</p> <p><b>NOTE:</b></p> <p>When you mention team members in the CLI, they are automatically added as team members.</p>
Context data	<p>View context data to see what information was returned. The context is a map (dictionary) created for each incident and is used to store structured results from the integration commands and scripts. Context keys are strings and the values can be strings, numbers, objects, and arrays.</p> <p>Context data acts as an incident data dump from which data is mapped into incident fields. When an incident is generated in Cortex XSOAR and a playbook or analyst begins investigating it, context data will be written to the incident to assist with the investigation and remediation process.</p> <p><b>NOTE:</b></p> <p>All incident data stored in incident fields are also stored in the context data. In most cases, not all context data is stored in incident fields. Incident fields represent a subset of the total incident data.</p> <p>When an incident is created, the incident data is stored in the context data, under the <code>incident</code> key. When an investigation is opened and integration commands are run, data returned from those commands is also stored outside of the main <code>incident</code> key.</p> <p>For more information, see Use incident context data.</p>

When viewing an incident, from the Actions dropdown, you can do the following:

Action	Description
Edit	Edit the incident, as required.
Report	Create a report to capture investigation-specific data and share it with team members. For more information, see Create an incident summary report.
Add a child incident	<p>Child investigations are used to compartmentalize sensitive War Room activity. You can create child investigations to collaborate discreetly with a select group of people on a specific topic of investigation. Child investigations are also used where a secondary investigation is needed and its content may add too much "noise" to the original investigation.</p> <p>Select the Restricted checkbox to turn the child investigation into a discrete investigation.</p>
Restrict/Permit an incident	Restrict an investigation for the incident owner and team. If restricted, select permit to open the incident to all users. For more information, see Limit access to investigations using access control.
Close/Reopen an incident	<p>Mark the incident as closed. If closed, you can select Reopen the incident.</p> <p>When you close an incident, the close reason is set to whatever value you last entered. For example, when closing an incident, if you initially selected False Positive as the Close Reason, reopened, and closed it again, leaving the Close Reason empty, the empty Close Reason will overwrite the previous Close Reason. To keep the close reason that was entered previously on the incident, add the previous value in the Close Reason argument.</p>
Retain/Undo Retain an incident	Mark the incident for retention or disable retention for the incident. For more information, see Retain incidents.
Delete	Delete the incident from the database.

Incident navigation

You can navigate directly to a specific incident via the incident ID or incident name, using Ctrl+ K for Windows or Command-K for macOS.

When investigating an incident opened from My Incidents or the main Incidents page, you can navigate to the next/previous incident from within the incident, without returning to the original list. The navigation buttons appear next to the Action button. The total number of incidents from the list of incidents is shown (depending on your search criteria) and where you are in the list. For example, in the last 30 days, there were 7000 incidents. When opening an incident, you can investigate 7000 incidents using the navigation buttons without returning to the Incidents page.

Only users with permission to edit incidents can view the navigation buttons.

The navigation buttons are only available if the incident is opened from My Incidents or the Incidents page. If you navigate directly to an incident, without going through the Incidents page or My Incidents list, no navigation buttons appear.

### 15.3.1 | Retain incidents

Abstract

Retain up to 1000 incidents.

You can mark up to 1000 incidents for permanent retention so that any important incidents can't be inadvertently deleted manually, or by an API call.

#### **NOTE:**

Up to 1,000 incidents per tenant can be selected. Retained incidents are not deleted. If you reach 1000 retained incidents, you won't be able to add additional incidents, unless you disable incident retention for some or all of your existing retained incidents.

Only user roles that have the Retain incident permissions, can retain or undo incident retention. For more information, see Role-based permissions.

How to retain an incident

1. On the Incidents page, select the incident you want to retain.
2. From the Actions dropdown button, select Retain Incident.

The lock icon appears when the incident has been marked for retention.

To disable retention for an incident, select Undo Retain Incident from the Actions menu.

To search for retained incidents in the Incidents search bar, use the retained field, with T (True) or F (False). You can also add the Retain Incident field to the Incidents table to easily view which incidents are retained.

### 15.3.2 | Limit access to investigations using access control

Abstract

Limit access to incidents and investigations in Cortex XSOAR.

In any SOC team, there are various roles and responsibilities. For example, you may have specific teams to deal with threats, such as threat intelligence researchers, security analysts (Tier 1), senior analysts (Tier 2), SOC leads, SOC managers, and SIEM engineers. Administrators can exclude access to incident actions and investigations using role-based permissions. For example, you may want to limit the ability to change the incident status or manage the Work Plan. For more information, see Role-based permissions.

You can limit access to investigations, by doing the following:

- Restrict an investigation
- Limit investigations according to specific user roles
- Give read-only access to certain user roles

Restrict an investigation

You can restrict an investigation to the incident owner and the team associated with the investigation.

Restrict an incident to only team members. For example, if an incident contains sensitive data, and you only want specific users to investigate the incident, you can mark the incident as restricted. Other users cannot view or access the incident. Team members are added automatically when you send them a notification in the CLI. You can remove the restricted investigation at any time.

#### **NOTE:**

All team members have read and write permissions. If you add team members, but their roles have read-only permission, the user still has read and write permission and can access the investigation.

1. Go to the Incidents page and select the incident you want to restrict.
2. Select Actions → Restrict incident.

To remove the restriction select Actions → Permit incident.

Confirmation appears in the War Room.

**NOTE:**

If using the CLI, run the `/investigation_restrict id=<id number>` or the `/investigation_permit id=<id number>` command.

Limit access to investigations according to specific roles

When you add a role to the incident, you restrict access to all roles other than those you have specifically added. For example, after an investigation is closed, add administrators or those with specialty roles, so only they can reopen or link incidents. The added roles have read and write permission, but all other roles do not have access (unless you have added them in the XSOAR Read Only Roles field).

**NOTE:**

- If you add a role, but the incident has been restricted to team members, and the user is not a team member, the user cannot access the incident regardless of the role. For example, if you restrict the incident to User A and User B team members who are Tier 1 analysts but then try to add Tier 2 analysts (none of whom are team members) to the list of roles, a Tier 2 analyst cannot access the incident.
- To access an incident, you must be assigned the same role that is assigned to the incident, even if you are the creator of the incident.

1. On the Incident page, open the incident to restrict access.

2. Do one of the following:

- If the Roles field is added to the incident layout, select the relevant role.
- In the CLI, run `!setIncident roles =<name of role>` to set the role.

You can also run the `/incident_set` command `roles <name of role>`, which has the same effect.

The War Room entry confirms that the role has been updated.

**NOTE:**

When you create or edit an incident, you can select the required Role.

You can add this field to the incidents table on the Incidents page (you can't add roles in the table).

Give access to Read-only roles

You can add a Read-only role to the incident, which restricts access to the incident. When granting read-only access, the user can view the incident but not edit it. For example, when an incident is in triage (phase 1), you may want all Tier-2 analysts to have read-only access, so that Tier-1 can edit the incident. When the phase changes to phase 2, Tier-1 has read-only access.

Adding a team member overrides this restriction, so if you add User A, (Tier 1) as a team member, even if you assign Tier-1 as a Read-only role, the user still has Read/Write access. You need to remove the user as a Team Member.

**NOTE:**

If you assign a role (read and write permission) and assign the same role as read-only, the user still has read/write permission. You need to remove the assigned role. If you restrict the incident, the read-only role does not override the restriction. In other words, team members' permission takes precedence.

1. On the Incident page, open the incident to restrict access.

2. Do one of the following:

- If the XSOAR Read Only Roles field is added to the incident layout, select the relevant role.
- In the CLI, you run `!setIncident xsoarReadOnlyRoles=<name of role>` to set the read-only role.

The War Room entry confirms that the role has been updated.

**NOTE:**

If added to the opening incident form, when editing or creating an incident, you can select the required XSOAR Read Only Roles.

You can add this field to the incidents table on the Incidents page (you can't add roles in the table).

### 15.3.3 | Incident Tasks

Abstract

Playbook tasks and to-do tasks are tasks users complete as part of an investigation. Add incident tasks as part of your investigation process.

Incident tasks are tasks for users to complete as part of an investigation, which is split according to the following:

Task	Description
Playbook task	A task that is part of the Work Plan (playbook) for an incident. When a playbook runs you can take action on any tasks that require attention in the Work Plan, such as assigning an owner, setting a due date, and completing the task. These tasks include the following subtypes: <ul style="list-style-type: none"> <li>• Automated tasks</li> <li>• Manual tasks</li> <li>• Manual conditional tasks</li> <li>• Data collection tasks</li> </ul>
To-Do tasks	An ad-hoc item that is not attached to the incident Work Plan. Create tasks for users to complete as part of an investigation. These are like a To-Do list that you keep in an investigation on an ad-hoc basis rather than the Work Plan which follows a pre-defined process.

**NOTE:**

You can close an incident even if there are open playbook tasks or open To-Do tasks.

You can view outstanding tasks in the INCIDENT TASKS pane, by clicking Side panels → Incident Tasks.

**NOTE:**

You can also access the INCIDENT TASKS pane from the Case Info tab, in the WORK PLAN section, or the TO-DO TASKS section if it has been added to the layout.

How to create a To-Do task

1. In the incident, click Side panels and then select Incident Tasks.
2. In the INCIDENT TASKS pane, click the To-Do Tasks tab.

**NOTE:**

If your Case Info tab in the incident layout includes a TO-DO TASKS section or has a WORK PLAN section you can access the INCIDENT TASKS section directly.

3. Click Add a task.
4. Add the Task Details as required:

Parameter	Description
Task Name	A meaningful name for the task (mandatory).
Task Description	A meaningful description of the task that provides sufficient information for the assignee to complete the task.
Assignee	The user to assign to the task. You can only assign a single user per task.
Set due date	The due date for the task. If the task is not completed by this date, it is marked as overdue but is not a roadblock for the investigation.
Tag the result with	Tags to apply to the to-do task, so you can easily find it in the War Room.

5. Save the task.

**TIP:**

Use the !MyToDoTasksWidget command in the CLI to see all your assigned tasks in the War Room. You can also use the !Todo command to manage the task, such as add, assign, and complete.

When you are added to a task, you receive a notification by email. To turn this on or off, go to <your name → User Preferences → Notifications and select the relevant section.

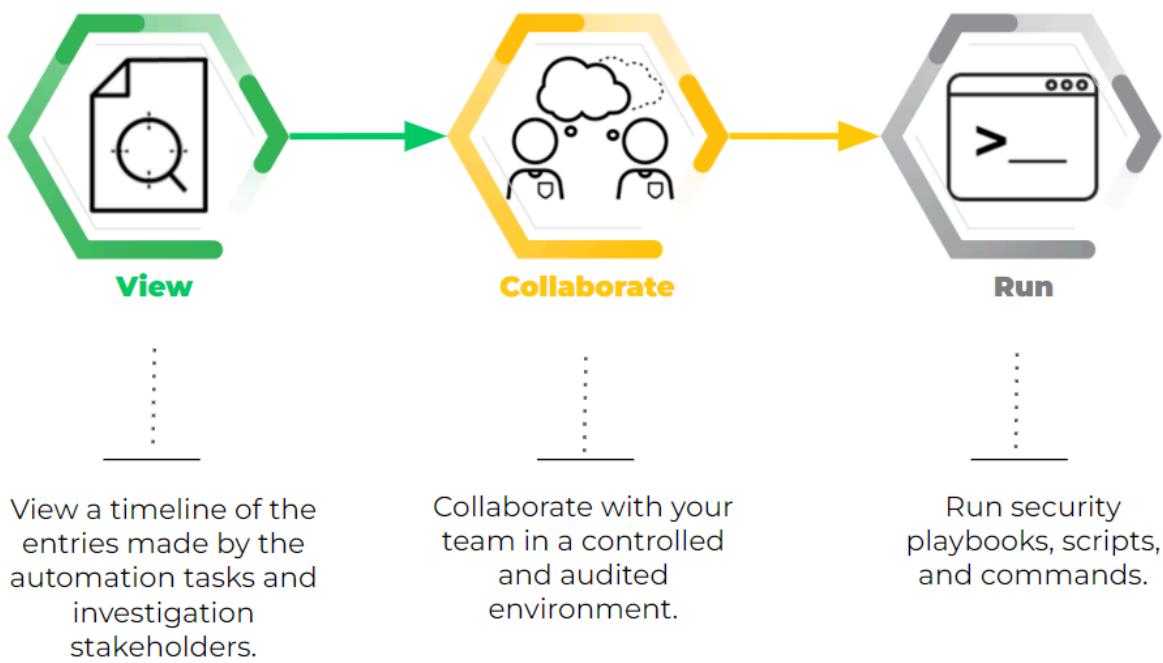
### 15.3.4 | Use the War Room in an investigation

#### Abstract

Use the War Room for real-time investigation into an incident, to filter war room entries, and to disable indicator notifications.

The War Room contains an audit trail of all automatic or manual actions that take place in an incident. A War Room is where you can review and interact with your incidents. Cortex XSOAR provides machine learning insights to suggest the most effective analysts and command-sets. Each incident has a unique War Room.

## The War Room: A Chronological Journal



Within Cortex XSOAR, real-time investigation is facilitated through the War Room, which is powered by ChatOps and helps you to do the following:

- Run real-time security actions through the CLI, without switching consoles
- Run security playbooks, scripts, and commands
- Collaborate and execute remote actions across integrated products
- Capture incident context from different sources
- Document all actions in one source
- Converse with others for joint investigations

Every Incident has a War Room, but every user has access, subject to permissions, to a private War Room called the Playground.

#### The Playground

The Playground is a non-production environment where you can safely develop and test data, such as scripts, APIs, and commands. It is an investigation area that is not connected to a live (active) investigation.

To access the playground you can do the following:

- Type any command in the CLI (not in the incident).
- If you type a command in the incident, the results are returned to the incident War Room, not the Playground.
- If you have an Admin role, in My Incidents on the sidebar, click Playground.
  - Type `ctrl + k` and then select the War Room
  - In any browser, type: <https://<tenant>/WarRoom/playground/>

**NOTE:**

If you want to erase an existing playground and create a new one, run the `/playground_create` command.

**The War Room**

When you open the War Room, you can see all the actions taken on an incident, such as commands, notes, and evidence in several formats such as Markdown, and HTML. When Markdown, HTML, or geographical information is received the content is displayed in the relevant format. You can schedule a command in the War Room to run at a specific time. For more information, see Schedule a command in the War Room.

To view specific data entries, you can filter entries by selecting the relevant checkbox, such as:

- Chats: Shows communication between team members.
- Notes: Any entries marked as notes.
- Files: Anything uploaded to the War Room in a playbook, script, or by the analyst
- Incident History: Any incident field or SLA Timer field that was modified
- Commands and playbook tasks: Any actions taken by playbook tasks or run manually by the analyst

You can also highlight any command thread for tracking commands.

**NOTE:**

Cortex XSOAR does not index notes, chats, and pinned as evidence entries.

In each War Room entry, you can take the following actions:

Action	Description
Edit	You can edit, format, or delete your entries. If an entry has been changed, a History link will appear where you can view all changes to the entry.
Mark as Evidence	Opens the Mark as evidence window where you specify the evidence details to be saved in the Evidence Board. The Evidence Board stores key artifacts for current and future analysis. You can also add evidence in the Case Info tab or the Evidence Board tab. For more information, see Evidence Handling.
Mark as note	<p>Marks the entry as a note, which can help you understand why certain action was taken and assist future decisions.</p> <p>You can also add a note by doing the following:</p> <ul style="list-style-type: none"> <li>• Upload a file to the War Room by selecting Mark as Note.</li> <li>• If the Case Info tab includes a NOTES section, add it to the section.</li> <li>• In a playbook task (Advanced tab)</li> </ul> <p>Tasks can be automatically added from script outputs as notes.</p> <ul style="list-style-type: none"> <li>• In the CLI by running the <code>!markAsNote entryIDs=&lt;ID of the war room entry&gt;</code> command.</li> </ul> <p>In the relevant War Room entry, click Copy to CLI to retrieve the ID of the War Room entry.</p> <p>When marked as a note, it is highlighted, so you can easily find them in the War Room or the Case Info tab.</p>
View artifact in new tab	Opens a new tab for the artifact.
Detach from task	Removes a task from the artifact.
Attach to a task	Adds a task to the artifact.
Download artifact	Downloads an artifact according to the entry type, such txt files for text, json for a JSON entry, etc.

Action	Description
Add tags	Add any relevant tags to use that help you find relevant information.
Copy to CLI	<ul style="list-style-type: none"> <li>ID: Entry IDs are used to uniquely identify War Room entries and take the format &lt;ENTRY_IDENTIFIER&gt;@&lt;INCIDENT_ID&gt;, for example, 54925dc3-a972-4489-8bef-793331fa6c77@1. Many out-of-the-box commands and scripts use entry IDs arguments to pass in files as inputs.</li> <li>URL: Copy the URL which is a direct link to the War Room entry</li> </ul> <p>To find the entry ID or URL of an entry in the War Room, click on the vertical ellipsis icon at the upper right of the entry, then copy the value.</p>

You can also upload files to the War Room by selecting the paperclip icon next to the CLI. Any files that have been uploaded can also be downloaded from the War Room entry.

#### CAUTION:

You are not protected from malicious content when downloading files from the War Room.

#### 15.3.5 | Schedule a command in the War Room

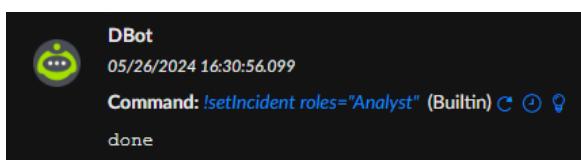
You can run a scheduled command once or on a recurring schedule, setting the start time, end time, and frequency. Some common use cases for scheduling a command:

- Mirroring a ticket from an external ticketing system.
- Sending an email to a user, waiting a determined amount of time, and then sending the email again if a response has not been received.

#### NOTE:

Scheduled commands run using the tenant's timezone.

1. Open an incident, locate the command entry in the War Room, and click the clock icon.



By default, scheduling options are displayed in a human-readable view. For recurring commands, you can use either a human-readable view or switch to the Cron view.

2. To schedule a command using the human-readable view:

- For a non-recurring command, set the date and time for the command to run.
- For a recurring command, select Recurring and then set the frequency.

3. To schedule a recurring command using the Cron view:

- a. Select Recurring, and then click Switch to Cron view.
- b. Define the Cron expression.
  - Expand Show cron examples to view sample schedules in Cron format.
  - Set a start date and time.
  - (Optional) Set an end date and time.

4. Save the Schedule.

To remove a scheduled command, click the clock icon and then click Remove schedule.

You can also schedule commands to run in the War Room by using a script. For more information, see the ScheduleCommand script.

### 15.3.6 | Run commands in the CLI

#### Abstract

Cortex XSOAR enables you to run system commands, integration commands, scripts, and more, from an integrated CLI.

Cortex XSOAR enables you to run system commands, integration commands, and scripts from an integrated command line interface (CLI), which enables you to make comments in your incident (in plain text or Markdown) and to execute automation scripts, system commands, and integration commands. This gives SOC teams the power to execute automations ad-hoc to support their investigations or make notes as they investigate incidents.

#### **NOTE:**

If you are unable to run commands in the CLI, you may not have sufficient user role permissions. Contact your Cortex XSOAR administrator for more details.

In the CLI, you can run various commands, by typing the following:

Action	Description
!	Runs integration commands, scripts, and built-in commands, such as adding evidence and assigning an analyst.
/	Runs system commands and operations, such as adding notes and closing an investigation.
@	Sends notifications to administrators, teams, and analysts by tagging users.

You can find relevant commands, scripts, and arguments with the CLI's auto-complete feature. This also includes fuzzy searching to help you find relevant commands based on keywords. If you type the exclamation mark (!) and start typing, autocomplete populates with options that might suit your needs. For example, if you want to work with tasks, type !task, and all commands and scripts that include the task in their name will display.

The CLI is available throughout Cortex XSOAR, except Marketplace and while editing Playbooks.

#### **NOTE:**

You can use the up/down arrow buttons in the CLI to do a reverse history search for previous commands with the same prefix.

You can hide the CLI when it is not needed by clicking on the down arrow to the right of the CLI. You can click the same button to restore the CLI. If you can't see the ^ button, remove the ? Help Center button. To restore the Help Center, click Help (left menu) and click In-App Help Center.

#### Using special characters

Characters	Description
&&,   , !, {, }, [, ], (, ), ~, *, ?	To use these characters, place them within single or double quotes. An escape character \ is not required.
\, \n, \t, \r, ", ^, :, comma, and space	To use these characters, place them within single or double quotes and use an escape character \.

#### **TIP:**

When writing a query or complex text in the CLI, we strongly recommend enclosing your text with the backtick (`) character. Text within the backticks does not require you to escape single quotation marks (''), double quotation marks (" "), or backslashes (\).

#### Common Arguments

The following common arguments are available for every script run from the CLI.

Argument Name	Description
auto-extract	Whether/when to extract indicators. Possible values: <ul style="list-style-type: none"> <li>inline: Extracts indicators within the indicator extraction run context (synchronously).</li> <li>outofBand: Extracts indicators in parallel (asynchronously) to other actions.</li> <li>none - Does not extract indicators (recommended for scripts with large outputs when indicator extraction is not required).</li> </ul>
execution-password	Supplies a password to run a password-protected script.
execution-timeout	Defines how long a command waits in seconds before it times out.
extend-context	Select which information from the raw JSON you want to add to the context data. For a single value: contextKey=RawJsonOutputPath For multiple values: contextKey1=RawJsonOutputPath1::contextKey2=RawJsonOutputPath2
ignore-outputs	Possible values: true or false. If set to true, it does not store outputs in the context (besides extend context).
raw-response	Possible values: true or false. If set to true, it returns the raw JSON result from the script.
retry-count	Determines how many times the script attempts to run before generating an error.
retry-interval	Determines the wait time (in seconds) between each script execution.
using	Selects which integration instance runs the command.
using-brand	Selects which integration runs the command. If the selected integration has multiple instances, the script may run multiple times. Use the using argument to select a single integration instance.
using-category	Selects which category of integrations runs the command. If the selected category includes multiple integration instances, the script may run multiple times. Use the using argument to select a single integration instance.

#### Run commands in the Automations browser

You can view and run commands and scripts (not system commands, operations, and notifications) in the Automations Browser, by clicking  next to the CLI.

The Automations Browser enables you to run commands and all associated arguments. The scripts and commands are separated into sections such as scripts and built-in commands. In each argument, you can do the following:

- Hardcode the value
- Use a dynamic value

You can dynamically pass information into the argument, by clicking the curly bracket. For example, the `EmailAskUser` command asks a user a question via email. In the `email` argument, rather than typing the user's email address, you can send it to whoever created the incident.

1. In the `email` field, click the curly brackets.

2. In the search box, enter `created`.

3. Under INCIDENT DETAILS click `Created by`.

The `email` argument appears as  `${incident.dbotCreatedBy}`.

4. Run the command.

An email is sent to the user who created the incident.

You can use transformers and filters to filter and transform data from the command. For more information, see [Filter and transform data](#).

#### Common arguments

Argument	Description
Using	Selects which integration instance runs the command.
Extend context	Determines the wait time (in seconds) between each script execution.  For a single value: <code>contextKey=RawJsonOutputPath</code>  For multiple values: <code>contextKey1=RawJsonOutputPath1::contextKey2=RawJsonOutputPath2</code>
Ignore outputs	Does not store outputs in the context (besides extend context).
Execution timeout (seconds)	Defines how long a command waits in seconds before it times out.
Number of retries	Determines how many times the script attempts to run before generating an error.
Retry interval (seconds)	Determines the wait time (in seconds) between each script execution.

#### Examples using the CLI

To run the `print` script with a value of "hello" and the key `a` from the context:

```
!Print value="hello ${a}"
```

To run the `searchIncidentv2` script with the query of `myfield` equals "this is a test" using escape characters:

```
!SearchIncidentsV2 query="myfield:\\"this is a test\\""
```

To run the same query using backticks:

```
!SearchIncidentsV2 query=`myfield:"this is a test"`
```

To run the Python command returning Hello World using escape characters:

```
!py script="demisto.results(\"hello world\")"
```

To run the Python command returning Hello World using backticks:

```
!py script=`demisto.results("hello world")`
```

## 15.3.7 | Evidence Handling

### Abstract

Add evidence to the evidence board to assist with your investigation. Mark any entity as evidence in the War Room by adding tags.

While you're investigating an incident, you can add notes and evidence to assist you with your investigation.

Notes can help you understand why certain actions were taken and assist future decisions. Notes are highlighted, so you can easily find them, especially in the War Room.

When marking an artifact as evidence, these artifacts are added to the Evidence Board tab, which enables you to see all artifacts for current and future analysis in a single location.

#### **NOTE:**

You can change a note to evidence or vice-versa and have the same entry as a note and evidence.

### How to add evidence

You can add evidence by doing the following:

Action	Description
War Room Entry	<p>In a War Room entry, click <b>Mark as Evidence</b>.</p> <p>Add a description that should contain enough information, so it can be used for future reference. Adding a tag helps you to find the evidence by searching for the tag. You can also add a time and date when it occurred.</p> <p>When adding a time/date you need to save it before updating the evidence.</p>
Upload a file	Upload a file to the War Room by selecting <b>Mark as Evidence</b> .
Using the CLI	<p>Run the <code>!AddEvidence entryIDs=ID</code> of the war room entry command.</p> <p>In the relevant War Room entry, click <b>Copy to CLI</b> to retrieve the ID of the War Room entry.</p>
Playbook task	In a Playbook task (Advanced tab). Tasks can be automatically added as evidence from script outputs.
Case Info tab	<p>If the Case Info tab includes an EVIDENCE section, you can add it to the section.</p> <p>Whenever you add evidence, this appears in both the Evidence Board tab and the EVIDENCE section in your layout.</p>

### Evidence Board

The Evidence Board tab shows all the entries marked as evidence for current and future analysis. Typically you can use the Evidence Board to do the following:

- Reconstruct attack chains
- Piece together key pieces of verification for root cause discovery
- Construct a timeline of events that can further clarify your incident response
- Use it for audit reports and compliance requirements to show how you reached a decision.

You can search for evidence and select the date range when the evidence occurred.

When viewing an Evidence artifact you can see the following fields:

- occurred: The time/date that you added when the artifact occurred. For example, when the file was created. If no time/date is specified it is marked as Unknown.
- fetched: The time/date when the entry was created in Cortex XSOAR.
- markedDate: The time/date when you marked it as evidence.
- MarkedBy The user who marked it as evidence.
- Any Evidence fields you have added to the tab.

You can also edit or remove evidence from the Evidence Board.



Use the toggle button to switch between Table View or Summary View. In the Table View, you can remove, export, or show evidence in the War Room. In the Summary View you can remove or edit the evidence.

### 15.3.8 | Use the Work Plan in an investigation

#### Abstract

A Work Plan is a visual representation of the running playbook that is assigned to an incident. Use it to monitor and manage a Playbook workflow.

The Work Plan is a visual representation of the running playbook assigned to the incident. Playbooks enable you to automate many security processes, such as managing your investigations and handling tickets. Work Plans enable you to monitor and manage a playbook workflow, and add new tasks to tailor the playbook to a specific investigation.

In an investigation, when you open the Work Plan tab you can see the playbook, the playbook name, and navigation tools.

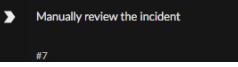
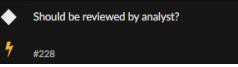
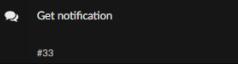
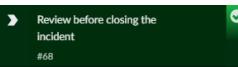
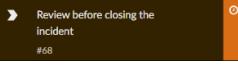
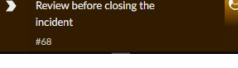
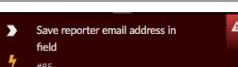
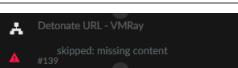
By default, the Follow checkbox is checked, which allows you to see the playbook executing in real-time. The playbook moves when a task is completed.

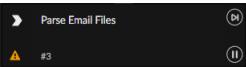
In the Work Plan you can do the following:

Action	Description
Change the default playbook	<p>On the left-hand side of the window, select the playbook you want to run.</p> <p>When changing the playbook, all completed tasks are removed and the new playbook will run. If you select playbooks several times you can view the history of which playbooks ran.</p>
Rerun the playbook	When changing the playbook, select the current playbook to run again.
View inputs and outputs	View the inputs and outputs of each task that has run. You can't view inputs and outputs of any task that hasn't run.
Manage tasks	<p>View, create, and edit a playbook task. For each task, you can do the following:</p> <ul style="list-style-type: none"> <li>Designate tasks as complete either manually or by running a script.</li> <li>Assign an owner</li> <li>Set a due date</li> <li>Add comments and completed notes, as required.</li> </ul> <p>You can manage these tasks in the CLI by using the /task command. For more information about tasks, see Incident Tasks.</p>
Export to a PNG	Export the Work plan to a PNG format for easy analysis.

The color coding and symbols in the Work Plan help you to easily troubleshoot errors or respond to manual steps. The following table displays the playbook tasks and icons in the Work Plan.

#### Playbook tasks and icons in the Work Plan

Task	Description
 Set Listener Mailbox #212	<p>Standard automated task</p> <p>The arrow and lightning bolt indicate a standard automated task. This task does not require any analyst intervention. They turn green automatically if they are successful.</p>
 Manually review the incident #7	<p>Standard manual task</p> <p>The arrow indicates a standard manual task. These tasks are used where usually it's not possible to automate them. You can add comments, assign them to an owner, and set a due date.</p> <p>You need to complete it before the Work Plan can continue.</p>
 Should be reviewed by analyst? #228	<p>Conditional task</p> <p>The diamond indicates a conditional task, which is either an automated conditional task (with the lightning bolt) or a manual conditional task. These tasks are used as decision trees in your work plan.</p>
 Get notification #33	<p>Data collection task</p> <p>The speech bubble indicates a data-collection task. This task prompts you to respond to multi-questions.</p>
 Process Email - Generic v2 #26	<p>Active task</p> <p>The gear icon indicates an active task.</p>
 Review before closing the incident #68	<p>Completed task</p> <p>The green check mark indicates a completed task.</p>
 Review before closing the incident #68	<p>Overdue task</p> <p>The clock icon indicates the task is overdue.</p>
 Review before closing the incident #68	<p>Pending manual task</p> <p>The orange user icon indicates that the playbook is pending action. The task requires you to open it and manually mark it as complete.</p>
 Save reporter email address in field #85	<p>Failed task</p> <p>The red warning icon indicates that the automation failed to complete as expected and requires manual inspection and troubleshooting. Contact your Cortex XSOAR administrator.</p>
 Detonate URL - VMRay skipped: missing content #139	<p>Skipped missing content</p> <p>The skipped task due to missing content, such as a missing integration.</p>
 Extract Indicators From File - Generic v2 #21	<p>Sub-playbook task</p> <p>The workflow icon indicates that the task is a playbook nested within the parent playbook. You can view that playbook by opening the task and selecting Open sub-playbook.</p>

Task	Description
	<p>Task containing a deprecated script</p> <p>The yellow warning indicates that the associated automation is deprecated. Deprecation means that the automation script is still available within the system but is no longer actively supported by the script author.</p>

### 15.3.9 | Link incidents

#### Abstract

Link incidents in the Linked Incidents section or the CLI.

When ingesting incidents, you may find that several incidents have similar or identical information. You have the following options:

- Set up automatic deduplication. Your administrator can set up pre-process rules or scripts in a playbook. For more information, see Incident deduplication in Cortex XSOAR.
- From the incidents table, mark the incident as duplicate. You select which incident to keep and which to close.
- From the Incident, in the LINKED INCIDENTS section, add linked incidents. These incidents are linked but not closed.
- In the CLI you can use the `!linkIncidents` command to deduplicate, and link/unlink incidents

When you link an incident without closing, you can view all similar incidents together without closing them as duplicates. When you link an incident you can see them all in one table and take action altogether, such as running commands or closing the incidents.

If you find during your investigation you want to unlink incidents, run the `!linkedIncidents` command in the CLI.

#### Link incidents in the incident layout

Before you start, note the incident ID you want to link.

1. In the Case Info tab, scroll to the LINKED INCIDENTS section.
2. Click the + icon.
3. Add the incident IDs you want to link, separated by a comma.
4. Click Submit.

The linked incident appears in the War Room and the LINKED INCIDENTS section.

5. (Optional) To take action or view all linked incidents, go to the Linked Incidents table by clicking  in the LINKED INCIDENTS section.

#### NOTE:

To unlink the incident, run the following command in the CLI:

```
!linkIncidents linkedIncidentIDs=<id> action=<unlink>
```

#### Link incidents using the CLI

1. In the CLI, run the following command:

```
!LinkIncidents linkedIncidentIDs=<id> action=<action value>
```

The linked incident appears in the LINKED INCIDENTS section and the War Room.

2. (Optional) To take action or view all linked incidents, go to the Linked Incidents table by clicking  in the LINKED INCIDENTS section.

To unlink the incident, run the following command:

```
!linkIncidents linkedIncidentIDs=<id> action=<unlink>
```

### 15.3.10 | Create an incident summary report

#### Abstract

Create and generate a custom Incident Summary report in Cortex XSOAR, from the incident page. Save reports as templates.

In an incident investigation, you can generate an incident summary report in PDF format, which enables you to capture investigation-specific data and share it with team members.

When generating a report, you can do the following:

Action	Description
Select a tab to generate a report from	<p>Apart from the War Room, Work Plan, and Evidence Board tabs, you can select which tab to generate a report from including any custom tabs or tabs from a layout installed from a content pack. For example, the Phishing Campaign layout includes the Campaign Overview and Campaign Management tabs. You can select any of those tabs to generate a report.</p> <p>When generating a report, you can decide what sections to include from the Case Info tab, by selecting Legacy Summary.</p> <p>You can save the reports as templates. Templates cannot be edited after they are created.</p>
Create a report from a template	<p>The Investigation Summary report is included out-of-the-box. This report includes the following sections:</p> <ul style="list-style-type: none"> <li>• General information</li> <li>• Close notes</li> <li>• Custom data</li> <li>• Investigation Timeline</li> <li>• Indicators</li> <li>• War Room notes</li> <li>• Evidence timeline and detailed evidence</li> <li>• Skipped tasks</li> <li>• Team members</li> <li>• Linked incidents</li> </ul>

#### TIP:

If you want a less detailed report, we recommend downloading the CaseManagement-Generic content pack which includes a Case Report. This report includes case details, investigation details, labels, closing information, indicators, team members, notes, and any War Room Chat.

The administrator can create a tab in your layout to include any information for reports. For more information about customizing layouts, see Incident layout customization.

After you create a template, it appears on the Reports page under Incident Reports.

How to create a summary report

Before you begin, enable popups in your browser.

1. Open the incident for which you want to create a report.
2. Select the tab that has the information you want to appear, and click Actions → Report.
3. Select one of the following:
  - To generate a new report, Select a tab to generate report from.  
Add the required properties. We recommend the landscape orientation, so that all information is displayed in the report.  
If you choose Legacy Summary, select the required sections.
  - To use an existing template, choose From Template tab and select the template.
4. If you want to use the report settings as a template, click the Save report as template checkbox.
5. Generate the report

#### NOTE:

You can also use the `!GenerateSummaryReports` command in the CLI to generate a report. If you want to automate the process, the administrator can use the Send Investigation Summary Reports Job playbook.

## 15.4 | Manage indicators

### Abstract

Perform actions (create, edit, export, delete) and search for indicators on the Cortex XSOAR Indicators (no TIM license).

After you start ingesting indicators into Cortex XSOAR, you can start your investigation, including extracting indicators, creating indicators, adding indicators to an incident, and exporting indicators.

#### **NOTE:**

You need a TIM license to investigate the indicator on the Indicator page, and use the Unit 42 features such as Sample Analysis, and Sessions and Submissions. For more information, see [Indicator investigation with a TIM license](#).

The Indicators page displays a list of indicators added to Cortex XSOAR, where you can perform the following indicator actions:

Action	Description
Create an indicator	<p>Indicators are added to the Indicators table from incoming incidents, feed integrations, or manually creating a new indicator.</p> <p>When creating an indicator, in the Verdict field, you can either select a verdict or leave it blank to calculate it by clicking Save &amp; Enrich, which updates the indicator from enrichment sources. After you select an indicator type, you can add any custom field data.</p>
Create an incident	Create an incident from the selected indicator and populate relevant incident fields with indicator data.
Edit	Edit a single indicator or select multiple indicators to perform a bulk edit.
Delete and Exclude	<p>Delete and exclude one or more indicators from all indicator types or a subset of indicator types. For more information, see <a href="#">Delete and exclude indicators</a>.</p> <p>If you select the Do not add to exclusion list checkbox, the selected indicators are only deleted.</p>
Export CSV	<p>Export the selected indicators to a CSV file. By default, the CSV file is generated in UTF8 format.</p> <p>You need administrator permission to change server configurations including the format. To change the format, see <a href="#">Export incidents and indicators to CSV using the UTF8-BOM format</a>.</p>
Export STIX	Export the selected indicators to a STIX file.
Upload a STIX file	To upload a STIX file, click the upload button (top right of the page) and add the indicators from the file.

#### **NOTE:**

By default, when editing a list or text values in an incident/indicator, the changes are not saved until you confirm your changes (clicking the checkmark icon in the value field). These icons are designed to give you additional security when updating fields in incidents and indicators.

You can change this default behavior by updating the server configuration. You need administrator permission to update server configurations. For more information, see [Configure inline value fields](#).

You can also undertake various actions on the indicator, such as:

Action	Description
Enrich an indicator	You can view detailed information about the indicator (WHOIS information for example), using third-party integrations such as VirusTotal and IPInfo. For more information, see <a href="#">Extract and enrich an indicator</a> .

Action	Description
Expire an indicator	You may want to expire an indicator to filter out less relevant alerts, allowing analysts to focus on active threats. For more information, see Expire an indicator.
View indicator relationships	Relationships enable you to enhance investigations with information about indicators and how they might be connected to other incidents or indicators. You can't create, edit, or delete relationships unless you have a TIM license. For more information, see View indicator relationships in an investigation.

#### 15.4.1 | Query indicators

##### Abstract

How to query indicators in the threat intel library (without a TIM license).

You can search for indicators using any of the available search fields. This is a partial list of the available search fields.

Field	Description
<code>type</code>	The type of the indicator, such as File or Email.
<code>verdict</code>	The reputation of the indicator: <ul style="list-style-type: none"> <li>• Malicious</li> <li>• Suspicious</li> <li>• Benign</li> <li>• Unknown</li> </ul>
<code>aggregatedReliability</code>	Searches for indicators based on a reliability score such as A - Completely reliable.
<code>sourceBrands</code>	Indicator feed or enrichment integrations.
<code>sourceInstances</code>	A specific instance of an indicator feed or enrichment integration.
<code>expirationSource</code>	The source (such as script or manual.) that last set the indicator's expiration status.
<code>tags</code>	Tags applied to indicators.
<code>comments</code>	Search for keywords within indicators' comments.

You can use a wildcard query, which finds indicators containing terms that match the specified wildcard. For example, the \* pattern matches any sequence of 0 or more characters, and ? matches any single character. For a regex query, use the following value:

```
"/.*\\?.*/"
```

#### 15.4.2 | View indicator relationships in an investigation

##### Abstract

How to use and create indicator relationships in Cortex XSOAR and how it benefits an investigation.

Indicator relationships are connections between different indicators. These relationships can be IP addresses related to one another, domains impersonating legitimate domains, etc. These relationships enable you to enhance investigations with information about indicators and how they might be connected to other incidents or indicators. For example, if you have a phishing incident with several indicators, one of those indicators might lead to another indicator, which is a malicious threat actor. Once you know the threat actor, you can investigate to see the incidents it was involved in, its known TTPs (tactics, techniques, and procedures), and other indicators that might be related to the threat actor. The initial incident which started as a phishing investigation immediately becomes a true positive and relates to a specific malicious entity.

Relationships are created from threat intel feeds and enrichment integrations that support the automatic creation of relationships, such as AlienVault OTX v2 and URLhaus, by selecting Create relationships in the integration settings. Based on the information that exists in the integrations, the relationships are formed.

You can view indicator relationships by clicking on the indicator from an incident, and then from the Quick View window click the Relationships tab.

#### NOTE:

To manage indicator relationships including how to create them, you need a TIM license. For more information, see [Manage indicator relationships](#).

## 16 | Threat Intel Management

### Abstract

Cortex XSOAR Threat Intel Management includes features such as Access to Threat Intel 42 data, investigate files using Sample Analysis, submit Sessions and Submissions, and deep dive into indicators.

Threat Intel Management enables you to unify the core components of threat intel, including threat intel aggregation, scoring, and sharing. Cortex XSOAR automates threat intel management by ingesting and processing indicator sources, such as feeds and lists, and exporting the enriched intelligence data to SIEMs, firewalls, and any other system that can benefit from the data.

### 16.1 | Get started with Threat Intel Management

#### Abstract

Learn how to use TIM in your investigation, utilizing Unit 42 Intel in your investigation.

Before diving in, you should understand the Cortex XSOAR Threat Intelligence Management's functionality and how it integrates with your needs. Review the use cases and key details to optimize your Cortex XSOAR experience from the start. Threat Intel management includes the following features:

- Access to Unit 42 Intel data
- Investigate files using sample analysis
- Submit Sessions and Submissions
- Manage Indicator Relationships
- Deep dive into an indicator on the Threat Intel page.
- Customize an indicator layout
- Manage TIM reports

#### NOTE:

Although some features are available without a TIM license such as indicator customization, you must have the Cortex XSOAR Threat Intel Management (TIM) license to use the TIM features.

### Licenses

Cortex XSOAR requires a yearly license per user. Multi-year licenses are available.

#### License usage

This table describes the types of Cortex XSOAR licenses which are used in the following circumstances:

Version	Usage	License
Cortex XSOAR (Enterprise) Edition	Built for customers who need a complete security automation solution.	Includes the SOAR Enterprise and TIM Enterprise licenses.

Version	Usage	License
Cortex XSOAR Threat Intel Management Edition	Built for Threat Intelligence and Security Operations teams who need threat intelligence-based automation.	Includes the TIM Enterprise license only.
Cortex XSOAR Starter Edition	Built for Security Operations and Incident Response customers who need case management with collaboration and playbook-driven automation.	Includes the SOAR Enterprise license only.

#### License quota

The following table describes the license quotas of each version in Cortex XSOAR.

XSOAR TIM (TIM Only)		XSOAR Starter Edition (SOAR Only)	XSOAR (SOAR + TIM)
Integrations	Unlimited	Unlimited	Unlimited
Incident Management	30-day history	Unlimited	Unlimited
Incident Triggered Automations	166 daily	Unlimited	Unlimited
Job Triggered Automations	Unlimited	Unlimited	Unlimited
Intel Feeds	Unlimited	5 active feeds, 100 indicators/fetch	Unlimited
Threat Intel Library	Unlimited	Intelligence detail view and relationships data are not included	Unlimited
Unit 42 Intelligence	Unlimited UI access, 5k/day API points	Not included	Unlimited UI access, 5k/day API points

#### NOTE:

Intel feed quotas are based on the selected Fetches Indicators field in the integration instance settings, not the enabled status. Disabling an integration instance does not affect the Intel feed quota. For example, if the AWS Feed is enabled and is fetching indicators and you don't want to include this in your quota, open the integration settings and clear the Fetches Indicators checkbox.

#### 16.1.1 | What is Threat Intel Management?

##### Abstract

Why use TIM with use cases.

The Cortex XSOAR native threat intel management capabilities allow you to unify the core components of threat intel, including threat intel aggregation, scoring, and sharing. Cortex XSOAR automates threat intel management by ingesting and processing indicator sources, such as feeds and lists, and exporting the enriched intelligence data to SIEMs, firewalls, and any other system that can benefit from the data. These capabilities enable you to sort through millions of indicators daily and take automated steps to make those indicators actionable.

#### NOTE:

Although some features are available without a TIM license such as indicator customization, you must have the Cortex XSOAR Threat Intel Management (TIM) license to use the TIM features.

## Why Threat Intel Management?

- Powerful native centralized threat intel

Supercharge investigations with instant access to a large repository of built-in, high-fidelity Palo Alto Networks threat intelligence crowdsourced from the largest footprint of network, endpoint, and cloud intel sources.

- Indicator relationships

Indicator connections enable structured relationships between threat intelligence sources and incidents. These relationships surface important context for security analysts on new threat actors and attack techniques.

- Hands-free automated playbooks with extensible integrations

Take automated action to shut down threats across over 600 third-party products with purpose-built playbooks based on proven SOAR capabilities.

- Granular indicator scoring and management

Take charge of your threat intel with playbook-based indicator lifecycle management and transparent scoring that can be easily extended and customized.

- Automated, multi-source feed aggregation

Eliminate manual tasks with automated playbooks to aggregate, parse, prioritize, and distribute relevant indicators in real-time to security controls for continuous protection

- Most comprehensive marketplace

The largest community of integrations with content packs that are prebuilt bundles of integrations, playbooks, dashboards, field subscription services, and all the dependencies needed to support specific security orchestration use cases. With a substantial amount of integrations and product integrations, you can buy intel on the go using Marketplace points.

## Threat Intel with Security orchestration

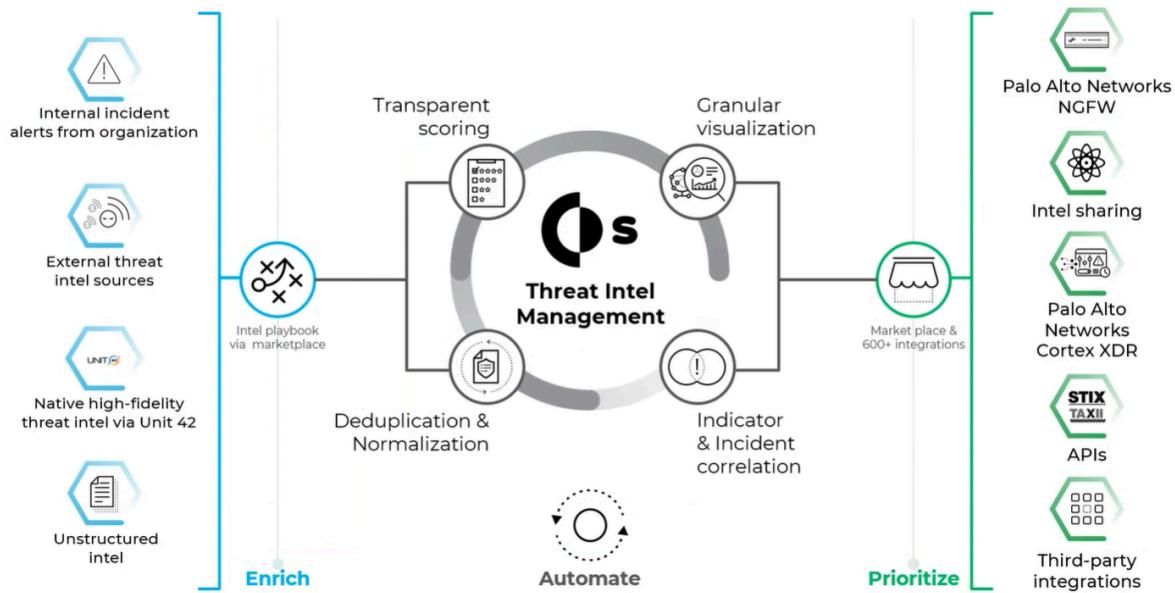
Security orchestration, automation, and response (SOAR) solutions have been developed to weave threat intelligence management into workflows by combining TIM capabilities with incident management, orchestration, and automation capabilities. SOAR solutions weave threat intelligence into a more unified and automated workflow. It matches alerts both to their sources and to compiled threat intelligence data and can automatically execute an appropriate response.

As part of the extensible Cortex XSOAR platform, TIM unifies threat intelligence aggregation, scoring, and sharing with playbook-driven automation. It empowers security leaders with instant clarity into high-priority threats to drive the right response across the entire enterprise.

Cortex XSOAR provides a common platform for incidents and threat information, where there is no disconnect between external threat data and your environment. Automated data enrichment of indicators provides analysts with relevant threat data to make smarter decisions.

Integrated case management allows for real-time collaboration, boosts operational efficiencies across teams, and automates playbooks to speed response across security use cases.

## Palo Alto Network's Cortex XSOAR Threat Intel Management



Cortex XSOAR collects data from sources such as incidents, Unit 42, and external threat intel feeds. After the data is ingested, Threat Intel playbooks examine the data proactively. The data gets deduped, normalized, and stored in the Threat Intel database so that a Threat Intel analyst can start a threat analysis. The analyst can then send that information to firewalls, share it with other stakeholders, and take remedial action as necessary.

### 16.1.2 | Threat Intel Management use cases

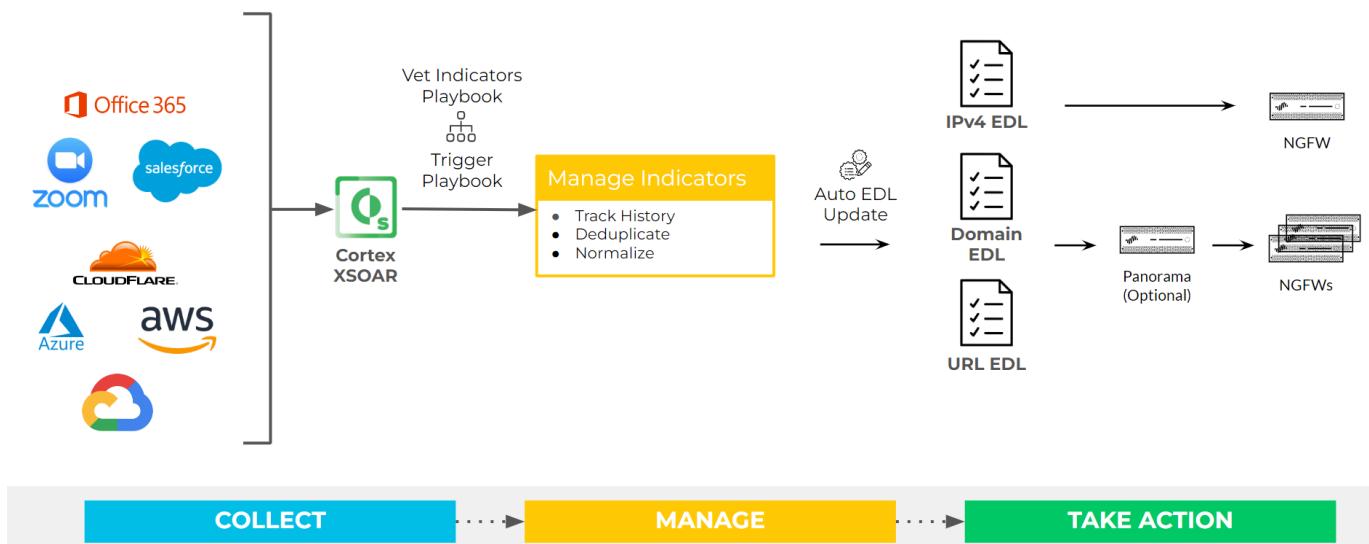
#### Abstract

Typical use cases for analysts and how to set up the use cases by administrators.

The following examples illustrate typical use cases for Threat Intel Management analysts, including how to configure playbooks and jobs for administrators.

#### Dynamic allow lists for business-critical SaaS apps

In this example, Firewall Admins are responsible for ensuring employees can always access SaaS applications such as Zoom and Office 365. They need to manage a stream of inbound change requests from the security team and other business units. Regardless of these daily changes, critical apps must always be allowed. The network infrastructure of SaaS applications is constantly changing/rotating IP addresses and Domains.



1. Configure a feed integration such as Office 365, Amazon AWS, Unit 42, etc.

1. Go to Settings & Info → Settings → Integrations → Instances and in the Category field, select Threat Intel Feeds.

2. Locate the relevant integration and select Add Instance.

In this example, add the AWS feed.

3. Set up the instance. In the Indicator Reputation field, select Benign.

4. Test and save the instance.

2. (Optional) Configure a playbook to filter indicators according to your requirements.

For example, the TIM - Indicator Auto Processing playbook identifies indicators that shouldn't be added to a block list, such as IP indicators that belong to business partners or important hashes you do not wish to process.

3. Go to Threat Intel page and run the following search to return IP, IPv6 or IPv6CIDR results:

```
sourceBrands:"AWS Feed" and expirationStatus:active and type:IP or type:IPv6 or type:IPv6CIDR
```

4. Configure the Generic Export Indicator Service integration.

1. In the Instances page, search for Generic Export Indicators Service and Add instance.

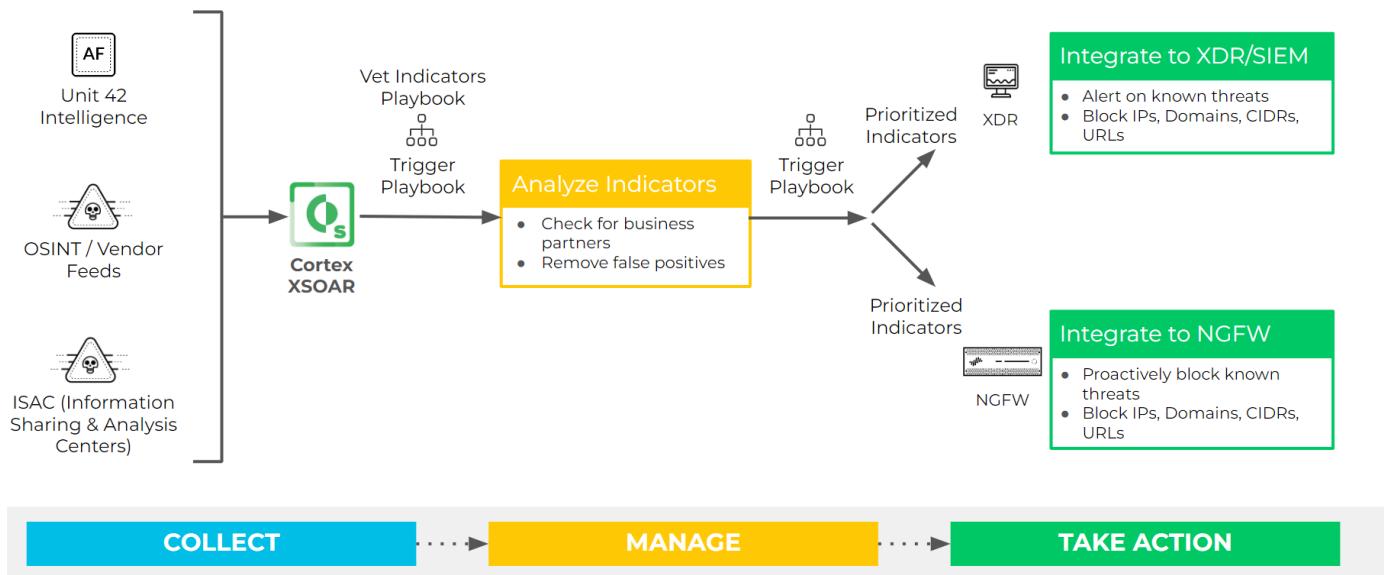
2. In the Indicator Query field, add the query in step 3.

3. Add the remaining fields, test, and save.

5. Test the EDL by running the Curl command: curl -v-u- user:pass https://ext-  
<tenant>crtx<region>.paloaltonetworks.com/xsoar/instance/execute/<instance-name>

#### Proactive blocking of known threats

The security team needs to leverage threat intelligence to block known bad domains, IPs, hashes, etc (indicators). The indicators are being collected from many different sources which need to be normalized, scored, and vetted (ensure not blocking business partners) before pushing to security devices such as Firewalls for blocking.



1. Configure feed integrations such as Unit 42 ATOMs feed, TAXII feed, etc.

1. Go to Settings & Info → Settings → Integrations → Instances and in the Category field, select Threat Intel Feeds.

2. Locate the relevant integration and select Add Instance.

3. Set up the instance.

In the Indicator Reputation field, blank.

4. Test and save the instance,

2. (Optional) Configure a playbook to filter indicators according to your requirements.

For example, the TIM - Indicator Auto Processing playbook identifies indicators that shouldn't be added to a block list, such as IP indicators that belong to business partners or important hashes you do not wish to process.

3. Go to the Threat Intel page and run the following search to return IP addresses with the verdict malicious with high reliability:

```
expirationStatus:active and type:IP and verdict:malicious and aggregatedReliability:A - Completely reliable
```

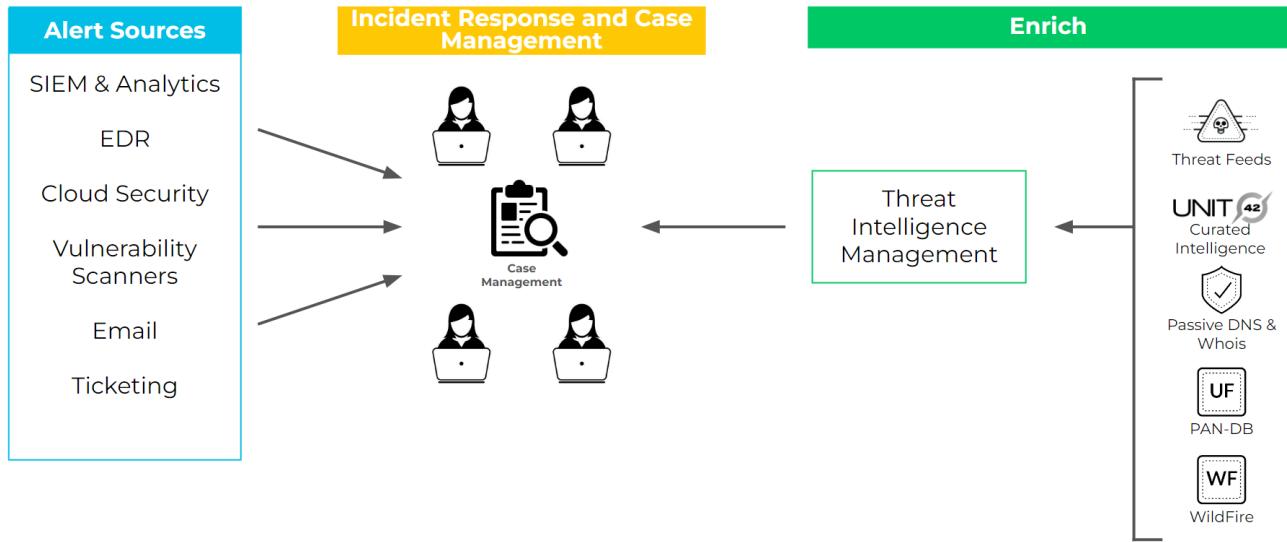
4. Configure the Generic Export Indicator Service integration.

1. In the Instances page, search for Generic Export Indicators Service and Add instance.
2. In the Indicator Query field, add the query in step 3.
3. Add the remaining fields, test, and save.
5. Test the EDL by running the Curl command: curl -v-u user:pass https://ext-<tenant>crtx<region>.paloaltonetworks.com/xsoar/instance/execute/<instance-name>

You can use this URL in your Next-Generation Firewall.

#### Incident enrichment

Incident Responders are receiving an endless stream of alerts, usually with little to no context of the external threat. Enriching alerts with curated threat intelligence from Unit 42 enables analysts to see the bigger picture and make more informed decisions when responding to alerts, ensuring comprehensive containment of the threat.



1. Use case management with Cortex XSOAR.
2. Ensure indicator extraction is enabled.
3. Configure threat feeds and enrichment sources, relevant to your use case. For example, Unit 42 ATOMs Feed, Feodo Tracker IP Blocklist Feed, TAXII Feed (to ingest ISAC data).

For example, configure the Palo Alto Networks Cortex XDR Investigation and Response integrations to ingest alerts from Cortex XDR. In the incidents page, open an incident from Cortex XDR. In the Case Info tab, you can see brief information, such as affected hosts, and affected users. In the Investigation tab, you can view the alert file artifacts or network artifacts. You can deep dive into the indicator by viewing the summary (verdict, sources, related incidents, timeline relationships, etc). In the Unit 42 Intel tab you can get additional details from Unit 42. For a file, you can see static and dynamic analysis. In the Work Plan, a playbook was run on whether an investigation is needed.

#### Weekly OSINT (Open Source Intelligence) Report

A new critical vulnerability is disclosed to the public which impacts the world's most popular applications (e.g. Log4J). The security team has already begun the search for the vulnerable software, however, the threat intel team needs to inform all technology employees of this critical threat. The intel team crafts a brief report summarizing the threat and adds analysis describing why this threat is relevant to the organization. This is also a great way to "advertise" the availability of threat intelligence services across the organization.



## COLLECT → MANAGE → TAKE ACTION

1. Ingest industry news events and security research blogs using the RSS Feed integration. E.g. threat post, Dark Reading, ZDNet security, Krebs on Security.
2. (Optional) Define any custom report types and templates.
3. Create a report.

In this example, you want to create a flash intel report about the Log4j security vulnerability, which will be sent to all internal stakeholders. You want to include the impact on the business with a brief analysis.

1. Get the relevant RSS feeds by going to fields by going to Threat Intel → Indicators and searching for sourceBrankds: RSS Feed log4j.
2. Start researching the Log4j issue by clicking the relevant entry.
3. Create a report by selecting Threat Intel Reports → New Threat Intel Report.
4. Complete the fields.

Each report type has different fields. After you create the report you can update all fields.

5. Create the report.
6. Edit the fields as required.

For example, you may want to add the RSS feeds to the relationship fields as well as the CVE file that it relates to.

7. In the Overview/Summary section, to use the Markdown editor, click M.

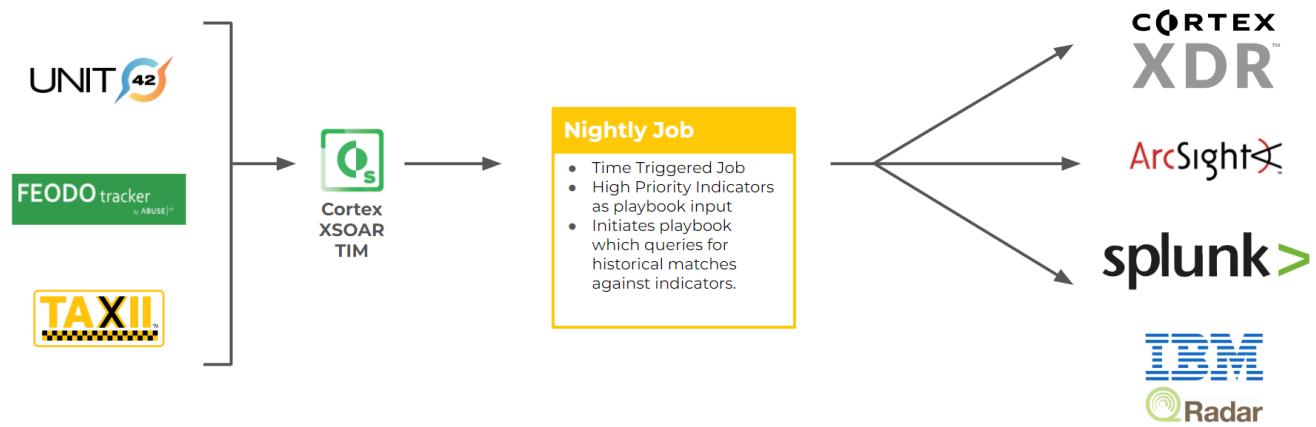
When finished, select Preview and then save.

8. (Optional) Mark the report for review and send it to one of your colleagues for review.
9. Publish the report which will be shared among a wider group.

You have the option to share it via PDF for a wider reach.

### Automated Exposure Check (Intel-Driven Threat Hunting)

The security team needs to perform due diligence, ensuring the organization has not been impacted by newly collected intelligence. Querying historical log data is a slow and tedious process for analysts (after acquisitions, organizations have multiple log stores). Additionally, running taxing historical queries is not possible during working hours, as compute resources are prioritized for SOC operations. The security team needs to automate this task during non-peak hours.



### COLLECT → MANAGE → TAKE ACTION

1. Configure feed integrations such as Unit 42 ATOMs feed, TAXII feed, etc.
2. (Optional) Configure a playbook to filter indicators according to your requirements.

For example, the TIM - Indicator Auto Processing playbook identifies indicators that shouldn't be added to a block list, such as IP indicators that belong to business partners or important hashes you do not wish to process.

3. Define the Triggered by delta in feed job to run that will trigger the playbook when the indicators are fetched.
4. To push the processed indicators to a SIEM, use the TIM - Add All Indicators Types to SIEM playbook.
5. Define a time-triggered job to push the indicators to the SIEM.

#### 16.1.3 | Indicator concepts

Before you start customizing and investigating you should be familiar with the following terms

##### Indicators

Indicators are artifacts associated with security incidents and are an essential part of the incident management and remediation process. They help correlate incidents, create hunting operations, and enable you to easily analyze incidents and reduce Mean Time to Response (MTTR).

##### Fetch indicators

Cortex XSOAR includes integrations that fetch indicators from either a vendor-specific source, such as TAXII, or from a generic source, such as a CSV or JSON file.

##### Indicator ingestion

Cortex XSOAR automates threat intel management by ingesting and processing indicator sources, such as feeds and lists, and exporting the enriched intelligence data to SIEMs, firewalls, and any other system that can benefit from the data. These capabilities enable you to sort through millions of indicators daily and take automated steps to make those indicators actionable in your security posture.

Indicators are added to Cortex XSOAR via the following methods:

Method	Description	Classification And Mapping
Integration	Feed integrations: Fetch indicators from a feed, for example, TAXII, Office 365, and Unit 42 ATOMS Feed.	Indicator classification and mapping is done in the Feed Integration and not in the Cortex XSOAR Settings & Info → Settings → Object Setup → Indicators → Classification & Mapping tab.
Indicator extraction	Indicators are extracted from selected incidents that flow into Cortex XSOAR, from an integration.	Only the value of an indicator is extracted, so no classification or mapping is needed.

Method	Description	Classification And Mapping
Manual	<ul style="list-style-type: none"> <li>Command line</li> <li>Mark: The user marks a piece of data as an indicator.</li> <li>STIX file: Manually upload a STIX file on the Threat Intel (Indicators) page.</li> </ul>	<p>Data is inserted manually via the UI so no classification or mapping is needed.</p> <p>If importing a STIX file, mapping is done via the STIX parser code.</p>

#### Common indicator data model

When indicators are ingested, regardless of their source, they have a unified, common set of indicator fields, including traffic light protocol (TLP), expiration, verdict, and tags.

#### Indicator smart merge

The same indicator can originate from multiple sources and be enriched with multiple methods (such as integrations, scripts, and playbooks). Cortex XSOAR implements a smart merge logic to make sure indicators are accurately scored (verdict) and aggregated. Indicator fields are merged according to the source reliability hierarchy. When there are two different values for a single indicator field, the field is populated with the value provided by the source with the highest reliability score. For multi-select and tag fields, new values are appended, rather than replacing the original values.

#### Indicators enrichment cache (Insightcache)

To avoid exceeding API quotas for third-party services, indicators are only updated after the cache expiration period. By default, the cache expires 4,320 minutes (3 days) after an indicator is updated, and cannot be cleared manually. The cache expiration can be set in the indicator type parameters. Indicator enrichment cache expiration only applies to automatic enrichment, triggered by the `enrichIndicators` command, and does not apply when you run reputation commands such as `!ip`.

#### Indicator timeline

The indicator timeline displays an indicator's complete history, such as the first-seen and last-seen timestamp and changes made to indicator fields.

#### Indicator expiration

When ingesting and processing many indicators daily, it's important to control whether or not they are active or expired and to define how and when indicators are expired. Cortex XSOAR offers multiple options to set indicator expiration.

#### Exclusion list

Indicators added to the exclusion list are disregarded by the system and are not created or involved in automated flows such as indicator extraction.

#### Jobs

Administrators can define a job to trigger a playbook when the specified feed or feeds finish a fetch operation that includes a modification to the list. The modification can be a new indicator, a modified indicator, or a removed indicator.

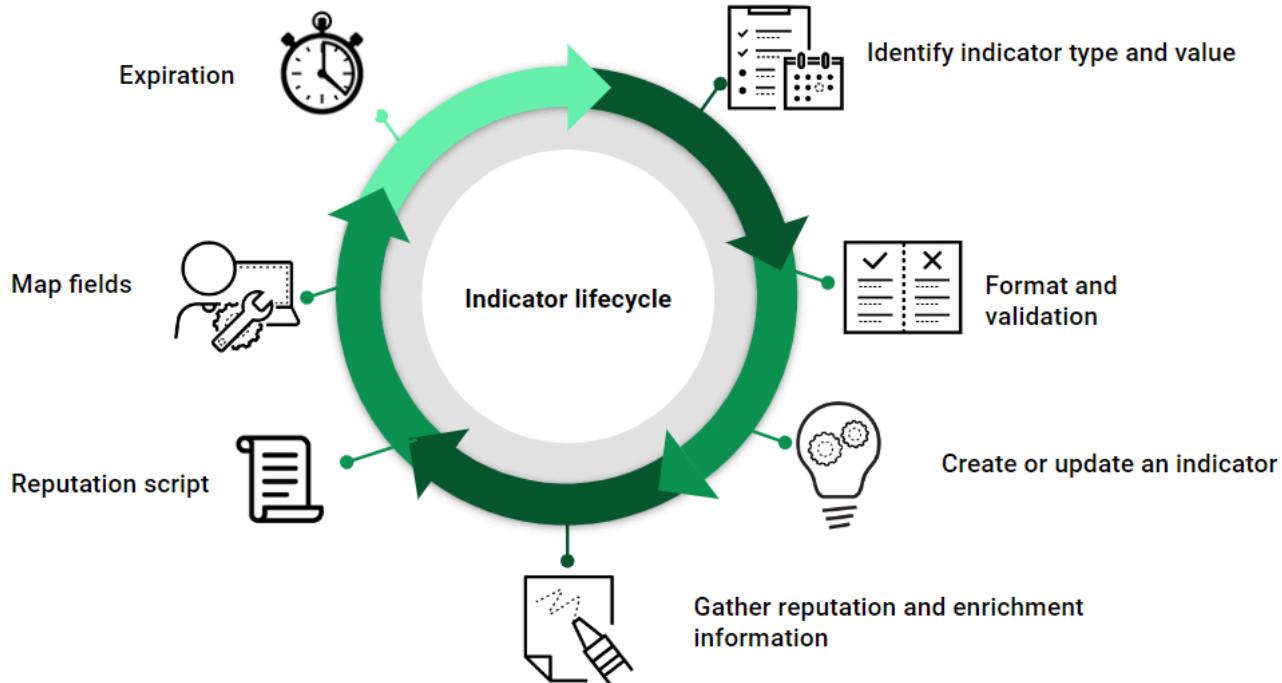
### 16.1.4 | Indicator lifecycle

#### Abstract

Indicators are artifacts associated with incidents and are an essential part of the incident management and remediation process.

Indicators are text-based artifacts associated with incidents, such as IP addresses, URLs, and email addresses, and are an essential part of the incident management and remediation process. They help correlate incidents, create hunting operations, and enable you to easily analyze incidents and reduce Mean Time to Response (MTTR).

The following diagram explains the indicator lifecycle in Cortex XSOAR.



Step	Details
1. Identify the indicator type and value	<p>Cortex XSOAR analyzes the text-based artifact and if it matches the indicator type profile. The indicator value is extracted, based on the indicator profile definition. You can set up indicator extraction automatically in the incident type, or playbook. Indicator extraction identifies indicators from various sources within Cortex XSOAR, such as email headers, IP addresses, email addresses, and file hashes in file attachments. For more information about indicator extraction, see <a href="#">Indicator extraction</a>.</p> <p>You can create or customize existing indicator types, fields, and layouts for your use case. For more information, see <a href="#">Customize indicator types, fields, and layouts</a>.</p>
2. Formatting and validation	<p>Formatting and validation of the indicator are done using a formatting script that validates the data that represents the indicator's value and determines how we want the data to appear in Cortex XSOAR. For example, the URL indicator type uses the <a href="#">FormatURL</a> script, which defangs URLs. For more information, see <a href="#">Formatting scripts</a>.</p>
3. Create or update an indicator	<p>If the indicator is not known to Cortex XSOAR, an indicator is created or you can create your own. If already known, it is updated with any new data including last seen dates. If the indicator is in an expired state but new data is received, it changes to active status.</p> <p>If you have a TIM license, you can add Unit 42 data by adding an indicator to Cortex XSOAR. For more information, see <a href="#">Query indicators with Unit 42 Intel data</a>.</p>
4. Gather reputation and enrichment information	<p>You can run reputation commands and enhancement script commands on indicator values. You need to set them to run in the indicator type. The enhancement script also runs on the indicator type. Both determine the indicator's verdict. For more information, see <a href="#">Enhancement scripts</a>.</p> <p>When a reputation command/enhancement script is run, the verdict gets added to the incident context, when attached to an incident. Generally, the information is found under the Dbot Score key, the specific Indicator type, and specific vendor information.</p> <p><b>NOTE:</b></p> <p>To run enhancement scripts and reputation commands, you must configure a relevant enrichment integration, such as VirusTotal, IPinfo v2, etc.</p> <p>You can exclude reputation commands from specific integrations in the indicator type settings if, for example, you are limited with API credits, or the integration is unreliable.</p>

Step	Details
5. Reputation scripts	Reputation scripts can be used if you want to override existing reputation commands with custom logic. For those indicator types without reputation commands, a custom reputation script can be applied. Use it to customize verdicts and DBotScore context entry. For more information, see Reputation scripts.
6. Map indicator fields	After your indicator is enriched, you can map fields. Some indicator fields are automatically mapped by Cortex XSOAR to contain the relevant values. The default settings can be changed for each indicator type. You can create and associate any custom fields with indicators. For more information, see Indicator classification and mapping.
7. Expiration	<p>Many indicators have expiration dates as threats are dynamic. IP addresses may change, systems may be fixed, etc. When configuring an indicator type, you can set it never to expire or after a time interval. For more information, see Configure indicator expiration.</p> <p><b>TIP:</b> We recommend defining your policy for handling expired indicators.</p>

### 16.1.5 | Roles and responsibilities in Threat Intel Management

#### Abstract

Roles and responsibilities in a Threat Intel Management environment.

A Threat Intel Management (TIM) analyst may have a different persona in the SOC. In some organizations, the TIM analyst is part of the SOC analyst's definition of work, but they have different workflows and use cases. The daily work of SOC analysts and TIM analysts are different.

Roles	Responsibility
Security Analyst (SOC Tier-1)	<ul style="list-style-type: none"> <li>• Triage Specialist</li> <li>• Monitor, manage, and configure security tools</li> <li>• Review incidents to assess their urgency</li> <li>• Escalate incidents when necessary</li> </ul>
Threat Intel Analyst (SOC Tier 2-3)	<ul style="list-style-type: none"> <li>• Incident responders and threat hunters</li> <li>• Remediation of escalated incidents from Tier 1 - investigation, response, and assessments</li> <li>• Proactive work to remove infrastructure weaknesses</li> </ul>

### 16.2 | Indicator configuration

#### Abstract

Create indicator types, fields, and layouts, customize the exclusion list, indicator reputation, and indicator extraction.

Customize your indicators to your specific needs. Edit existing indicator types, fields, and layouts, add scripts, and configure tailored extraction and expiration settings for optimal insights.

#### 16.2.1 | Customize indicator types, fields, and layouts

#### Abstract

Learn more about the options available for customizing indicators.

Cortex XSOAR provides out-of-the-box indicator types, fields, and layouts. However, you may need to customize indicators to suit your use case, either by editing existing indicator types, fields, or layouts or by creating new ones to help investigate and respond to potential security threats specific to your

organization.

Custom indicators can provide more accurate and efficient identification of potential cyber security threats. For example, you can customize indicators to monitor and detect unusual activity within your organization's internal network. This can include creating indicators to flag unauthorized access attempts or unusual data transfers, or identifying insider threats or compromised accounts.

Before customizing an indicator, review the ingested indicator and then customize it as needed. After ingesting incidents and indicators, check the indicator information associated with your incident. From an incident, review the context data (from Side panels). If there is information in the context data that you don't see in the indicator, map it into indicator fields and display it in the layout.

You can customize the following:

Option	Description
Indicator type	Customize an indicator type by setting the relevant fields, display layout, scripts to run, and reputation command for the indicator type. You can create a new indicator type or you can edit an out-of-the-box indicator type. For more information, see Create an indicator type.
Indicator fields	Custom indicator fields add specific details or attributes to indicators, helping to better classify and understand the nature of potential security threats. You can edit an existing indicator field or create a new one. After creating a new indicator field, map the field to the relevant context data. You can add the field to an indicator type and view it in an indicator layout.  For more information, see Create an indicator field.
Indicator layout	Custom indicator layouts enable you to organize and display specific details about potential threats in a way that makes sense for your organization, making it easier to quickly understand and respond to security issues. You can view, customize, import and export indicator layouts as well as add a custom layout to an indicator type.  <b>NOTE:</b> If you do not have a TIM license, you cannot edit the Indicator Summary layout. You can only edit the Quick view and the New/Edit form tabs.  For more information, see Indicator layout customization.

#### 16.2.1.1 | Create an indicator type

##### Abstract

In addition to the system-level indicator types, you can create custom indicator types in Cortex XSOAR.

Indicators are categorized by indicator type, which determines the indicator layout and fields that are displayed and which scripts are run on indicators of that type. Cortex XSOAR includes several out-of-the-box indicator types, such as:

- IP Address
- Domain
- URL
- File

For more information about file indicators and how to configure the file hash, see File indicators.

When you create a new indicator type, you define its properties, including whether and how to format the indicator data and how the verdict is calculated.

1. Go to Settings & Info → Settings → Object Setup → Indicators → Types.
2. Click New.
3. In the Settings tab, add the required indicator profile, such as name and Regex.  
  
For more information, see Indicator type profile.
4. In the Custom Fields tab, map the fields, as required.  
  
For more information, see Map custom indicator fields.

See this video for an example of creating a custom indicator type.

Terjadi error.

Cobalah menonton video ini di [www.youtube.com](http://www.youtube.com), atau aktifkan JavaScript jika dinonaktifkan di browser Anda.

#### Example 27. Create a company email indicator type

The following example describes how to create a new indicator type to manage employee emails, for example for resource management or inside threat investigation.

Create a new indicator type for the employee email addresses which contain the "our\_company.com" company domain.

1. Under Settings & Info → Settings → Object Setup → Indicators → Types → New, in the Settings tab, define the following.

- Name: Company email
- Regex: `.*?@our_company.com` (simplified to capture all the email addresses using the our\_company.com domain).
- Reputation command: Not relevant for this example, since we don't want any external enrichment.
- Formatting script: If more formatting is needed, you can use a formatting script to edit the saved value.
- Reputation script: If needed, you can create a reputation script to affect the DBot score given to the new custom indicator.

2. In the Custom Fields tab, map custom fields for the new indicator type.

You can map fields returned using an integration such as Active Directory to obtain more data about the actual user to whom the email belongs. You can also collect data using integrations such as Okta (MFA, SSO), SIEM, and email security. Fields such as Username, Full name, and various groups the user is part of as well as other identifiers are returned to context and mapped into the indicator using the custom fields.

Field	Mapping
Action	<a href="#">Choose data path</a>
Actor	<a href="#">Choose data path</a>
Architecture	<a href="#">Choose data path</a>
Assigned role	<a href="#">Choose data path</a>
Assigned user	<a href="#">Choose data path</a>
Associations	<a href="#">Choose data path</a>
Behavior	<a href="#">Choose data path</a>
Blocked	<a href="#">Choose data path</a>
CPE	<a href="#">Choose data path</a>
CVSS Score	<a href="#">Choose data path</a>

Indicator Sample: [Enter an indicator](#)

[Load](#) [Cancel](#) [Save](#)

#### NOTE:

If you miss mapping any field, you can create additional new indicator fields and either relate them to all indicator types, or relate them only to the new indicator type (recommended).

3. Design a custom layout for the new indicator type.

You can use the Dynamic section in the indicator layout to run python scripts and return results from within the layout itself.

Each indicator type has its own profile that enables Cortex XSOAR to recognize it across the platform. During the indicator extraction flow, the order of execution is regex, formatting script, reputation command, and reputation script. You can update the following fields when updating an indicator type.

Field	Description
Name	A meaningful name for the indicator type.
Reputation script	<p>The output of the reputation script is a verdict score, which is used as the basis for the indicator verdict. Reputation scripts must be tagged reputation to appear in the list for the indicator type. For more information, see Reputation scripts</p> <p>The results of reputation scripts do not print to the War Room in the extraction flow.</p>
Formatting script	<p>Modifies how the indicator displays in Cortex XSOAR.</p> <p>Formatting scripts must be tagged indicator-format to appear in the list for the indicator type. For more information, see Formatting scripts.</p>
Enhancement script	<p>The enhancement script is not part of the indicator extraction flow and is run manually on the indicator type. Examples of enhancement scripts include an enrichment script and a script that runs a search in an SIEM for the indicator.</p> <p>After indicators are identified, you can go to the Indicator Quick View page, click the Actions button, and run an enhancement script directly on an indicator. For these scripts to be available in the menu, they need the enhancement tag. For more information, see Enhancement scripts.</p> <p>When you run an enhancement script, it is the equivalent of running the script in the CLI. The script can write to context, return an entry, etc.</p>
Reputation command	<p>Calculates the reputation of indicators of this type. The verdict (reputation) is only associated with the specific indicator value on which it's run (not the indicator type). The command returns the reputation of the indicator value as an entry with entry context and in some cases also returns context values that can be mapped to the indicator type custom fields.</p> <p>The results of the reputation command do not print to the War Room in the indicator extraction flow. For more information, see Reputation commands.</p>
Regex	The regular expression (regex) to identify indicators for this indicator type.
Layout	Select the indicator layout to use.
Exclude these integrations for the reputation command	Integrations to exclude when calculating the verdict, evaluating, and enriching indicators of this indicator type. This only applies to the indicator extraction and enrichment mechanism and does not apply when directly running reputation commands such as !ip, !url, !domain, etc.
Indicator Expiration Method	<p>The method by which to expire indicators of this type. The expiration method that you select is the default expiration method for indicators of this indicator type.</p> <p>The expiration can also be assigned when configuring a feed integration instance, which overrides the default method.</p> <ul style="list-style-type: none"> <li>• Never Expire: indicators of this type never expire.</li> <li>• Time Interval: indicators of this type expire after the specified number of days or hours. For more information, see Configure indicator expiration.</li> </ul>

Field	Description
Context path for verdict value (Advanced)	When an indicator is extracted, the entry data from the command is mapped to the incident context. This path defines where in context the data is mapped.
Context value of verdict (Advanced)	The value of this field defines the actual data that is mapped to the context path.
Cache expiration in minutes (Advanced)	<p>The amount of time (in minutes) after which the cache for indicators of this type expire. The default is 4,320 minutes (three days). The cache enables you to limit API requests by only updating indicators after a specific time period has passed. The cache cannot be cleared manually.</p> <p><b>NOTE:</b></p> <p>Indicator cache expiration rules only apply to standard enrichment (for example, running the <code>enrichIndicators</code> command). If you run a reputation command, such as <code>!ip</code>, the command executes even if the cache has not expired.</p>

16.2.1.1.2 | [File indicators](#)**Abstract**

You can have a single file indicator for file objects in Cortex XSOAR or each file can have a hash as its own indicator.

Cortex XSOAR uses a single File indicator for file objects. As a result, files that appear with their SHA256 hash and all other hashes associated with the file, (MD5, SHA1, and SSDeep) are listed as properties of the same indicator. In addition, when ingesting an incident through an integration, all file information is presented as one object.

When investigating an incident, in the Indicators field (Investigation or Case info tabs), click a File indicator. You can see additional information for that indicator, including:

- SHA256
- MD5
- SHA1
- SSDeep
- Associated File Names

The `File.Name` values associated with the indicator hash, based on `File` context objects created in Cortex XSOAR (automatically populated).

- Modified

The date and time the `File` indicator was last modified.

- First Seen

The date and time the file was first seen in Cortex XSOAR.

If the file appears in a different incident with a different name and has any of the same hash values, it automatically associates with the original indicator.

**NOTE:**

A new File indicator only affects new indicators ingested to the Cortex XSOAR platform. Indicators that were already in Cortex XSOAR continue to appear as their respective hash-related indicators.

Configure each file hash to appear as a separate indicator

By default, Cortex XSOAR uses a single file indicator for file objects. As a result, files that appear with their SHA256 hash and all other hashes associated with the file, (MD5, SHA1, and SSDeep) are listed as properties of the same indicator. In addition, when ingesting an incident through an integration, all file information is presented as one object.

If the file appears in a different incident with a different name, and has any of the same hash values, it automatically associates with the original indicator.

If you want to have each file hash appear as its own indicator, do the following:

1. Go to Settings & Info → Settings → Objects Setup → Indicators → Types.
2. Select the File indicator and click Disable.

3. Select the following required hashes:

- File SHA-256
- File SHA-1
- File MD5
- SSDeep

4. Click Enable.

The file indicator merging method

When a file is created in the system, whether from a feed, indicator extraction or manually added, its original value is created as the indicator's value, while its complementing hashes are saved as fields.

For example, if a SHA256 indicator is extracted from an email and enriched, an indicator with the SHA256 hash as the value will be created, and any other hash that is found in the enrichment phase (such as MD5, SHA1) will be added as a field. If in the future a file indicator with the same MD5 is created in the system, Cortex XSOAR automatically identifies it and merges the two indicators together into one.

For example, the executable cmd.exe's SHA256 **FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5** was found in an incident and extracted. It also went through enrichment, which provided the information that the file's MD5 is **D7AB69FAD18D4A643D84A271DFC0DBDF**.

The file indicator includes:

```
ID: 1
Type: File
Value: FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5
SHA256: FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5
MD5: D7AB69FAD18D4A643D84A271DFC0DBDF
```

Afterwards, through a custom feed, the cmd.exe's MD5 **D7AB69FAD18D4A643D84A271DFC0DBDF** hash is brought in, and Cortex XSOAR creates an indicator of type File with the MD5 hash as its value.

A new file indicator is created:

```
ID: 2
Type: File
Value: D7AB69FAD18D4A643D84A271DFC0DBDF
MD5: D7AB69FAD18D4A643D84A271DFC0DBDF
```

The automatic merging flow for the File indicator type identifies that the two indicators are the same file and merges them together.

The final file indicator is a consolidation of the two, and is the same as the first example above:

```
ID: 1
Type: File
Value: FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5
SHA256: FF79D3C4A0B7EB191783C323AB8363EBD1FD10BE58D8BCC96B07067743CA81D5
MD5: D7AB69FAD18D4A643D84A271DFC0DBDF
```

#### 16.2.1.1.3 | Formatting scripts

##### Abstract

Formatting scripts validate input and modify how indicators are displayed.

A formatting script has the following main functions:

- Validate inputs, for example, to check that the top-level domain (TLD) is valid.
- Modify how the indicator appears in Cortex XSOAR such as the War Room.

After indicator values are extracted according to the defined regex, the formatting script can be used to modify how the indicator value appears in the War Room and reports. For example, the IP indicator type uses the **UnEscapeIPs** formatting script, which removes any defanged characters from an IP address, so **127[.]0[.]0[.]1** is formatted to **127.0.0.1**. When you click the IP address in the War Room, you see the formatted IP address. This extracted indicator value is then added to the Threat Intel database.

##### Out-of-the-box Formatting Scripts

You can create a new script, or you can use an out-of-the-box formatting script on the Scripts page, for example:

- **UnEscapeIPs:** Removes escaping characters from IP addresses. For example, 127[.]0[.]0[.]1 transforms to 127.0.0.1.
- **ExtractDomainAndFQDNFromUrlAndEmail:** Extracts domains and FQDNs from URLs and emails, used by the Domain indicator. It removes prefixes such as proofpoint or safelinks, removes escaped URLs, and extracts the FQDN.
- **ExtractEmailV2:** Verifies that an email address is valid and only returns the address if it is valid.

#### Formatting Script example

In the following example, the RemoveEmpty script removes empty items, entries, or nodes from an array.

```
// pack version: 1.2.30
const EMPTY_TOKENS = args.emptyList(args.empty_values);

function toBoolean(value) {
    if (typeof(value) === 'string') {
        if (['yes', 'true'].indexOf(value.toLowerCase()) != -1) {
            return true;
        } else if (['no', 'false'].indexOf(value.toLowerCase()) != -1) {
            return false;
        }
        throw 'Argument does not contain a valid boolean-like value';
    }
    return value ? true : false;
}

function isObject(o) {
    return o instanceof Object && !(o instanceof Array);
}

function isEmpty(v) {
    return (v === undefined) ||
           (v === null) ||
           (typeof(v) == 'string' && (!v || EMPTY_TOKENS.indexOf(v) !== -1)) ||
           (Array.isArray(v) && v.filter(x => !isEmpty(x)).length === 0) ||
           (isObject(v) && Object.keys(v).length === 0);
}

function removeEmptyProperties(obj) {
    Object.keys(obj).forEach(k => {
        var ov = obj[k];
        if (isObject(ov)) {
            removeEmptyProperties(ov);
        } else if (Array.isArray(ov)) {
            ov.forEach(av => isObject(av) && removeEmptyProperties(av));
            obj[k] = ov.filter(x => !isEmpty(x));
        }
        if (isEmpty(ov)) {
            delete obj[k];
        }
    });
}

var vals = Array.isArray(args.value) ? args.value : [args.value];

if (toBoolean(args.remove_keys)) {
    vals.forEach(v => isObject(v) && removeEmptyProperties(v));
}
return vals.filter(x => !isEmpty(x));
```

#### Formatting Script input

The formatting script requires a single input argument named **input** that accepts a single indicator value or an array of indicator values. The input argument should be an array to accept multiple inputs and return an entry-result per input.

Argument	Description
<b>input</b>	Accepts a string or array of strings representing the indicator value(s) to be formatted. Will be accessed within the script using <code>demisto.args().get('input', [])</code> .  In the script settings, the Is Array checkbox must be selected (see screenshot below). The script code must be able to handle a single indicator value (as string), multiple indicator values in CSV format (as string) and an array of single indicator values (array).



#### Formatting Script outputs

The indicators appear in a human-readable format in Cortex XSOAR. The output should be an array of formatted indicators or an array of entry results (an entry result per indicator to be created). The entry result per input can be a JSON array to create multiple indicators. If the entry result is an empty string, it is ignored and no indicator is created.

Use the `return_results` function to generate the script output. For more information, see [https://xsoar.pan.dev/docs/integrations/code-conventions#return\\_results](https://xsoar.pan.dev/docs/integrations/code-conventions#return_results).

#### Single-value result:

```
results = CommandResults(
    outputs_prefix='VirusTotal.IP',
    outputs_key_field='Address',
    outputs={
        'Address': '8.8.8.8',
        'ASN': 12345
    }
)
return_results(results)
```

#### Multiple-value results:

```
results = [
    CommandResults(
        outputs_prefix='VirusTotal.IP',
        outputs_key_field='Address',
        outputs={
            'Address': '8.8.8.8',
            'ASN': 12345
        }
    ),
    CommandResults(
        outputs_prefix='VirusTotal.IP',
        outputs_key_field='Address',
        outputs={
            'Address': '1.1.1.1',
            'ASN': 67890
        }
    )
]
return_results(results)
```

#### Add a Formatting Script to an indicator type

1. Go to Settings & Info → Settings → Indicators → Types.
2. Select the indicator type and click Edit.
3. Select the desired formatting script.

#### NOTE:

Formatting scripts must have the `indicator-format` tag to appear in the list.

#### NOTE:

Formatting scripts for out-of-the-box indicator types are system-level, which means that the formatting scripts for these indicator types are not configurable. To create a formatting script for an out-of-the-box indicator type, you need to disable the existing indicator type and create a new (custom) indicator type. If you configured a formatting script before this change and updated your content, this configuration reverts to content settings (empty).

#### Run a Formatting script in the CLI

You can run out-of-the-box or custom formatting scripts in the CLI to check the extracted indicator data is properly formatted.

The following are examples of the syntax for running the out-of-the-box UnEscapeIPs formatting script in the CLI.

- !UnEscapeIPs !UnEscapeIPs input=127.0.0[.]1
- !UnEscapeIPs input=127.0.0[.]1,8.8.8[.]8
- !UnEscapeIPs input=\${contextdata.indicators} (where the key contextdata.indicators in the context object is an array)

## 16.2.1.1.4 | Enhancement scripts

## Abstract

Enhancement scripts are run manually and can enrich indicators, write to context, and return entries to the War Room.

Enhancement scripts enable you to gather additional data about the highlighted entry in the War Room. They can enrich indicators, search a SIEM for a specific indicator, write indicator details to context, and return entries to the War Room.

Enhancement scripts are run manually from the Indicator Quick View window or the CLI after indicators are extracted to allow you to collect additional information about an indicator. If you have an incident that contains an IP indicator and you want to run one or more enhancement scripts, go to Indicator Quick View → Actions and under Run Scripts, select the desired script.

**NOTE:**

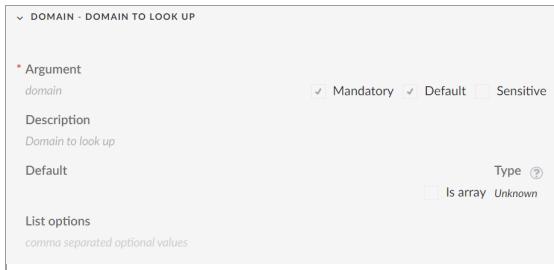
Enhancement scripts are different from reputation commands. A reputation command runs every integration that has that command within it, to enrich the indicator. The reputation command `ip`, for example, runs every IP integration command in your enabled integrations, to collect data from multiple sources. An enhancement script is manually run after the initial extraction and enrichment for the indicator type is complete.

## Enhancement script input

The enhancement script requires the indicator value as the input argument.

Argument	Description
The value of the indicator	For example <code>ip</code> , <code>email</code> , <code>url</code> . The argument name should match the indicator type in lower case. For example, the <code>IPReputation</code> script requires the <code>ip</code> input. For an <code>EmailReputation</code> script the input is <code>email</code> .

In the following example, the `DomainReputation` script uses `domain` as the input.



## Enhancement script outputs

The enhancement script output depends on its input because the script is run manually. If you want the output to be added to indicator enrichment or the Threat Intelligence screen, it should follow the DBotScore convention in the content output as described in <https://xsoar.pan.dev/docs/integrations/dbot>.

```
output =
{
    'Type': entryTypes['note'],
    'ContentsFormat': formats['json'],
    'Contents': 'this is the enrichment data',
    'EntryContext': {
        'Email': 'xsoar@test.com',
        'DBotScore': {}
    }
}

return_results(output)
```

## Add an enhancement script to an indicator type

1. Go to Settings & Info → Settings → Indicators → Types.
2. Select the indicator type and click Edit.
3. Select one or more desired enhancement scripts.

**NOTE:**

Enhancement scripts must have the `enhancement` tag applied to appear in the list.

## Run an enhancement script in the CLI

You can run out-of-the-box or custom enhancement scripts in the CLI to enrich specific indicator values.

The following are examples of the syntax for running the out-of-the-box IPReputation and URLReputation enhancement scripts in the CLI.

- !IPReputation ip=8.8.8.8
- !URLReputation url=cardcom.com

#### 16.2.1.1.5 | Reputation scripts

##### Abstract

Reputation scripts for indicator enrichment.

Reputation scripts are used to assess and assign reputation scores to indicators. These scripts integrate external threat intelligence or internal data sources to evaluate the reputation of indicators (such as IP addresses, URLs, or file hashes). Reputation scripts enable you to implement custom logic and algorithms for determining the reputation of indicators.

Reputation scripts return the verdict of an indicator as a number. The number overrides the verdict returned from the reputation command but does not override a manually set verdict. The reliability of the score from a reputation script is by default A++ - Reputation script.

You can modify the reliability by navigating to Settings & Info → Settings → System → Server Settings → + Add Server Configuration and adding the server configuration enrichment.reputationScript.reliability with the desired reliability score.

##### NOTE:

The Reputation script overrides any default settings for the indicator that relates to the verdict.

##### Out-of-the-box reputation scripts

You can create a new reputation script, or you can use an out-of-the-box reputation script in the Scripts page, for example:

- CertificateReputation
- cveReputation
- MaliciousRatioReputation
- SSDeepReputation

##### Reputation Script input

The reputation requires a single input argument named `input` that accepts an indicator value.

Argument	Description
input	The indicator value.



##### Reputation Script outputs

Either a number or a dbotScore. It can either be a raw number which is the score, or a full entry with DBotScore.

```
from CommonServerPython import *

def main():
    url_list = argToList(demisto.args().get('input'))
    entry_list = []

    for url in url_list:
        entry_list.append({
            'Type': entryTypes['note'],
            'ContentsFormat': formats['json'],
            'Contents': 2,
```

```

'EntryContext': {
    'DBotScore': {
        'Indicator': url,
        'Type': 'Onion URL',
        'Score': 2, # suspicious
        'Vendor': 'DBot'
    }
}
}

demisto.results(entry_list)

if __name__ in ('__main__', 'builtin', 'builtins'):
    main()

```

Values for Common.DbotScore

Constant	Value
Common.DbotScore.NONE	NONE = 0
Common.DbotScore.GOOD	GOOD = 1
Common.DbotScore.SUSPICIOUS	SUSPICIOUS = 2
Common.DbotScore.BAD	BAD = 3

Add a Reputation Script to an indicator type

1. Go to Settings & Info → Settings → Object Setup → Indicators → Types.
2. Select the indicator type and click Edit.
3. Select the relevant reputation script.

**NOTE:**

Reputation scripts must have the reputation tag applied to appear in the list.

Run a Reputation Script in the CLI

You can run out-of-the-box or custom reputation scripts in the CLI to set the verdict for a specific indicator.

The following are examples for running the out-of-the-box CertificateReputation and MalicioiusRationReputation reputation scripts in the CLI.

- !CertificateReputation input=<value of the indicator>
- !MalicioiusRationReputation input=<value of the indicator>

#### 16.2.1.1.6 | Reputation commands

##### Abstract

Reputation commands run based on the indicator type and return a verdict for the indicator.

Reputation commands are built-in or custom commands that use integrations such as Unit 42 to provide predefined functionalities for obtaining an indicator verdict for specific indicator types. These commands simplify the process of fetching reputation data from external services or threat intelligence feeds without requiring extensive scripting. Reputation commands come with preconfigured parameters and settings for commonly used threat intelligence sources.

You can set an indicator type to run reputation commands. The command returns the verdict of the indicator as an entry with entry context and may also return context values that can be mapped to the custom fields of the indicator.

**NOTE:**

Running a reputation command directly (such as !ip) might not apply the result to an indicator, nor does it use the enrichment cache. To ensure an indicator is enriched, and to take advantage of caching, use the enrichIndicators command or the Enrich button in the UI. This runs the appropriate reputation command/script based on the indicator type settings. Note that extracted indicators are enriched in the same way.

##### Out-of-the-box reputation commands

You can create a new reputation command, or you can use an out-of-the-box reputation command, for example:

- ip
- file
- url
- email
- domain

For more details on using out-of-the-box reputation commands or developing new reputation commands, see [Generic Commands Reputation](#).

#### Reputation command input

The reputation command uses the indicator value as the input argument.

Arguments	Description
The value of the indicator	<p>For example <code>ip</code>, <code>email</code>, <code>url</code>. Inputs are based on different integrations. Basic inputs are common to all reputation commands.</p> <p>For example, the <code>!ip</code> command has the following basic inputs:</p> <pre>- name: ip   arguments:     - name: ip       default: true       description: List of IPs.       isArray: true</pre>

In this example, the `ip` script uses `ip`, as the input, with the `is array` field checked.



#### Reputation command output

Outputs return a dbotScore.

#### Run a Reputation command in the CLI

The following are examples of the syntax for running the `ip`, `domain`, and `file` reputation commands in the CLI.

- `!ip ip=<value of the indicator>`
- `!domain domain=<value of the indicator>`
- `!file file=<value of the indicator>`

#### 16.2.1.1.7 | Map custom indicator fields

##### Abstract

Learn more about mapping custom indicator fields.

Indicator mapping enables you to automatically update the value of an indicator field without having to manually change it. For example, the IP indicator automatically maps the Country field. If it was not mapped, each time the IP address changes country the analyst would have to update the country every time that indicator type is ingested.

The value of an indicator field is determined by the value of the key in context data the field is mapped to in Cortex XSOAR.

When you start ingesting indicators, the incoming fields are automatically mapped to the relevant indicator fields. Sometimes you may want to change the default settings or map custom indicator fields to specific context data. Before you map custom indicator fields, you need to create the indicator field and add it to the relevant indicator type layout.

**NOTE:**

Some integrations have indicator mappers and classifiers, such as AWS. If you want to use an integration mapper or classifier, see Indicator classification and mapping.

To map custom fields to the indicator type, you need to enrich the indicator either by using the `!enrichindicators` command in the CLI, in a playbook, or by opening an indicator and click Enrich indicator. Enrichment returns an entry, with the `EntryContext` property as the source of the mapping process. When editing an indicator type, in the Custom Fields tab, type the name of the indicator exactly how it appears (in the Threat Intel page) and click Load.

For the enrichment data to be considered valid, `EntryContext` must include a `DBotScore` with the fields: `Indicator`, `Score`, `Vendor`, and `Type`. If `DBotScore` has those fields, all the data of `EntryContext` is used as the source for the mapping, and not only the data under `EntryContext.DBotScore`.

How to map indicator fields

1. Go to Settings & Info → Settings → Object Setup → Indicators → Types

2. Select the indicator type and click Edit.

3. Click the Custom Fields tab.

The custom fields associated with this indicator type are listed in the table. If you do not see a custom field in the list, verify that you associated the custom field with this indicator type.

4. (Optional) In the Indicator Sample panel, enter an indicator relevant to the indicator type to load sample data.

5. Click Choose data path to map the custom field to a data path.

- a. (Optional) Click the curly brackets to map the field to a context path.

- b. (Optional) From the Indicator Sample panel, select a context key to map to the field.

6. Save the indicator type.

#### 16.2.1.2 | Create an indicator field

##### Abstract

Create a new indicator field in the Fields tab in Cortex XSOAR. Add specific indicator information to incidents.

Indicator fields are used to add specific indicator information to incidents. When you create an indicator field, you can associate the field to a specific indicator type or all indicator types. You can then map the custom field to the relevant indicator type. You can also add an indicator field trigger script.

**NOTE:**

Cortex XSOAR IOC fields are based on the STIX 2.1 specifications. For more information, see Indicator field structure.

##### Field types

Field Type	Description
Boolean	Checkbox
Date picker	Adds the date to the field.
Grid (table)	<p>Include an interactive, editable grid as a field type for selected indicator types or all indicator types.</p> <p>To see how to create a grid field and to use a script, see Add an indicator field trigger script to an indicator field.</p> <p>When you select Grid (table) you can format the table and determine if the user can add rows.</p>
HTML	Create and view HTML content, which can be used in any type of indicator.

Field Type	Description
Long text	<ul style="list-style-type: none"> <li>Long text is analyzed and tokenized, and entries are indexed as individual words, enabling you to perform advanced searches and use wildcards.</li> <li>Long text fields can't be sorted and used in graphical dashboard widgets.</li> <li>While editing a long text field, pressing enter will create a new line (case is insensitive).</li> </ul> <p>Add a placeholder, if required.</p>
Markdown	<p>Add markdown formatted text as a template, which will be displayed to users in the field after the indicator is created. Markdown lets you add basic formatting to text to provide a better end-user experience.</p>
Multi select/Array	<p>Select the following options:</p> <ul style="list-style-type: none"> <li>Multi-select from a prefilled (static) list</li> <li>An empty array field for the user to add one or more values as a comma-separated list</li> </ul> <p>Add a placeholder, if required.</p>
Number	<p>Can contain any number. Default is 0.</p>
Role	<p>The role assigned to the indicator. Determines which users (by role) can view the indicator.</p>
Short text	<ul style="list-style-type: none"> <li>Short text is treated as a single unit of text and is not indexed by word. Advanced search, including wildcards, is not supported.</li> <li>Short text fields are case-sensitive by default but can be changed to case-insensitive when creating the field.</li> <li>While editing a short text field, pressing enter will save the change.</li> <li>Maximum length 60,000 characters</li> </ul> <p>Recommended use is one-word entries, such as username and email address.</p> <p>Select a placeholder, if required.</p>
Single select	<p>Select a value from a list of options. Add comma-separated values.</p>
Tags	<p>Accepts a single tag or a comma-separated list, not case-sensitive.</p> <p>Add a placeholder, if required.</p>
URL	<p>Add a URL when completing the field.</p>
User	<p>A user in Cortex XSOAR.</p>

#### How to create a field

1. Select Settings & Info → Settings → Object Setup → Indicators → Fields → New Field.
2. Select the relevant field type.
3. Complete the following fields (if relevant):

Parameter	Description
Mandatory	If selected, this field is mandatory when used in a form.
Field Name	A meaningful display name for the field. After you type a name, you will see below the field that the Machine name is automatically populated. The field's machine name is applicable for searching and the CLI.
Tooltip	An optional tooltip for the field.

4. In the Basic Settings tab, define the values (according to the selected field type).

5. In the Attributes tab define the following:

Field	Description
Script to run when field value changes	The script dynamically changes the field value when script conditions are met. For a script to be available, it must have the <b>field-change-triggered-indicator</b> tag when defining the script. For more information, see Indicator field trigger scripts.
Add to all indicator types	This option is selected by default, which means this field is available to use in all incident types.  Clear the checkbox to associate this field with a subset of indicator types.
Make data available for search	The values for this field can be returned in searches.

6. Save the field.

If you subsequently edit the field, you can optionally select Don't show in the indicators layout. If you select this, the indicator field does not appear in the layout but the data is displayed in the context data.

7. (Optional) Add a custom field to a section in the indicator layout.

If you select Don't show in the indicators layout, the field will not appear in the layout.

8. (Optional) In the indicator type, map custom indicator fields, so an indicator field is automatically updated, without the analyst having to manually change it.

#### 16.2.1.2.1 | Indicator fields structure

##### Abstract

Indicator fields structure aligned with STIX standards to more easily share and work with IOCs.

Cortex XSOAR IOC fields are based on the STIX 2.1 specifications. These fields provide a guideline for the fields we recommend you maintain within an IOC. None of the fields are mandatory, except the value field. Maintaining this field structure enables you to share and export IOCs to additional threat intel based systems as well as to other cybersecurity devices.

Like STIX, Cortex XSOAR indicators are divided into two categories, STIX Domain Objects (SDOs) and STIX Cyber-observable Objects (SCOs). The category determines which fields are presented in the layout of that specific IOC. In Cortex XSOAR, all SCOS can be used in a relationship with either SDOs or SCOS.

Each IOC table of fields is separated into three parts:

- System fields - Fields created and managed by Cortex XSOAR.
- Custom core fields - Custom fields shared by all IOCs of the same type (SDO or SCO). Fields may be empty.
- Custom unique fields - Fields unique to a specific type of IOC. If a user associates more fields with the IOC, the additional fields are also treated as unique.

## STIX Cyber-observable Objects (SCO)

## Account

Similar to STIX User Account Object, this indicator type represents a user account in various platforms such as operating system, social media accounts, and Active Directory. The value for the object is usually the username for logging in.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Blocked	A Boolean switch to mark the object as blocked in the user environment.	
Community Notes	Comments and free-form notes regarding the indicator.	
Description	The description of the object.	
STIX ID	The STIX ID for the object in the format of account--<UUID>.	
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
Account Type	Specifies the type of the account, comes from account-type-ov by STIX.	
Creation Date	The date the account was created (not the date the indicator was created).	
Display Name	The display name of the account as it is shown in the UI.	
Groups	The groups the account is a member of.	
User ID	The account's unique ID according to the system it was taken from.	

## Domain / DomainGlob

Network domain name, similar to the STIX Domain Name object. The value is the domain address.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Blocked	A Boolean switch to mark the object as blocked in the user environment.	
Community Notes	Comments and free form notes regarding the indicator.	
Description	The description of the object.	
STIX ID	The STIX ID for the object in the format of domain--<UUID>.	
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
Creation Date	The date the domain was created.	
DNS Records	All types of DNS records with a timestamp and their values (GRID).	
Expiration Date	The domain expiration date.	
Certificates	Any certificates issued for the domain.	
WHOIS Records	Any records from WHOIS about the domain (GRID).	

## Email

A single user email address.

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

Custom Fields - Core		Description
Blocked		A Boolean switch to mark the object as blocked in the user environment.
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of email--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique
None

#### File

Represents a single file. For backward compatibility, the indicator has multiple fields for different types of hashes. New hashes, however, should be stored under the Hashes grid field. The file value should be its hash (either MD5, SHA-1, SHA-256, or SHA-512, in that order).

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.

<b>System Fields</b>		<b>Description</b>
Modified		When the object was last modified.

<b>Custom Fields - Core</b>		<b>Description</b>
Blocked		A Boolean switch to mark the object as blocked in the user environment.
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of file--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

<b>Custom Fields - Unique</b>		<b>Description</b>
Creation Date		The file creation date.
File Extension		The file extension.
Associated File Names		Names the file is associated with.
File Type		The type of the file.
Hashes		Any hashes not specified in a separate field.
imphash		The imphash.
MD5		The MD5 hash.
Modified Date		When the file was modified on the origin.
Path		The path to the file.
Quarantined		Was the file quarantined?
SHA1		The SHA1 hash.

Custom Fields - Unique		Description
SHA256		The SHA256 hash.
SHA512		The SHA512 hash.
Size		The file size.
SSDeep		The SSDeep hash.

IPv4 / IPv6 / CIDR / IPv6CIDR

Represents an IP address and its subnet (CIDR). If no subnet is provided, the address is treated as a single IP (same as a /32 subnet).

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

Custom Fields - Core		Description
Blocked		A Boolean switch to mark the object as blocked in the user environment.
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of type--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique		Description
Geo Country		The country where the object is located.

Custom Fields - Unique		Description
Geo Location	A set of geographic coordinates for the object.	
WHOIS records	Any records from WHOIS about the domain (GRID).	

## URL

Represents the properties of a uniform resource locator.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Blocked	A Boolean switch to mark the object as blocked in the user environment.	
Community Notes	Comments and free form notes regarding the indicator.	
Description	The description of the object.	
STIX ID	The STIX ID for the object in the format of url--<UUID>.	
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
Certificates	Any certificates issued for the domain.	

## STIX Domain Objects (SDO)

## Attack Pattern

Attack patterns are a type of TTP (Tactics, Techniques and Procedures) that describe ways adversaries attempt to compromise targets. Attack patterns help categorize attacks, generalize specific attacks to the patterns that they follow, and provide detailed information about how attacks are performed. An example

of an attack pattern is spear phishing, where an attacker sends a carefully crafted email message with the intent of getting the target to click a link or open an attachment that delivers malware. Attack patterns can also be more specific, such as spear phishing by a particular threat actor (for example, an email saying the target won a contest).

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

Custom Fields - Core		Description
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of attack-pattern--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique		Description
Kill Chain Phases		The list of kill chain phases this Attack Pattern is used for.
External References		List of external references consisting of a source and ID. For example, {source: mitere, id: T1189}

## Campaign

A campaign is a grouping of adversarial behaviors that describes a set of malicious activities or attacks (sometimes called waves) that occur over a period of time against a specific set of targets. Campaigns usually have well defined objectives and may be part of an intrusion set.

Campaigns are often attributed to an intrusion set and threat actors. The threat actors may reuse known infrastructure from the intrusion set or may set up new infrastructure specifically for conducting that campaign.

Campaigns can be characterized by their objectives and the incidents they cause, people or resources they target, and the resources (such as infrastructure, intelligence, and malware, tools) they use.

For example, a campaign can describe a crime syndicate's attack using a specific variant of malware and new C2 servers against the executives of ACME Bank during the summer of 2020 to gain secret information about an upcoming merger with another bank.

<b>System Fields</b>		<b>Description</b>
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

<b>Custom Fields - Core</b>		<b>Description</b>
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of campaign--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

<b>Custom Fields - Unique</b>		<b>Description</b>
Aliases		Alternative names used to identify this campaign.
Objective		The campaign's primary goal, objective, desired outcome, or intended effect.

#### Course of action

A course of action is an action taken either to prevent an attack or to respond to an attack that is in progress. It may describe technical, automatable responses (applying patches, reconfiguring firewalls), but can also describe higher level actions such as employee training or policy changes. For example, a course of action to mitigate a vulnerability could describe applying the patch that fixes it.

<b>System Fields</b>		<b>Description</b>
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.

<b>System Fields</b>		<b>Description</b>
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

<b>Custom Fields - Core</b>		<b>Description</b>
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of course-of-action--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

<b>Custom Fields - Unique</b>		<b>Description</b>
Action		Reserved to capture structured/automated courses of action.

## CVE

To preserve backward compatibility, our vulnerability indicator is referred to as CVE, but it is equivalent to the Vulnerability object defined by STIX. Unlike STIX, in TIM the object is identified by its CVE number. A vulnerability is a weakness or defect in the requirements, designs, or implementations of the computational logic (code) found in software and some hardware components (firmware) that can be directly exploited to negatively impact the confidentiality, integrity, or availability of that system.

<b>System Fields</b>		<b>Description</b>
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

<b>Custom Fields - Core</b>		<b>Description</b>
Community Notes		Comments and free form notes regarding the indicator.

Custom Fields - Core		Description
Description	The description of the object.	
STIX ID	The STIX ID for the object in the format of vulnerability--<UUID>.	
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
CVSS Version	The version of the CVSS scoring system.	
CVSS Score	The score given to the CVE.	
CVSS Vector	The full CVSS vector.	
CVSS Table	All CVSS data by Metric - Value pairs.	

#### Infrastructure

The Infrastructure SDO represents a type of TTP and describes any systems, software services and any associated physical or virtual resources that support some purpose (for example, C2 servers used as part of an attack, a device or server that is part of a defense, and database servers targeted by an attack). While elements of an attack can be represented by other SDOs or SCOs, the Infrastructure SDO represents a named group of related data that constitutes the infrastructure.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Community Notes	Comments and free form notes regarding the indicator.	
Description	The description of the object.	

Custom Fields - Core		Description
STIX ID		The STIX ID for the object in the format of <code>infrastructure--&lt;UUID&gt;</code> .
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique		Description
Aliases		Alternative names used to identify this infrastructure.
Infrastructure types		The type of infrastructure being described. Values should come from STIX <code>infrastructure-type-ov</code> open vocabulary.

#### Intrusion set

An intrusion set is a grouped set of adversarial behaviors and resources with common properties that is believed to be orchestrated by a single organization. An intrusion set may capture multiple campaigns or other activities that are all tied together by shared attributes indicating a commonly known or unknown threat actor. New activity can be attributed to an intrusion set even if the threat actors behind the attack are not known. Threat actors can move from supporting one intrusion set to supporting another, or they may support multiple intrusion sets.

Whereas a campaign is a set of attacks over a period of time against a specific set of targets to achieve an objective, an intrusion set is the entire attack package and may be used over a very long period of time in multiple campaigns to achieve potentially multiple purposes.

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

Custom Fields - Core		Description
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of <code>intrusion-set--&lt;UUID&gt;</code> .
Tags		Tags attached to the object.

<b>Custom Fields - Core</b>		<b>Description</b>
Traffic Light Protocol		Red, Amber, Green, or White.

<b>Custom Fields - Unique</b>		<b>Description</b>
Aliases		Alternative names used to identify this intrusion set.
Goals		The high-level goals of this intrusion set, what it is trying to do.
Primary Motivation		The primary reason, motivation, or purpose behind this intrusion set. Values should come from STIX attack-motivation-ov open vocabulary.
Secondary Motivation		The secondary reason, motivation, or purpose behind this intrusion set. Values should come from STIX attack-motivation-ov open vocabulary.
Resource level		Specifies the organizational level at which this intrusion set typically works. Values should come from STIX attack-resource-level-ov open vocabulary.

## Malware

Malware is a type of TTP that represents malicious code. It generally refers to a program that is inserted into a system, usually covertly. The intent is to compromise the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or otherwise annoy or disrupt the victim.

<b>System Fields</b>		<b>Description</b>
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

<b>Custom Fields - Core</b>		<b>Description</b>
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of malware--<UUID>.

Custom Fields - Core		Description
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
Aliases	A list of other names the malware is known as.	
Architecture	The processor architectures (for example, x86, ARM) that the malware instance or family is executable on. The values should come from the STIX processor-architecture-ov open vocabulary.	
Capabilities	Any of the capabilities identified for the malware instance or family. The values should come from STIX malware-capabilities-ov open vocabulary.	
Implementation Languages	The programming language(s) used to implement the malware instance or family. The values should come from the STIX implementation-language-ov open vocabulary.	
Is Malware Family	Whether the object represents a malware family (if true) or a malware instance (if false).	
Malware Types	Which type of malware. Values should come from STIX malware-type-ov open vocabulary.	
Operating System Refs	Identifier of a software object.	

#### Report

Reports are collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details. They are used to group related threat intelligence together so that it can be published as a comprehensive cyber threat story.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of report--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique		Description
Publications		Links to publications of the report.

#### Threat actor

Threat actors are individuals, groups, or organizations believed to be operating with malicious intent. A threat actor is not an intrusion set but may support or be affiliated with various intrusion sets, groups, or organizations over time.

System Fields		Description
Value		Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.
Verdict		Malicious, Suspicious, Benign, or Unknown.
Expiration		The expiration date of the object.
Source Time Stamp		When the object was created in the system.
Modified		When the object was last modified.

Custom Fields - Core		Description
Community Notes		Comments and free form notes regarding the indicator.
Description		The description of the object.
STIX ID		The STIX ID for the object in the format of threat-actor--<UUID>.
Tags		Tags attached to the object.
Traffic Light Protocol		Red, Amber, Green, or White.

Custom Fields - Unique		Description
Alias	A list of other names the threat actor is known as.	
Geo country	The country the threat actor is associated with.	
Goals	The high-level goals of this threat actor, what it is trying to do.	
Resource Level	The organizational level at which this threat actor typically works. Values for this property should come from STIX attack-resource-level-ov open vocabulary.	
Primary Motivation	The primary reason, motivation, or purpose behind this threat actor. Values for this property should come from STIX attack-motivation-ov open vocabulary.	
Secondary Motivation	The secondary reasons, motivations, or purposes behind this threat actor. Values for this property should come from STIX attack-motivation-ov open vocabulary.	
Sophistication	The skill, specific knowledge, special training, or expertise a threat actor must have to perform the attack. Values for this property should come from STIX threat-actor-sophistication-ov open vocabulary.	
Threat actor type	The type(s) of this threat actor. Values should come from STIX threat-actor-type-ov open vocabulary.	

**Tool**

Tools are legitimate software used by threat actors to perform attacks. Knowing how and when threat actors use such tools can help understand how campaigns are executed. Unlike malware, these tools or software packages are often found on a system and have legitimate purposes for power users, system administrators, network administrators, or even regular users. Remote access tools such as RDP and network scanning tools such as Nmap are examples of tools that may be used by a threat actor during an attack.

System Fields		Description
Value	Defines the indicator on Cortex XSOAR. The value is the main key for the object in the system.	
Verdict	Malicious, Suspicious, Benign, or Unknown.	
Expiration	The expiration date of the object.	
Source Time Stamp	When the object was created in the system.	
Modified	When the object was last modified.	

Custom Fields - Core		Description
Community Notes	Comments and free form notes regarding the indicator.	
Description	The description of the object.	

Custom Fields - Core		Description
STIX ID	The STIX ID for the object in the format of tool--<UUID>.	
Tags	Tags attached to the object.	
Traffic Light Protocol	Red, Amber, Green, or White.	

Custom Fields - Unique		Description
Alias	Alternative names used to identify this tool.	
Tool Types	The kind(s) of tool(s) being described. Values for this property should come from STIX tool-type-ov open vocabulary.	
Tool Version	The version identifier associated with the tool.	
Kill Chain Phases	The list of kill chain phases this attack pattern is used for.	

#### 16.2.1.2.2 | Indicator field trigger scripts

##### Abstract

Associate Cortex XSOAR indicator fields with scripts that are triggered when the field changes.

Indicator field trigger scripts are automated responses that are triggered by a change in an indicator field value. In the script, you define the change in the indicator field value to check for and the actions to take when the change occurs. For example, you can:

- Create a script that runs when the Verdict field of an indicator changes. For example, the script will fetch all incidents related to the indicator and take any action that is configured, such as reopening or changing severity.
- Create a script that runs when the Expiration Status field changes. For example, you can define a script that will immediately update the relevant allow/block list and not wait for the next iteration, as seen in the following sample script:

```
indicators = demisto.args().get('indicators')
new_value = demisto.args().get('new')

indicator_values = []
for indicator in indicators:
    current_value = indicator.get('value')
    indicator_values.append(current_value)

if new_value == "Expired":
    # update allow/block list regarding expired indicators
else:
    # update allow/block list regarding active indicators
```

##### NOTE:

You must have a TIM license to run field change-triggered scripts on indicator fields.

##### Indicator field trigger script arguments

Scripts can be created in Python, PowerShell, or JavaScript on the Scripts page. To use a field trigger script, you need to add the field-change-triggered-indicator tag when creating the script. You can then add the script in the Attributes tab when you edit or Create a Custom Indicator Field. If you did not add the tag when creating the script, the script will not be available for use.

Indicator field trigger scripts have the following triggered field information available as arguments (args):

Argument	Description
<code>associatedToAll</code>	Whether the field is associated with all or some indicators. Value: <code>true</code> or <code>false</code> .
<code>associatedTypes</code>	An array of the indicator types the field is associated with.
<code>cliName</code>	The name of the field when called from the CLI.
<code>description</code>	The description of the field.
<code>indicators</code>	A list of indicators that have the current change.
<code>isReadOnly</code>	Specifies whether the field is non-editable. Value: <code>true</code> or <code>false</code> .
<code>name</code>	The name of the field.
<code>new</code>	The new value of the field.
<code>old</code>	The old value of the field.
<code>ownerOnly</code>	Specifies that only the creator of the field can edit. Value: <code>true</code> or <code>false</code> .
<code>placeholder</code>	The placeholder text.
<code>required</code>	Specifies whether this is a mandatory field. Value: <code>true</code> or <code>false</code> .
<code>selectValues</code>	If this is a multi-select type field, these are the values the field can take.
<code>system</code>	Whether it is a Cortex XSOAR defined field.
<code>type</code>	The field type.
<code>user</code>	The username of the user who triggered the script.

Indicator field trigger script best practices

- Indicator field trigger scripts can be configured on the Verdict, Related Incidents, Expiration Status, and Indicator Type fields, as well as any custom indicator fields.
- Indicator field trigger scripts work in all TIM (Threat Intelligence Management) scenarios and workflows, except for feed ingestion.
- Fields that can hold a list (related incidents, multi-select/tag/role type custom fields) will provide an array of the delta. For example, if a multi-select field value has changed from ["a"] to ["a", "b"], the new argument of the script will get a value of ["b"].
- Indicator field trigger scripts run as a batch. This means that if multiple indicators are changed in the same way and are set to trigger the same action, it will happen in one batch.

For example, in the following scenario for a configured indicator field trigger script named `myTriggerScript` on the Verdict indicator field:

- The Threat Intel Library has two existing Malicious indicators: 1.1.1.1 and 2.2.2.2.
- The user runs the following command `!setIndicators indicatorsValues="1.1.1.1,2.2.2.2" verdict=Benign`.
- The `myTriggerScript` script will run just once, with the following parameters:
  - new - "Benign"
  - old - "Malicious"
  - indicators - "[{<indicator\_1.1.1.1>},{<indicator\_2.2.2.2>}]"
- When writing indicator field trigger scripts, avoid scenarios that call the scripts endlessly (for example, a change in field A triggers script X, which changes field B's value, which in turn calls script Y, which changes field A's value).

#### Add an indicator field trigger script to an indicator field

After creating an indicator field trigger script in the Scripts page in Python, PowerShell, or JavaScript, you can then associate it with an indicator field.

- Go to Settings & Info → Settings → Indicators → Fields.
- Select the indicator field and click Edit.
- In the Attributes tab, under Script to run when field value changes, select the desired indicator field trigger script.

#### **NOTE:**

Indicator field trigger scripts must have the `field-change-triggered-indicator` tag to appear in the list.

### 16.2.1.3 | Indicator layout customization

#### Abstract

Customize an indicator layout for an indicator type in Cortex XSOAR. View the layout in the indicator Summary and Quick View.

Each indicator type has a unique set of data relevant to that specific indicator type, including layouts. It is important to display the most relevant data for users. Each out-of-the-box indicator comes with a layout. You can customize almost every aspect of the layout, including which tabs appear, in which order they appear, who has permission to view the tabs, what information appears, and how it is displayed.

You can see which indicator type uses the indicator layout in the Types tab under Settings & Info → Settings → Object Setup → Indicators. The indicator layout name appears in the Layout column.

You can customize the display information including fields for existing indicators, by modifying the sections and fields for the following views:

Section	Description
Indicator Summary	<p>You can customize almost every aspect of the layout, including which tabs appear, the order they appear, and who has permission. In each field or tab, you can add filters by clicking the eye icon, which enables you to add conditions that show specific fields or tabs relevant to the indicator.</p> <p>You can add a script in the indicator layout, such as a mapping script, which determines where an IP address originates and displays it on a map.</p> <p><b>NOTE:</b></p> <p>Only available if you have a TIM license.</p>

Section	Description
Quick View	Add, edit, and delete sections, fields, and filters in the Quick View section from an incident.
"New"/"Edit" form	Add, edit, and delete fields and buttons to be displayed when creating or editing an indicator.

**NOTE:**

By default, when editing a list or text values in an incident/indicator layout, the changes are not saved until you confirm your changes (clicking the checkmark icon in the value field). These icons are designed to give you additional security when updating fields in incidents and indicators.

You can change this default behavior by adding a server configuration. For more information, see Configure inline value fields.

## Create an indicator layout

1. Select Settings & Info → Settings → Object Setup → Indicators → Layouts → New Layout.

You can see that you can customize the Indicator Summary section, Quick View, and the New/Edit form.

2. Add a meaningful name for the layout.

3. Customize the tabs by clicking the settings wheel icon and then doing the following:

**NOTE:**

You can click and drag a tab to reorder the tabs.

Action	Description
Rename	You can also edit a tab's name by clicking the tab.
Duplicate	Copies the existing tab.
Delete	Deletes the tab.
Show empty fields	The setting that you configure in the layout becomes the default value seen in the report for the specific tab, which can then be overridden.  You can also set a global default value using the <code>UI.summary.page.hide.empty.fields</code> server configuration, which can also be overridden for a specific tab.
Hide tab	Hides the tab. Rather than deleting the tab, you may want to use the tab again for future use.
Format for exporting	Build your layout based on A4 proportions to match the format used for exporting. Selecting this option hides the tab by default, but the tab will remain available for export.
Viewing Permissions	Select which roles can view the tabs.
Display Filter	Add or view a filter applied to the tab. If the filters apply, the specific fields or tabs are shown in the layout. If the mandatory field is not shown in the layout, the user is not obliged to complete it.

4. Do the following:

- Drag and drop the required sections, fields, buttons, and tabs.
- Customize sections and create buttons.
- Add any required filters.
- Create new tabs

5. Repeat step 3 for the Quick View tab.

6. In the New/Edit Form, drag and drop the required fields and buttons.

You can also edit the Basic Information and the Custom Field sections.

7. Save the indicator layout and add it to the indicator type.

#### Edit an indicator layout

1. Go to Settings & Info → Settings → Object Setup → Indicators → Layouts.

2. Click the name of the indicator type layout you want to edit.

You are presented with the current layout, which is populated with demo data so you can see how the fields fit.

3. If using a Content Pack Indicator Type Layout, detach or duplicate the layout.

#### NOTE:

If you duplicate the layout, you need to update the indicator type to add the new layout.

While an indicator layout is detached, it does not receive content pack updates. If you detach an indicator type layout, edit, and later want to receive content pack updates for that layout, we recommend you duplicate the indicator layout before reattaching the original to protect your changes from content pack updates. When detached, you can also edit the layout from the Indicator Type tab.

a. Select the checkbox for the indicator layout you want to detach.

b. Right-click and select Detach.

#### Customize sections

1. Create or edit a layout.

2. From the Sections tab in the Library, drag and drop the following sections:

Section	Description
New Section	After creating a new section, click the Fields and Buttons tab and drag and drop the fields as required.
Cortex XSOAR out-of-the-box sections	Out-of-the-box sections such as Expiration Status and Verdict.
General Purpose Dynamic Section	You can add a script in the indicator layout. For example, to assign a script that determines and displays the Geolocation of an IP address on a map. For more information, see Set up Google Maps.

#### NOTE:

To remove or duplicate a section, select the section, click  and then select Duplicate, or Remove.

3. Define the section properties, by clicking  and then Edit section settings.

#### TIP:

Limit the number of incident fields to 50 in each section. You can create additional sections as needed.

You can determine how a section in the layout appears in the layout. For example, you may want a section header, or configure the fields to appear in rows or as cards. If some of the field values will be very long, use rows instead of cards. If the field values are short, you might want to use cards so you can fit more fields into a section.

4. Drag and drop fields, and add any filters as required.

5. If relevant, create a new tab and repeat the steps as required.

#### Add a script to the layout

You can add content to the Indicator Summary tab, based on a script, by adding the script in the General Purpose Dynamic Section. The script can return simple text, markdown, or an HTML, the results of which appear in the General Purpose Dynamic Section.

You can add any required information from a script. For example:

- Add a mapping script that determines where an IP address originates and displays it on a map.
- Add a custom widget to the indicator page. The procedure is similar for indicators and incidents.
- Add the FeedRelatedIndicator script from the Scripts page, which contains information about the relationship between an indicator, entity (such as malware), and other indicators (such as a MITRE ATT&K indicator), and connects externally to those indicators, if relevant.

Before you begin, you need to create a script.

#### NOTE:

Ensure that you have added the **dynamic-indicator-section** tag, otherwise, you can't select it when adding a script

1. Go to Settings & Info → Settings → Objects Setup → Indicators → Layouts.
2. Click on the indicator layout you want to edit.  
The layout must either be custom content (a layout you created), a layout duplicated from a content pack layout, or a detached layout from a content pack. You cannot edit a layout that is attached. To detach an attached layout, select the indicator layout and click Detach. The layout must either be custom content (a layout you created) or a detached layout from a content pack. You cannot edit a layout that is attached.
3. Drag and drop the General Purpose Dynamic Section onto the page.
4. Select the General Purpose Dynamic Section, click  , and then click Edit section settings.
5. In the Name and Description fields, add a meaningful name and a description for the dynamic section that explains what the script displays.
6. In the Automation script field, select the script that returns data for the dynamic section.
7. Click OK.

#### Create custom buttons

You can add existing buttons or create buttons and then drag and drop them in the layout.

To add a custom button, create a script and then add the new button to the indicator layout and choose the script, as described in the example below. These buttons can simplify and assist an analyst in carrying out various tasks. For example, you can create a button to run an enrichment script on an identified indicator.

For fields (script arguments) that are optional, you can define whether to show them to analysts when they click on buttons. To expose an optional field, select the Ask User checkbox next to the script arguments in the button settings page.

#### NOTE:

When creating a script for use in an indicator layout, the **indicator-action-button** tag must be assigned for the script to be available for custom buttons.

In the following example, create a button that adds the indicator to a Hunt incident type so the Threat Intel team can review it.

1. Save the following script on your computer. On the Scripts page, click the upload script icon and upload the file.

```
commonfields:  
  id: d3716514-4c2b-453c-8072-4fd4807bca0a  
  version: 30  
  vcShouldKeepItemLegacyProdMachine: false  
  name: newIncidentFromIndicator  
  script: |+  
    from pprint import pformat  
  
  args = demisto.args()  
  
  fields = {}  
  fields['type'] = args['type']  
  fields['details'] = args['indicator']['value']  
  fields['name'] = args['type'] + " for " + args['indicator']['value']  
  
  res = demisto.executeCommand('createNewIncident', fields)  
  
  newID = res[0]['EntryContext']['CreatedIncidentID']  
  
  demisto.executeCommand("associateIndicatorsToIncident", {"indicatorsValues": args['indicator']['value'], "incidentId":int(newID)})  
  
type: python  
tags:
```

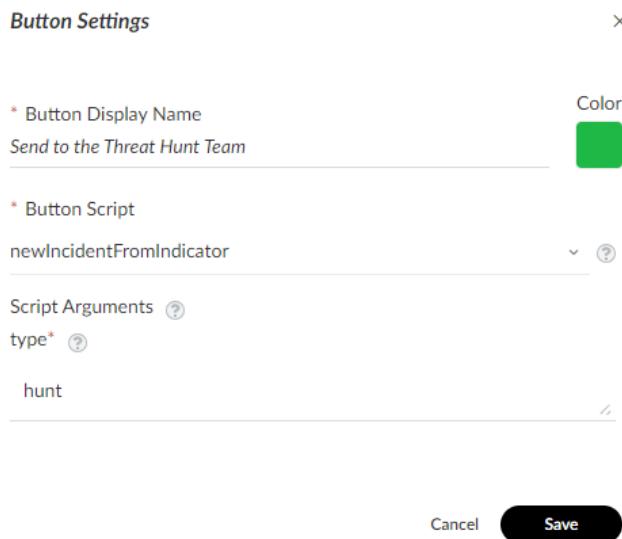
```

- indicator-action-button
enabled: true
args:
- name: type
  required: true
  description: Incident Type
scripttarget: 0
subtype: python3
pswd: ""
runonce: false
dockerrimage: demisto/python3:3.8.5.11789
runas: DBotWeakRole

```

When uploading to Cortex XSOAR the newIncidentFromIndicator name and the indicator-action-button is already populated.

2. Go to Settings & Info → Settings → Object Setup → Indicators → Layouts and click the relevant indicator type layout.
3. From the Fields and Buttons tab, drag the +New Button and drop into the relevant section.
4. Click to configure.
5. Enter a descriptive name for the button. For this example, we call it **Send to the Threat Hunt Team**.
6. Select a color.
7. Select the script we added above: newIncidentFromIndicator.
8. In the Script Arguments field, under the type field, add Hunt.



9. Save the button.

When you view an indicator and click this button, an incident is created with the Hunt incident type.

To test the button, add the layout to the indicator type, go to the Threat Intel (Indicators) page, create a new indicator, and assign it to the relevant indicator type. View the indicator, click the Pass to Threat Hunt Team button, and verify that a new incident has been created.

Add the layout to the indicator type

1. Go to Settings & Info → Settings → Object Setup → Indicators → Types.
2. Select the indicator type and click Edit.
3. In the Layout field, from the dropdown list, add the customized layout.
4. (Optional) For a customized layout, you can contribute it to the Marketplace.
  - a. In the Layouts page, select the new layout and then click Contribute to Marketplace.
  - b. In the dialog box select either Save and submit your contribution or Save and download your contribution for later use, which you can view in the Contributions tab in Marketplace.

If you select Save and submit your contribution, your layout is validated and you are prompted to submit to review. You can also view your contribution in Marketplace.

## 16.2.2 | Indicator classification and mapping

### Abstract

Learn about the classification and mapping for indicators.

The following table shows methods by which indicators are detected and ingested in Cortex XSOAR and how they are classified and mapped.

Method	Description	Classification And Mapping
Integration	Feed integrations: Fetch indicators from a feed, for example, TAXII, Office 365, and Unit 42 ATOMS Feed.	<p>Indicator classification and mapping is done in the integration code by duplicating the integration in Integrations → Instances and not in the Indicators → Classification &amp; Mapping tab. For more information, see Feed Integrations.</p> <p>Some integrations come with a classifier and mapper, which you can customize.</p>
Indicator extraction	Indicators are extracted from selected incidents that flow into Cortex XSOAR, for example from an SIEM integration.	<p>Only the value of an indicator is extracted, so no classification or mapping is needed.</p> <p>For more information, see Indicator extraction.</p>
Manual	<ul style="list-style-type: none"> <li>• Command line</li> <li>• Mark: The user marks a piece of data as an indicator.</li> <li>• STIX file: Manually upload a STIX file on the Threat Intel (Indicators) page.</li> </ul>	<p>Data is inserted manually via the UI so no classification or mapping is needed.</p> <p>If importing an STIX file, mapping is done via the STIX parser code.</p>

### Classify and map an indicator type for an integration

The indicator classification and mapping feature enables you to take the data that Cortex XSOAR ingests from integrations, and classify and map the data to indicator types and indicator fields. By classifying the data as different indicator types, you can process them with different playbooks suited to their respective requirements.

#### NOTE:

When creating a new indicator type, you classify and map the indicator fields in the indicator type settings. For more details, see Map custom indicator fields.

Classification determines the type of indicator that is created for data ingested from a specific integration. You create a classifier and define that classifier in an integration.

You can map the fields from your third-party integration to the fields in your indicator layouts as follows:

- Map your fields to indicator types irrespective of the integration or classifier. This means that you can create a mapping before defining an instance and ingest indicators. By doing so, when you do define an instance and apply a mapper, the data that comes in is already mapped.
- Create default mapping for all of the fields that are common to all indicator types, and then map only those fields that are specific to each alert type individually. You can still overwrite the contents of a field in the specific indicator type.

### Classify an indicator type

When an integration fetches indicators, it populates the raw JSON object for the indicator. The raw JSON object contains all of the attributes (fields) for an indicator. For example, source, when the event was created, the priority that was designated by the integration, and more. When classifying ingested indicator data, you want to select an attribute (field) that can determine the indicator type.

Use this procedure to create a classifier or duplicate an existing classifier for ingested indicator data.

1. Select Settings & Info → Settings → Object Setup → Indicators → Classification & Mapping.

2. Do one of the following:

a. To create a new classifier, select + New → Indicator Classifier.

b. To edit an existing classifier, select it and click Edit.

If the classifier is installed from a content pack, you need to duplicate and then edit.

3. Under Get data, select from where you want to import the indicator data. You will classify the indicator type based on this information.

**NOTE:**

You can optionally skip importing data. Click the pencil on the right of each indicator type on the right pane to enter the value manually.

- Pull from instance: Select an existing integration instance to import indicator data from.
- Upload JSON: Upload a formatted JSON file that includes the fields you want to classify by.

4. Under Fetched data, select from the attributes (fields) in the imported indicator object a field that will serve as the classifier (key) to route to a specific indicator type.

Cortex XSOAR searches through the imported indicator objects for the values for the field you select.

5. Drag the found values from the Unmapped Values column to the relevant indicator type on the right pane.

6. Save the classifier.

7. Apply the indicator classifier to the relevant feed integration.

- Go to Settings & Info → Settings → Integrations → Instances.
- Select an existing integration instance you want to apply the indicator classifier to or create a new integration instance.
- In the integration instance settings under Classifier, select the classifier you created and click Save.

**Map indicator fields**

Mappers enable you to map the information from ingested indicator data to the indicator fields that you have in your system.

Mapping data takes place in two stages:

- Map all of the fields that are common to all indicators in the default mapping.
- Map the additional fields that are specific for each indicator type, or overwrite the mapping that you used in the default mapping.

**NOTE:**

In the Classification & Mapping page, the mapping does not indicate for which indicator types they are configured. When creating a mapper, it is best practice to add to the mapper name and the indicator type the mapper is for. For example, Mail Listener - Phishing.

When mapping a list, we recommend you map to a multi-select field. Short text fields do not support lists. If you need to map a list to a short text field, add a transformer in the relevant playbook task to split the data back into a list.

Use this procedure to create a mapper or duplicate an existing mapper to map all of the ingested indicator fields to an indicator layout.

1. Select Settings & Info → Settings → Object Setup → Indicators → Classification & Mapping.

2. Do one of the following:

- To create a new mapper, select + New → Indicator Mapper (incoming).
- To edit an existing mapper, select it and click Edit.

If the mapper is installed from a content pack, you need to duplicate and then edit.

3. Under Get data, select from where you want to import the indicator data. You will map the indicator data based on this information.

- Pull from instance: Select an existing integration instance to import indicator data from.
- Upload JSON: Upload a formatted JSON file that includes the fields you want to map.

4. Under Indicator Type, start by mapping out the Common Mapping. This mapping includes the fields that are common to all of the indicator types and saves you time having to define these fields individually in each indicator type.

5. Click the attribute (field) to which you want to map. You can further manipulate the field using filters and transformers.

6. Repeat this process for the other indicator types for which this mapping is relevant.

7. Save the mapper.

8. Apply the indicator mapper to the relevant feed integration.

- Go to Settings & Info → Settings → Integrations → Instances.
- Select an existing integration instance you want to apply the classifier to or create a new integration instance.
- In the integration instance settings under Mapper, select the mapper you created and click Save.

### 16.2.3 | Indicator extraction

#### Abstract

Extract indicators from Cortex XSOAR incident fields and enrich them with commands and scripts defined for the indicator type.

Indicator extraction identifies indicators from different text sources in the system (such as War Room entries, email content, etc.), extracts them (usually based on regex) and creates indicators in Cortex XSOAR. After extraction, the indicator can be enriched.

After indicators are extracted, they are enriched using commands and scripts defined for the indicator type. Indicator enrichment provides detailed information about the indicator, based on enrichment feeds such as VirusTotal and IPinfo.

#### NOTE:

Reputation commands, such as `!ip` and `!domain`, can only be used after you configure and enable a reputation integration instance, such as Virus Total and Whois.

Some content packs include a dashboard and widget that track API rate limit errors. You can use this information for troubleshooting and to make decisions about indicator enrichment.

#### Indicator Extraction Methods

You can customize indicator extraction using the following methods:

- Incident types

You can extract indicators from incident fields when an incident is created and when an incident field changes. Indicator extraction rules for content pack incident types are determined by the content pack. For example, in a Phishing incident type, by default, in the Destination IP field, IPv6 and IP indicators are extracted. For the Detection URL field, the URL indicator field is extracted.

If enabled, indicator extraction is automatic. For example, in a Phishing incident, indicator extraction is set to extract the IP indicator (in the incident type). When the incident field updates, the IP indicator field is extracted automatically. In the War Room, you can check that the IP indicator field has been extracted by typing `1.1.1.1`. Cortex XSOAR recognizes the indicator as an IP indicator by matching it to the IP indicator's regex. It then extracts and enriches the indicator using an integration that includes the IP command (such as IPinfo).

#### NOTE:

To change the indicator extraction rules for an incident type installed with a content pack, including an incident type propagated to a tenant in a multi-tenant environment, you need to detach the incident type. Once detached, the incident type does not receive new content from Cortex XSOAR. If you want to receive content updates reattach the incident type. If you want to instead receive content updates and save the content, duplicate the incident type and edit the duplicate type. For more information, see Incident layout customization.

#### CAUTION:

Extracting indicators can adversely affect system performance. We recommend that you define extraction settings for each incident type, as needed.

For example, for Malware you may want to extract all IP addresses, for Phishing you may only want to extract IP addresses from specific email headers. For attachments, you may want to disable indicator extraction to reduce external API usage and protect restricted data (the hash) from being sent.

- Playbook tasks. For more information, see Set the indicator extraction mode for a playbook task.
- Commands: Run a command using the command line in Cortex XSOAR during an investigation. For more information, see Extract and enrich an indicator.

#### Indicator Extraction Mode Options

Indicator Extraction supports the following modes:

- None
- Inline
- Out of band
- Use system default

For detailed information about the modes and how to set them up, see Indicator extraction modes.

#### Indicator Scripts

When creating or editing an indicator type, you can add the following scripts:

- Formatting scripts
- Enhancement scripts
- Reputation scripts

During the indicator extraction and extraction flow, the order of execution is regex, formatting script, and reputation command, reputation script. Enhancement scripts are not part of the flow.

Indicators are identified using regex, and then the formatting script transforms the regex into a usable indicator for use in Cortex XSOAR in the War Room, reports, dashboards, etc. Reputation commands and scripts enable you to change the reputation of the indicator.

Enhancement scripts enable you to gather additional data about the highlighted entry in the War Room.

#### Indicator Extraction and Enrichment in the CLI

You can run commands in the CLI, such as `!extractIndicators`, `!enrichindicators`, `!ip`, `!domain`, and reputation script commands such as `!1URLReputation`, `!IPReputation`. For more information, see Extract and enrich an indicator.

##### 16.2.3.1 | Indicator extraction modes

###### Abstract

Configure the indicator extraction mode. Options are none (no extraction), inline, out-of-band, or use system default.

Indicator extraction supports the following modes:

- **None:** Indicators are not extracted automatically. Use this option when you do not want to extract and enrich the indicators.
- **Inline:** Indicators are extracted within the context that the indicator extraction runs (synchronously). The findings are added to the context data. For example, if you define indicator extraction for the phishing incident type as inline:
  - For incident creation, by default, the playbook you defined to run does not run until the indicators have been extracted.
  - For an on field change, extraction occurs before the next playbook tasks run. Use this option when you need to have the most robust information available per indicator.

###### NOTE:

This configuration may delay playbook execution. While indicator creation using the command `createIndicator` is asynchronous, automatic indicator extraction and enrichment is run synchronously. Data is placed into the incident context and is available via the context for subsequent tasks.

- **Out of band:** Indicators are extracted in parallel (asynchronously) to other actions. The extracted data is available within the incident, but it is not available for immediate use in task inputs, or outputs, since the information is not available in real time.

For incident creation, out of band is used in rare cases where you do not need the indicators extracted for the playbook flow. You still want to extract them and save them in the system as indicators, so that they can be reviewed at a later stage for manual review. System performance may be better as the playbook flow does not stop to extract, but if the incident contains indicators that are needed or expected in the playbook execution flow, inline should be used, as it will not execute the playbook before all indicators are extracted from the incident.

###### NOTE:

When using Out of band, the extracted indicators do not appear in the context. If you want the extracted indicators to appear select Inline.

- **Use system default:** Indicators are extracted according to the following defaults:

Component	Description	Default
Incident creation	Sets the indicator extraction mode for incident creation. It extracts from all associated fields at the point of incident creation. You can change the value when editing an incident type.	Inline
Incident field change	Sets the indicator extraction mode for incident field change. You can change the value when editing an incident type.	Out of band
Tasks	Applies to the result of the task. You can change the value when editing a task.	None
Manual	Applies to commands triggered from the CLI. You can change the value when using the indicator extraction parameter.	Out of band

### 16.2.3.2 | Create indicator extraction rules for an incident type

#### Abstract

Create indicator extraction rules for an incident type. Customize indicator extraction in Cortex XSOAR.

You can extract indicators from incident fields on creation of an incident and when a field changes. For example, you might want to extract the IP address upon incident creation and again when the field changes.

NAME	USE FIELD VALUE	INDICATOR TYPES TO EXTRACT	⋮
abc Destination Network	<input type="checkbox"/>	IP <input type="button" value="x"/>	<input type="button" value="⋮"/>

The indicator extraction feature extracts indicators from incident fields and enriches them using commands and scripts defined for the indicator type.

1. Go to Settings & Info → Settings → Object Setup → Incidents → Types.
2. For a content pack installed incident type, detach or duplicate the incident type, and then click the detached or duplicated incident type. For custom incident types, click the incident type.
3. From the Indicators Extraction Rules tab, in the On incident creation and the On field change fields, select the required indicator extraction mode. If you select Out of band, the extracted indicators do not appear in the context. If you want the extracted indicators to appear, select Inline. For more information, see Indicator extraction modes.
4. In the What to Extract section, if you want to extract all incident fields, select Extract all indicators from all fields.
5. If you want to choose which indicators are extracted according to each field, select Extract specific indicators.

You can search and filter the incident fields. For each field, use the dropdown menu to control the indicator types to extract:

(Optional) You can select all indicators, set all indicators to none, or copy settings from an incident type by clicking ⋮ (to the right of the table's column headers).

Indicator Type To Extract	Description
None	No indicators are extracted.
All indicator types with regex	<p>Some indicator types are associated with a regex (such as IP), and some are not (such as Registry Key).</p> <p>Only indicators that are associated with a regex are extracted.</p>
Specific indicator types	You can choose one or more indicator types based on regex. The system extracts values that match the regex from this incident field.

Select the Use field value checkbox, to use any indicator based on the field value (not regex based). This creates an indicator out of the entire value of the field, regardless whether the indicator type has a configured regex. This can be used in cases such as extracting hostnames.

NAME	USE FIELD VALUE	INDICATOR TYPES TO EXTRACT	⋮
abc Phone Number	<input checked="" type="checkbox"/>	All indicators (with regex) <input type="button" value="x"/>	<input type="button" value="⋮"/>
abc Alert Name	<input checked="" type="checkbox"/>	All indicators (with regex) <input type="button" value="x"/>	<input type="button" value="⋮"/>
abc Destination Hostname	<input checked="" type="checkbox"/>	Domain <input type="button" value="x"/>	<input type="button" value="⋮"/>
abc Country Code	<input checked="" type="checkbox"/>	All indicators (with regex) <input type="button" value="x"/>	<input type="button" value="⋮"/>

#### NOTE:

- We recommend turning off (setting to None) incident extraction for the Labels incident field. When an incident JSON is received from an integration, the JSON members are mapped to incident fields (based on the mapping configuration). Every member in the JSON that was not mapped to a field, will be written to the Labels field. If the Labels field extracts indicators, it can expose unmapped or unknown data to external sources. You should only map the relevant data to fields and set their extraction settings.
- If you want to extract attachments, select the attachment field and then select File as the indicator type to extract. The File extracts a hash (usually SHA-256), which can be viewed in the War Room. You may want to disable indicator extraction for attachments to reduce external API usage and protect restricted data (the hash) from being sent.

6. Click Save.

7. (Optional) If you want to configure which scripts and commands the indicator type executes, go to Settings & Info → Settings → Object Setup → Indicators → Types and edit or Create an indicator type.

Add scripts and reputation commands for the indicator type. When indicator extraction occurs, indicators are extracted as defined in an indicator type, and enriched using the commands and scripts associated with the indicator type. For example, the URL indicator is enriched using the !ur1 command.

In this example, if an email is forwarded that potentially includes phishing, we want to extract at incident creation (inline) and upon a field change (out of band):

- Campaign Email Subject: Extract all indicators.
- Campaign Email Body: Extract all indicators.
- Email Delete Result: Extract email only.
- Email Delete Reason: Extract email only.

NAME	USE FIELD VALUE	INDICATOR TYPES TO EXTRACT
Campaign Email Subject	<input type="checkbox"/>	All indicators (with regex)
Campaign Email Body	<input type="checkbox"/>	All indicators (with regex)
Email Delete Result	<input type="checkbox"/>	Email
Email Delete Reason	<input type="checkbox"/>	Email

#### 16.2.3.3 | Set the indicator extraction mode for a playbook task

##### Abstract

Create indicator extraction rules for a playbook task in Cortex XSOAR. Auto extract for a playbook task. Edit task. Use case indicator extraction.

You can set the indicator extraction mode for specific playbook tasks.

1. Select the playbook where you want to add indicator extraction to a task, and click Edit.
2. In the playbook, click a task to open the Edit Task window.
3. Click the Advanced tab.
4. In the indicator extraction drop-down menu, select the mode you want to use.
5. Click OK.

#### 16.2.3.4 | Disable indicator extraction for scripts or integrations

##### Abstract

Disable indicator extraction for a specific script or integration in Cortex XSOAR.

This procedure describes how to disable indicator extraction for a specific script or an integration.

- To disable indicator extraction for a script, add the `IgnoreAutoExtract` entry with the value of `true`, when returning an entry.

For example:

```
entry = {
    'Type': entryTypes['note'],
    'Contents': {
        'Echo': demisto.args()['echo']
    },
    'ContentsFormat': formats['json'],
    'ReadableContentsFormat': formats['markdown'],
    'HumanReadable': hr,
    'IgnoreAutoExtract': True
}
```

- To disable indicator extraction for an integration, add the '`IgnoreAutoExtract`' entry with the value of `true`, when returning an entry.

For example in the ServiceNow integration:

```
entry = {
    'Type': entryTypes['note'],
    'Contents': result,
    'ContentsFormat': formats['json'],
    'ReadableContentsFormat': formats['markdown'],
    'HumanReadable': tableToMarkdown('ServiceNow ticket', hr, headers=headers, removeNull=True),
    'EntryContext': {
        'Ticket(val.ID==obj.ID)': context,
        'ServiceNow.Ticket(val.ID==obj.ID)': context
    },
    'IgnoreAutoExtract': True
}
entries.append(entry)
return entries
```

For more information about command results in Python, see [Python code conventions for CommandResults](#).

#### 16.2.3.5 | Troubleshoot indicator extraction

If indicators are not extracting, check whether the indicator mode is set to none. Even if you select the relevant incident fields and the indicators to extract, if the mode is set to none, indicators do not extract.

When creating new incident types, if you select Extract all indicators from all fields, all fields are extracted including custom fields. If you select Extract specific indicators by default, indicator extraction for new custom fields is set to none.

#### 16.2.4 | Configure the indicator timeline

##### Abstract

Add a server configuration to manage the indicator timeline in Cortex XSOAR and improve indicator timeline performance.

The indicator timeline displays a list of dates and events that affect the timeline, such as change of verdict and traffic light protocol.

A large number of indicators can affect the performance of the indicator timeline. You can configure advanced server configurations to manage indicator timeline performance.

- Select Settings & Info → Settings → System → Server Settings → Server Configuration → Add Server Configuration.
- Add the following server configurations.

Key	Value	Description
<code>indicator.timeline.enabled</code>	<code>true</code> or <code>false</code>	Enables the indicator timeline in all flows. The default is <code>true</code> .
<code>indicator.timeline.auto.extract.enabled</code>	<code>true</code> or <code>false</code>	Enables the indicator timeline in the indicator extraction flow. The default is <code>true</code> .

## 16.2.5 | Configure indicator expiration

### Abstract

Cortex XSOAR indicators have an active or expired status which can be set to expire after a specific period or never to expire. Set default expiration method.

Indicators can have the Expiration Status field set to Active or Expired, which is determined by the Expiration field. When indicators expire, they still exist in Cortex XSOAR, meaning they are still displayed and you can still search for them. A job that runs every hour checks for newly expired indicators and updates the Expiration Status field.

When indicators expire, the expiration status and expiration fields are updated. You can use it to take actions based on indicator expiration. For more information, see Indicator field trigger scripts.

You can set the default expiration method for indicators either to never expire or to expire after a specific period. The default expiration method is set by the indicator type. For more information see Indicator type profile.

The following table shows the hierarchy by which indicators are expired.

Method	Description
Manual	<p>Manually expire the indicator either in the indicator layout or CLI. This method overrides all other methods.</p> <p><b>NOTE:</b></p> <p>You need a TIM license to access the indicator layout.</p> <p>Use the <code>!expireIndicators</code> command to change the expiration status to Expired for one or more indicators. This command accepts a comma-separated list of indicator values and supports multiple indicator types. For example, you can set the expiration status for an IP address, domain, and file hash: <code>!expireIndicators value=1.1.1.1,safeurl.com,45356A9DB614ED7161A3B9192E2F318D0AB5AD10</code></p> <p>Use the <code>!setIndicator</code> or for multiple indicators use the <code>!setIndicators</code> command to reset the indicators' expiration value. The value can also be set to <code>Never</code>, so that the indicators never expire. For example, <code>!setIndicators indicatorsValues=watson.com expiration=Never</code>.</p> <p>You can also use these commands in a script, but the user can override this if running a command in the CLI or the indicator layout.</p>
Feed integration	<p>Some integrations support setting the expiration method on an integration instance level, which overrides the method defined for the indicator type.</p>
Indicator type	<p>The expiration method (interval or never) is defined according to indicator type, which applies to all indicators of this type. This is the default expiration method for an indicator.</p>

## 16.2.6 | Configure Threat Intel feed integrations

You can download and install Threat Intel content packs including the following Threat Intel integrations such as:

- MITRE ATT&CK
- Unit 42 ATOMs
- Unit 42 Intel Objects Feed

If you have a TIM license, this feed is preinstalled.

- AlienVault
- AWS

**NOTE:**

If you have a TIM license you can set up unlimited feeds. If not, you are limited to 5 active feeds and 100 indicators. For more information, see Understand Cortex XSOAR licenses.

### How to configure threat intel feed integrations

1. Go to Marketplace and install the relevant Threat Intel content pack.

2. Configure the Threat Intel integration by going to Settings → Settings & Info → Integrations → Instances, search for your integration, and click Add Instance.

The following table is a non-exhaustive list of the most common feed integration parameters. Each feed integration may have parameters unique to that integration. Read the documentation for specific feed integrations for more details.

Parameter	Description
Fetches indicators	Select this option for the integration instance to fetch indicators.  Some integrations can fetch indicators or incidents. Select the relevant option for what you need to fetch in the instance.
URL	The URL of the feed.
Feed Fetch Interval	When the integration instance should fetch indicators from the feed.
Indicator verdict	The indicator verdict that will apply to all indicators fetched from this integration instance. See Indicator verdict.
Source reliability	The reliability of the source that provides the threat intelligence data.
Indicator Expiration Method	The method by which to expire indicators from this integration instance. The default expiration method is the interval configured for the indicator type to which this indicator belongs. <ul style="list-style-type: none"> <li>• Indicator Type: The expiration method defined for the indicator type to which this indicator belongs (interval or never).</li> <li>• Time Interval: Expires indicators from this instance after the specified time interval, in days or hours.</li> <li>• Never Expire: Indicators from this instance never expire.</li> <li>• When removed from the feed: When the indicators are removed from the feed they are expired in the system.</li> </ul>
Bypass exclusion list	When selected, the exclusion list is ignored for indicators from this feed. This means that if an indicator from this feed is on the exclusion list, the indicator might still be added to the system.
Trust any certificate (not secure)	When selected, certificates are not checked.
Use system proxy settings	Runs the integration instance using the proxy server (HTTP or HTTPS) when an engine is selected.
Do not use in CLI by default	Excludes this integration instance when running a generic command that uses all available integrations.

#### 16.2.7 | Configure Threat Intelligence Management playbooks to process indicators

##### Abstract

Jobs trigger TIM playbooks and process large numbers of indicators. TIM playbook configuration and settings.

TIM (Threat Intelligence Management) playbooks run on an indicator search query and are used for processing large numbers of incoming indicators from feeds. Feed integrations enable you to ingest indicators from external sources into Cortex XSOAR. Once indicators are in Cortex XSOAR, they can be enriched and assigned a verdict. Enriched indicators can be used for incident investigations in Cortex XSOAR and can be pushed to a SIEM or other external system.

The TIM playbook performs an indicator query. For example, the query might return indicators using the `from-feed` tag. The TIM playbook runs using the indicators matching the query as an input. When configuring your TIM Playbook to use an indicator query, we recommend you first run your query on the main Threat Intel page, which enables you to view the indicators returned and verify you have the results you need for your playbook. Copy and paste the query into the playbook or save the query that you ran on the Threat Intel page and access that saved query from the playbook. Queries use a modified Lucene syntax.

#### NOTE:

By default, a query run on the Threat Intel page is limited to the last 7 days, unless otherwise specified. This same limit does not apply when you enter the query in Playbook Inputs and Outputs, but you can add your required time filter to the query.

If you do not have a TIM license, there are several limitations, such as the number of active feeds and indicators. For more information, see Understand Cortex XSOAR licenses.

#### Large batches of indicators

In most cases, the following workflow applies:

If more than 1000 indicators are returned, the indicators are processed in batches of 1000. For example, if there are 4000 indicators returned, the playbook runs the first time on the first 1000. Each task receives 1000 indicators as a list, or if the task does not support lists, loops over the 1000 indicators. When the playbook reaches the end, it runs again with the next batch of 1000 indicators and repeats until all indicators have been processed. The playbook loops automatically through batches of indicators, you do not need to configure the playbook to loop. After all indicators have been processed, the playbook automatically closes the incident. You do not need to include a close incident task.

#### Quiet mode

TIM playbooks often process thousands of indicators. By default, quiet mode is enabled for TIM Playbooks. In quiet mode, entries are not written to the War Room and inputs and outputs are not presented for Work Plan tasks. For troubleshooting purposes, you can temporarily disable quiet mode during playbook development. Quiet mode can be disabled in the playbook settings or on a per-task basis.

We strongly recommend that you have quiet mode enabled for any playbook that is in production, to prevent possible performance issues.

#### NOTE:

While quiet mode is disabled, any changes you make to the playbook indicators query will turn quiet mode back on.

#### TIM playbook tasks

The Playbook search query returns all of the indicators that match a particular search, including all fields for each indicator. Individual tasks may only require a subset of that data. If you need to run different tasks for different types of indicators, use a conditional task and set the input to check for the indicator type. For example, in the TIM - Indicator Auto Processing playbook, the Are there IP results? conditional task searches for any IP indicators. If it finds any IP indicators, the condition is met.

The screenshot shows a configuration interface for a conditional task. At the top, there are tabs: Condition, Details, Timers, Advanced, and On Error. The Condition tab is selected. Below the tabs, it says "Condition for: yes". To the right, there is a "Remove condition" link. The main area contains two boxes connected by an equals sign. The left box is labeled "Get playbookQuery.indicator\_type" and the right box is labeled "Get IP". Between them is the text "Equals (String)".

If no IP indicator types are found, the condition is not met and the playbook proceeds to the else branch.

You can also use filters based on indicator attributes. For example, you can limit a task to only run on indicators where the type is IP.

#### NOTE:

In the Get field, if you change `playbookQuery.indicator_type` to `playbookQuery.value` it returns the indicator values, such as the IP addresses. Using `playbookQuery` returns all of the indicator attributes, not only the indicator value.

#### Process indicators using a TIM playbook workflow

In most cases, the following workflow applies:

1. Indicators are added to Cortex XSOAR through feed ingestion. You can configure your integration to automatically tag all new/updated indicators from a particular instance. For example, you can tag them using the `from-feed` tag.
2. Customize a TIM playbook to process the indicators.

3. Define a job to run that triggers the playbook when the indicators are fetched.

When a feed has been completed and there is a change of content you can add a TIM playbook to process indicators to a job. Create a Job Triggered by Delta in a Feed that runs when the ingestion is completed. The job runs a TIM playbook, which performs an indicator query. For example, the query might return indicators using the `from-feed` tag, and that were added or modified since the last time the job that triggered the playbook was run.

4. If you want to push the enriched indicators to a SIEM, you can set up a time-triggered job to run a playbook.

To see how you can use a job to process indicators, see [Create jobs to process indicators example](#).

## 16.3 | Export indicators

### Abstract

Export indicators from the Indicators table, using an integration, or playbook, or set up an External Dynamic list (EDL) by using the Generic Export Indicators integration.

In the Indicators table, you can export indicators in a CSV or STIX file. You can also export indicators using an integration or a playbook.

### Export indicators using the Generic Export Indicators Integration

You can export indicators in a hosted text file (External Dynamic list) from Cortex XSOAR or an engine using the Generic Export Indicators Service integration. Exported indicators can be used for example in firewall block lists, allow lists, and monitoring and analysis in Splunk. See [Generic Export Indicators Service](#).

The Generic Export Indicators Service integration can be configured to export specific fields in different output formats. Multiple instances of the integration can be configured for different indicator queries, and the output can be customized to work with a variety of third-party services.

You can set up the Generic Export Indicators Service integration by setting up a long-running integration. See [Forward requests to long-running integrations](#).

If you configure the Generic Export Indicator to run on-demand, use the `!export-indicators-list-update` command for the first time to initialize the export process.

### Export incidents and indicators to CSV using the UTF8-BOM format

By default, when exporting an incident or an indicator to CSV format, Cortex XSOAR generates the report in UTF8 format. If you need to export an incident or an indicator that contains Cyrillic characters such as Russian or Greek, you need to change the format to UTF8-BOM.

#### NOTE:

When changing the format to UFT8-BOM you also change the format for incidents.

1. Select Settings & Info → Settings → System → Server Settings → Add Server Configuration.
2. Add the following key and value.

Key: `export.utf8bom`

Value: `true`

3. Save the server configuration.

### Export indicators using playbooks

Cortex XSOAR provides numerous out-of-the-box playbooks for TIM, including playbooks that enable you to export indicators. All TIM-related playbooks have the 'TIM' prefix. Some are generic (for example, TIM - Process Indicators - Fully Automated), and some are dedicated to a specific vendor, like QRadar (for example, TIM - QRadar Add Domain Indicators) and ArcSight (for example, TIM- Arcsight Add IP Indicators).

#### NOTE:

For TIM-related playbooks, you need a TIM license.

If you define a playbook task input that pulls from indicators, the entire playbook runs in Quiet Mode. This means the task or playbook information is not written to the War Room, and inputs and outputs are not displayed in the playbook. However, errors and warnings are still written to the War Room.

#### CAUTION:

You should not run a query on a field that you might change in the playbook flow. For example, you shouldn't have a playbook with query `Verdict:Malicious` and then change the indicator verdict as a part of the playbook.

## 16.4 | Customize Threat Intel Reports

### Abstract

Set up and customize threat intel report types in Cortex XSOAR.

Threat intel reports summarize and share threat intelligence research conducted within your organization by threat analysts and threat hunters. Threat intelligence reports help you communicate the current threat landscape to internal and external stakeholders, whether in the form of high-level summary reports for C-level executives, or detailed, tactical reports for the SOC and other security stakeholders.

#### **NOTE:**

To customize and manage Threat Intel Reports, you must have a TIM license.

Threat intel reports help address multiple relevant reporting use cases:

- Global cybersecurity threats

Report to colleagues and executives if, and how, such threats affected your organization, and what was done to remediate and prevent future attacks.

- Periodic monitoring

Keep track of infiltration attempts by adversaries within your industry vertical, and publish periodic status updates on any new behaviors.

- Open-source intelligence (OSINT) reports

Aggregate highlights of external publications that should be actively brought to the attention of your SOC. This is usually done to ensure that relevant employees are up-to-date with the latest security trends so they can make more informed decisions. For a practical example, see Threat Intel Management use cases.

- Threat hunting

Report to colleagues, and the larger threat intelligence community about proactive searches and detection of advanced threats not found by traditional prevention and detection tools.

Each report consists of the following:

- **Report type:** Determines which report types your organization needs. Each type has an associated layout. You can create report types and report layouts, or customize existing ones. When analysts create a report, they select the report type.
- **Report layout:** Ensures the most relevant information is shown for each report type. The layout includes customizable fields for your use case.
- **Report fields:** Create fields or add existing fields to report layouts. After a report is created, the analyst can populate the report with relevant data.

Cortex XSOAR Threat Intel Management comes out-of-the-box with the following report types and layouts:

Report Type	Report Layout	Description
Campaign	Campaign Report	Describes a campaign run by a threat actor. Includes fields such as Campaign Details and a free text field to add the threat type, origin, etc.
Executive Brief	Executive Brief Report	Used for an executive summary or any kind of generic report.
Malware	Malware Report	A report tailored for malware such as Operating System, Aliases, and Malware type.
Threat Actor	Threat Actor Report	A report tailored for Threat Actors with a special section for Threat Actor metadata, such as the Threat Actor's name, goals, and motivation.
Vulnerability	Vulnerability Report	A report tailored for vulnerability with a special section for vulnerability details such as CVE and CVSS.

These report types, layouts, and fields are part of the Threat Intel Reports (Beta). For more details including screenshots, see Threat Intel Reports (BETA).

#### **NOTE:**

By default, when editing the dropdown or text values in a threat intel report, the changes are not saved until you confirm your changes (clicking the checkmark icon in the value field).

These icons are designed to let you have an additional level of security before you make changes to the fields in threat intel reports, incidents, and indicators.

To change the default behavior set the `inline.edit.on.blur` server configuration to `true`, which enables you to make changes to inline fields without clicking the checkmark. The changes are automatically saved when clicking anywhere on the page or when navigating to another page. For text values, you can also click anywhere in the value field to edit.

#### 16.4.1 | Create a Threat Intel Report type

##### Abstract

Create or detach a Threat Intel Report type to suit your use case.

Threat intel reports are categorized by type, which determines the layout that is displayed for the report.

You can create new threat intel report types to support use cases not covered by the out-of-the-box types, which may require different report layouts. For example, you may want to add a new type for a specific threat-hunting report that your organization needs, which is not covered by one of the out-of-the-box template types.

If you want to customize a report type by attaching a new layout, you need to detach the existing report type. If you detach it, it does not receive content pack updates. If you reattach it, any content pack updates override any changes made.

##### NOTE:

If you disable the report, it is not available for selection when you create a report.

You cannot delete out-of-the-box report types.

Before creating a Threat Intel Report type, review, customize, or create a new layout, which is then added to the report type.

1. Select Settings & Info → Settings → Object Setup → Threat Intel Reports → Types → New Type.

2. Enter a name and select the layout you want to use.

You can leave this blank and add the layout later.

3. (Multi-tenant only) Add or select Propagation labels. You can also view any dependencies.

4. Save the type.

#### 16.4.2 | Create a Threat Intel Report field

##### Abstract

Create a Threat Intel Report and add it to a report layout.

Add/create Threat Intel Report fields to populate a report layout with relevant data.

##### Field types

Field Type	Description
Boolean	Checkbox
Date picker	Adds the date to the field
Grid (table)	Include an interactive, editable grid as a field type for selected report types or all report types. To see how to create a grid field and to use a script, see Create a grid field for an incident type.  When you select Grid (table) you can format the table and determine if the user can add rows,
HTML	HTML: Create and view HTML content, which can be used in any type of report.

Field Type	Description
Long text	<ul style="list-style-type: none"> <li>Long text is analyzed and tokenized, and entries are indexed as individual words, enabling you to perform advanced searches and use wildcards.</li> <li>Long text fields cannot be sorted and cannot be used in graphical dashboard widgets.</li> <li>While editing a long text field, pressing enter will create a new line. Case insensitive.</li> </ul> <p>Add a placeholder if required.</p>
Markdown	<p>Add markdown-formatted text as a Template which will be displayed to users in the field after the indicator is created. Markdown lets you add basic formatting to text to provide a better end-user experience.</p>
Multi select/Array	<p>Select the following options:</p> <ul style="list-style-type: none"> <li>Multi-select from a prefilled (static) list.</li> <li>An empty array field for the user to add one or more values as a comma-separated list.</li> </ul> <p>Add a placeholder if required.</p>
Number	<p>Can contain any number. Default is 0.</p>
Role	<p>The role assigned to the Threat Intel Report determines which users (by role) can view the report.</p>
Short Text	<ul style="list-style-type: none"> <li>Short text is treated as a single unit of text and is not indexed by word. Advanced search, including wildcards, is not supported.</li> <li>Short text fields are case-sensitive by default but can be changed to case-insensitive when creating the field.</li> <li>While editing a short text field, pressing enter will save and close.</li> <li>Maximum length 60,000 characters.</li> </ul> <p>Recommended use is one-word entries, such as username and email address.</p> <p>Select a placeholder, if required.</p>
Single select	<p>Select a value from a list of options. Add comma-separated values.</p>
Tags	<p>Accepts a single tag or a comma-separated list, not case-sensitive.</p> <p>Add a placeholder if required.</p>
Timer/SLA	<p>Set up when an SLA is due, the risk threshold, and configure actions to take if the SLA does pass.</p>
URL	<p>Add a URL when completing the field.</p>
User	<p>A user in Cortex XSOAR.</p>

#### How to create a new field

1. Select Settings & Info → Settings → Object Setup → Threat Intel Reports → Fields → New Field.
2. Select the relevant field type.
3. Complete the following fields:

Parameter	Description
Mandatory	If selected, this field is mandatory when used in a form.
Field Name	A meaningful display name for the field. After you type a name, you will see below the field that the Machine name is automatically populated. The field's machine name is applicable for searching and the CLI.
Tooltip	An optional tooltip for the field.

4. Configure the attributes:

Name	Description
Script to run when field value changes	The script dynamically changes the field value when script conditions are met. For a script to be available, it must have the <b>field-change-triggered-ThreatIntelReport</b> tag, which is added when defining a script.
Run the field triggered script after the new field value is saved	By default, the script executes before the threat intel report is stored in the database. If you select this option, the script instead executes after the threat intel report is modified, so that the script cannot make changes to the threat intel report.
Add to all Threat Intel Report types	Determines which threat intel report types have this field available. By default, fields are available to all types. To change this, clear the checkbox and select the specific threat intel report types.
Make data available for search	Determines if the values in these fields are available when searching. Enabled by default.

5. (Multi-tenant only) In the Propagation tab, add or select Propagation labels. You can also view any dependencies.

#### 16.4.3 | Create a Threat Intel Report layout

##### Abstract

Configure threat intel report layouts. Add script-based content in the layout.

You can customize almost every aspect of the layout, including which tabs appear, in which order they appear, who has permission to view the tabs, which information appears, and how it is displayed.

In the Object Setup → Threat Intel Reports → Layouts tab, you can view out-of-the-box layouts and any custom layouts. Each out-of-the-box layout is attached to the out-of-the-box Threat Intel Report types.

If you want to customize an existing layout, you can detach it without creating or duplicating another one. When a layout is detached, it does not receive content pack updates.

##### TIP:

If you detach a layout, make edits, and later want to receive content pack updates for that layout, we recommend you duplicate the report layout before reattaching the original, to protect your changes from content pack updates.

##### Step 1. Create a Threat Intel Report layout

The following procedure describes how to create a new layout, but you can follow similar steps to customize an existing layout.

1. Select Settings & Info → Settings → Object Setup → Threat Intel Reports → Layouts → New Layout.

2. To add a description click Settings.

(Multi-tenant only) Add or select Propagation labels. You can also view any dependencies.

3. Customize the tabs by clicking the settings wheel icon and then doing the following:

**NOTE:**

You can click and drag a tab to reorder the tabs.

Action	Description
Rename	You can also edit a tab's name by clicking the tab.
Duplicate	Copies the existing tab.
Delete	Deletes the tab.
Show empty fields	The setting that you configure in the layout becomes the default value seen in the report for the specific tab, which can then be overridden.  You can also set a global default value using the <code>UI.summary.page.hide.empty.fields</code> server configuration, which can also be overridden for a specific tab.
Hide tab	Hides the tab. Rather than deleting the tab, you may want to use the tab again for future use.
Format for exporting	Build your layout based on A4 proportions to match the format used for exporting. Selecting this option hides the tab by default, but the tab will remain available for export.
Viewing Permissions	Select which roles can view the tabs.
Display Filter	Add or view a filter applied to the tab. If the filters apply, the specific fields or tabs are shown in the layout. If the mandatory field is not shown in the layout, the user is not obliged to complete it.

4. From the LIBRARY section, drag and drop the following sections:

Section	Description
New Section	After creating a new section, click the Fields and Buttons tab and drag and drop the fields as required.  When hovering over a field, click the eye icon to add a filter to the field.
General Purpose Dynamic Section	Add a script to the layout, such as adding a script to create a button on the layout that sets a threat intel report as published. For more information, see Step 2. (Optional) Add a script to the Threat Intel Report layout.
Relationships	The user can manually create indicator relationships between the report and an indicator. For more information about indicator relationships, see Manage indicator relationships.

5. Define the section properties.

Determine how a section appears in the layout, such as name and showing the section header. In most sections, you can also configure the fields to appear in rows, or as cards, and wrap the text labels. For example, if you know that some of the field values are very long, use rows. If the field values are short, use cards so you can fit more fields in a section.

a. Click the section, click the pencil icon, and then select Edit section settings.

b. Edit the section as required and click OK.

**NOTE:**

To remove or duplicate click the pencil icon in the section, and select the relevant option.

6. If relevant, create a New tab and repeat the steps as required.

7. When finished, save the layout.

**Step 2. (Optional) Add a script to the Threat Intel Report layout**

You can add content to threat intel report layouts, based on a script. You need to add the General Purpose Dynamic Section when editing layouts.

The General Purpose Dynamic Section allows you to configure a section in a layout tab from a script. The script can return text, markdown, or HTML, the results of which appear in the General Purpose Dynamic Section. You can add any required information from a script. Before you begin, you need to create a script.

The following is an example of a script that can be added. This script can be used to add a button to the layout that sets a threat intel report as published.

```
def publish():
    now_utc = datetime.now(timezone.utc)
    object = demisto.args('object')
    object_id = object.get('id')
    roles = execute_command('getRoles', {})

    execute_command(
        'setThreatIntelReport',
        {
            'id': object_id,
            'xsoarReadOnlyRoles': demisto.dt(
                roles, 'DemistoRoles.name'
            ),
            'reportstatus': 'Published',
            'published': now_utc.isoformat(),
        },
    )

    demisto.results('ok')

if __name__ in ('__main__', '__builtin__', 'builtins'):
    publish()
```

1. Edit the relevant threat intel report layout.

2. Drag and drop the General Purpose Dynamic Section onto the layout.

3. Select the General Purpose Dynamic Section, click , and then Edit section settings.

4. In the Name and Description fields, add a meaningful name and a description for the dynamic section that explains what the script displays.

5. In the Automation script field, from the dropdown list, select the script that returns data for the dynamic section.

**NOTE:**

Only scripts to which you have added the general-dynamic-section tag appear in the dropdown list.

6. Click OK.

7. Save the layout.

**Step 3. Add the layout to the Threat Intel Report type**

1. Go to Settings & Info → Settings → Object Setup → Threat Intel Reports → Types.

2. Select the report type and click Edit.

If the report type is an out-of-the-box type from a content pack you need to detach the report. Otherwise, you need to create a new report.

3. In the Layout field, from the dropdown list, add the customized layout.

4. Save the report type.

5. (Optional) If you have created a new layout (not detached), you can do the following:

- Contribute it to Marketplace.
  1. From Marketplace , in the Contributions tab, click Contribute Content. From the dropdown menu, select Layouts, Add the new layout you want to contribute to Marketplace and click Save and Contribute.
  2. Complete the information in the Contribute form and click Contribute.
- If using a dev/prod environment, in the development machine push the layout to the prod machine.
- (Multi-tenant) In the Main tenant propagate it to the child tenant.

## 16.5 | Indicator management

### Abstract

Perform actions (create, edit, export, delete) and search for indicators on the Cortex XSOAR Threat Intel page.

Indicators are artifacts associated with security incidents and are an essential part of the incident management and remediation process. They help correlate incidents, create hunting operations, and enable you to easily analyze incidents and reduce Mean Time to Response (MTTR).

If you have a TIM license, Cortex XSOAR Threat Intel includes access to the Unit 42 Intel service, enabling you to identify threats in your network and discover and contextualize trends. Unit 42 Intel provides data from WildFire (Palo Alto Networks' cloud-based malware sandbox), the PAN-DB URL Filtering database, Palo Alto Networks' Unit 42 threat intelligence team, and third-party feeds (including both closed and open-source intelligence). Unit 42 Intel data is continually updated to include the most recent threat samples analyzed by Palo Alto Networks, enabling you to keep up with threat trends and take a proactive approach to securing your network.

The Threat Intel page is split into the following tabs:

- Indicators
- Sample Analysis
- Sessions and Submissions
- Threat Intel Reports

#### NOTE:

If you don't have a TIM license, you can only view the Indicators tab. For more information, see Manage indicators.

### Indicators

Displays a list of indicators added to Cortex XSOAR, where you can perform several indicator actions, including adding Unit 42 data.

#### NOTE:

If you are unable to perform a specific action or view data, you may not have sufficient user role permissions. Contact your Cortex XSOAR administrator for more details.

You can perform the following actions on the Threat Intel page.

Action	Description
Investigate an indicator	Click on an indicator to view and take action on the indicator.
Create an indicator	<p>Indicators are added to the Indicators table from incoming incidents, feed integrations, adding Unit 42 data, or manually creating a new indicator.</p> <p>When creating an indicator, in the Verdict field, you can either select a verdict or leave it blank to calculate it by clicking Save &amp; Enrich, which updates the indicator from enrichment sources. After you select an indicator type, you can add any custom field data.</p> <p><b>NOTE:</b></p> <p>In the CLI, you can run the !createNewIndicator command.</p>
Create an incident	Create an incident from the selected indicator and populate relevant incident fields with indicator data.

Action	Description
Edit	Edit a single indicator or select multiple indicators to perform a bulk edit.
Delete and Exclude	Delete and exclude one or more indicators from all indicator types or a subset of indicator types. If you select the Do not add to exclusion list checkbox, the selected indicators are only deleted.
Export CSV	Export the selected indicators to a CSV file. By default, the CSV file is generated in UTF8 format. Administrator permission is required to update server configurations, including changing the format, see Export incidents and indicators to CSV using the UTF8-BOM format.
Export STIX	Export the selected indicators to a STIX file.
Upload a STIX file	To upload a STIX file, click the upload button (top right of the page) and add the indicators from the file.

**NOTE:**

By default, when editing a list or text values in an incident/indicator, the changes are not saved until you confirm your changes (clicking the checkmark icon in the value field). These icons are designed to give you additional security when updating fields in incidents, indicators, and Threat Intel Reports.

You can change this default behavior by updating the server configuration. You need administrator permission to update server configurations. For more information, see Configure inline value fields.

**Sample analysis**

Unit 42 Intel provides sample analysis for files. This helps you conduct in-depth investigations, find links between attacks, and analyze threat patterns. If the file indicator is in the Unit 42 Intel service, you have access to a full report on activities, properties, and behaviors associated with the file. In addition, you can see how many other malicious, suspicious, or unknown file samples included the same activities, properties, and behaviors, and also build queries to find related samples. For more information, see Investigate files using sample analysis.

**Sessions and Submissions**

Unit 42 Intel provides in-depth information on device communication.

Cortex XSOAR users can use their sessions and submissions data for investigation and analysis. Sessions and Submissions data are available for users with the following products:

- **Firewall** - Samples that a Palo Alto Networks firewall forwarded to WildFire.
- **WildFire Appliance** - Samples that a WildFire appliance submitted to the WildFire public cloud.
- **Cortex XDR** - Samples submitted through Cortex XDR.
- **Prisma SaaS** - Samples submitted through Prisma SaaS.
- **Prisma Access** - Samples submitted through Prisma Access.

For example, if you have a file indicator that has been determined as malicious, and you have a Cortex XDR integration configured, in the Sessions & Submissions tab, you can see where this file came from and where it is in your network by viewing the firewall sessions this file passed through. You can see which XDR agents in your system reported the file, which tells you which machines might be infected. You can block the external IP address with your firewall, and, if needed, isolate the affected machines to contain the attack. If the source is internal, you can investigate that endpoint. For more information, see Use sessions and submissions in your investigation.

**Threat Intel Reports**

Threat Intel Reports summarize and share threat intelligence research conducted within your organization by threat analysts and threat hunters. Threat Intel Reports help you communicate the current threat landscape to internal and external stakeholders, whether in the form of high-level summary reports for C-level executives, or detailed, tactical reports for the SOC and other security stakeholders. For more information, see Manage Threat Intel Reports.

**16.5.1 | Query indicators with Unit 42 Intel data****Abstract**

How to query indicators in the threat intel library and in Unit 42 Intel.

You can access Threat Intel data through the following methods:

- On the Threat Intel page, select an indicator to start investigating. If the indicator also exists in Unit 42 Intel, the Unit 42 Intel tab is available.
- When investigating an incident, select an extracted indicator. The Quick View shows basic information about the indicator in Cortex XSOAR and Unit 42 (if available). Full view shows the full Cortex XSOAR indicator summary.
- On the Threat Intel page, query an indicator, which may or may not be in the Cortex XSOAR intel library.

Unit 42 Intel data is cloud-based and remotely maintained so that you can view data from Unit 42 Intel and add only the information you need to your Cortex XSOAR threat intel library. When you search for an IP address, domain, URL, or file, you can view the indicator in Cortex XSOAR and the additional information provided by Unit 42 Intel. When an indicator does not yet exist in Cortex XSOAR, but does exist in Unit 42 Intel, you can add the indicator to the Cortex XSOAR threat intel library. You can add the indicator and enrich it with your existing integrations, or add the indicator without enrichment. When the indicator already exists in Cortex XSOAR, but additional information is available from Unit 42 Intel, you can update your indicator with the most recent data from Unit 42 Intel.

The Threat Intel library is a centralized space for all indicators, whether they are found in an incident, brought in as a feed, or added manually. You can view in-depth information on collected indicators and filter the library based on common attributes.

#### NOTE:

You can search or look up indicators. A search, which can include wildcards and complex queries, can return multiple results. Searches are only performed in Cortex XSOAR. Lookups are exact values, are performed in both Cortex XSOAR and Unit 42 Intel data, and can only return one result.

#### Indicator query considerations

When querying directly on the Threat Intel page, the following considerations apply:

- Querying an IP address, domain, URL, or SHA256 file hash, without a wildcard or complex search (Boolean search, type:file, etc.), queries both the Cortex XSOAR threat intel library and Unit 42 Intel, with no date range limit.
- If you enter an indicator type that is not an IP address, domain, URL, or SHA256 file hash, or you enter a wildcard or complex option (Boolean search, type:file, etc.), no lookup is performed in Unit 42. In Cortex XSOAR, a search is performed. By default, the search is for the last 7 days, but you can adjust the date range.
- Wildcard searches can only be performed in the local Cortex XSOAR threat intel library, and not in Unit 42 Intel data. Example: \*example.com
- Complex searches are only conducted in the local Cortex XSOAR threat intel library, and not in Unit 42 Intel data. Example: type:URL and verdict:Malicious.
- For files, only the SHA256 hash returns Unit 42 Intel data.
- For a query to include Unit 42 Intel results, it must be a lookup for an exact match.

You can search for indicators using any of the available search fields. This is a partial list of the available search fields.

Field	Description
<b>type</b>	The type of the indicator, such as File or Email.
<b>verdict</b>	The reputation of the indicator: <ul style="list-style-type: none"> <li>Malicious</li> <li>Suspicious</li> <li>Benign</li> <li>Unknown</li> </ul>
<b>aggregatedReliability</b>	Searches for indicators based on a reliability score such as A - Completely reliable.
<b>sourceBrands</b>	Indicator feed or enrichment integrations.
<b>sourceInstances</b>	A specific instance of an indicator feed or enrichment integration.
<b>expirationSource</b>	The source (such as script or manual.) that last sets the indicator's expiration status.

Field	Description
<b>tags</b>	Tags applied to indicators.
<b>comments</b>	Search for keywords within indicators' comments.

You can use a wildcard query, which finds indicators containing terms that match the specified wildcard. For example, the \* pattern matches any sequence of 0 or more characters, and ? matches any single character. For a regex query, use the following value:

```
"/.*\?\*/"
```

#### Indicator queries and Unit 42

Unit 42 Intel data is not automatically added to the Cortex XSOAR Threat Intel library. When you query for an indicator on the Threat Intel page, in some cases the indicator is not in the Threat Intel library, but exists in Unit 42 Intel. In other cases, the indicator may already be in the Cortex XSOAR Threat Intel library, but more in-depth information is available from Unit 42 Intel.

When a query is performed in both Cortex XSOAR and Unit 42 Intel, there are four possible results:

The indicator exists in Cortex XSOAR but does not exist in Unit 42 Intel

The Cortex XSOAR search result is displayed in a table. Click on the value to reach the Summary tab. The Summary tab presents information about the indicator stored in Cortex XSOAR. The Unit 42 Intel tab is disabled.

The indicator exists in Unit 42 Intel, but does not exist in the Cortex XSOAR threat intel library

To view the Unit 42 Intel data for this indicator, click on the indicator search term in blue.

From the Unit 42 Intel tab, you have the option to add the indicator to Cortex XSOAR or to add and enrich the indicator to Cortex XSOAR.

- Add to XSOAR

The indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (ingested through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. No third-party enrichments are run on the indicator. We recommend using this option if, for security reasons, you do not want to expose the indicator to any third-party services.

- Add to XSOAR & Enrich

The indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (ingested through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. Your configured third-party enrichments are run on the indicator.

When you add indicators to the Cortex XSOAR threat intel library from Unit 42 Intel, the indicators are available for use in scripts and playbooks.

The indicator exists in Cortex XSOAR and in Unit 42 Intel

The Cortex XSOAR result is displayed in a table. Click on the value to reach the Summary tab. The Summary tab presents information about the indicator stored in Cortex XSOAR. Click on the Unit 42 Intel tab to view Unit 42 data. From the Unit 42 Intel tab, you have the option to do the following:

- Update

Updated Unit 42 Intel for the indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (brought in through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. No third-party enrichments are run on the indicator. We recommend using this option if, for security reasons, you do not want to expose the indicator to any third-party services.

- Update & Enrich

Updated Unit 42 Intel for the indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (brought in through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. Your configured third-party enrichments are run on the indicator.

The indicator does not exist in Cortex XSOAR or in Unit 42 Intel

If the query was for an indicator type that is not an IP address, domain, URL, or SHA256 file hash OR if the query included a wildcard or a complex search, the search was performed on Cortex XSOAR data from the last 7 days. You can extend the date range to see if the indicator is in Cortex XSOAR but is older than 7 days.

A screenshot of a search interface. At the top is a search bar with a magnifying glass icon. Below it is a section labeled "Your search:" followed by a list of two items:

- Did not match any indicator in Local Threat Inventory (Within the query date range)
- Did not match an exact lookup in unit 42 intel (Date range filter was not applied)

## 16.6 | Indicator investigation

### Abstract

Learn how to use TIM in your use case, such as creating a TIM report, accessing and using Unit 42 Intel data, investigating an indicator and creating indicator relationships.

Cortex XSOAR enables you to centralize and manage every aspect of your TIM investigation. Create, extract, and enrich indicators using Unit 42 Intel data and explore their relationships to gain deeper insights.

After you start ingesting indicators into Cortex XSOAR, you can start your investigation, including creating indicators, adding indicators to an incident, extracting indicators, exporting indicators, etc.

Cortex XSOAR Threat Intel includes access to the Unit 42 Intel service, enabling you to identify threats in your network and discover and contextualize trends. Unit 42 Intel provides data from WildFire (Palo Alto Networks' cloud-based malware sandbox), the PAN-DB URL Filtering database, Palo Alto Networks' Unit 42 threat intelligence team, and third-party feeds (including both closed and open-source intelligence). Unit 42 Intel data is continually updated to include the most recent threat samples analyzed by Palo Alto Networks, enabling you to keep up with threat trends and take a proactive approach to securing your network.

When investigating an indicator, you can see the following tabs:

- Summary

View verdict, enrich, expire, delete and exclude the indicator, add relationships, view related incidents, and add comments. Add or remove tags, which can help classify known threats. For example, you may want to group specific malware indicators that are part of ransomware, such as trojan or loader. Unit 42 Intel data also publishes tags to assist your classification.

- Additional Details

Add or view any community notes for sharing and any custom details.

- Unit 42 Intel

If the indicator is available in Unit 42, you can view related Unit 42 Intel data.

If the indicator has been found in the Unit 42 database you can view the following information (and download the Wildfire report (if available), according to indicator type:

Available data according to indicator type

Indicator Type	Layout Sections
IP address	<ul style="list-style-type: none"> <li>◦ Verdict</li> <li>◦ Source</li> <li>◦ Relationships</li> <li>◦ PAN-DB Categorization</li> <li>◦ Passive DNS</li> </ul>
URL	<ul style="list-style-type: none"> <li>◦ Verdict</li> <li>◦ Source</li> <li>◦ Relationships</li> <li>◦ PAN-DB Categorization</li> <li>◦ WHOIS</li> </ul>
Domain	<ul style="list-style-type: none"> <li>◦ Verdict</li> <li>◦ Source</li> <li>◦ Relationships</li> <li>◦ PAN-DB Categorization</li> <li>◦ Passive DNS</li> <li>◦ WHOIS</li> </ul>
File	<ul style="list-style-type: none"> <li>◦ Verdict</li> <li>◦ Source</li> <li>◦ Relationships</li> <li>◦ Summary</li> <li>◦ WildFire Analysis</li> <li>◦ Related Sessions &amp; Submissions</li> </ul>

When investigating an indicator, you can perform actions on the indicator, such as:

Action	Description
Enrich an indicator	You can view detailed information about the indicator (WHOIS information for example), using third-party integrations such as VirusTotal and IPinfo. For more information, see Extract and enrich an indicator.
Expire an indicator	You may want to expire an indicator to filter out less relevant alerts, allowing analysts to focus on active threats. For more information, see Expire an indicator.
Manage indicator relationships	Threat Intel Management in Cortex XSOAR includes a feed that brings in a collection of threat intel objects as indicators. These indicators are stored in the Cortex XSOAR threat intel library and include Malware, Attack Patterns, Campaigns, and Threat Actors. When you add or update an indicator from Unit 42 Intel, a relationship is formed in the database between the relevant threat intel object and the new, or updated, indicator. For more information, see Manage indicator relationships.
Delete and exclude indicators	Indicators added to an exclusion list are disregarded by the system and are not created or involved in automated flows. For more information, see Delete and exclude indicators.

### 16.6.1 | Indicator verdict

#### Abstract

Cortex XSOAR analyzes indicators to determine whether they are malicious. Create indicator types and custom layouts, exclusion lists, and indicator verdicts.

An indicator's verdict is assigned according to the verdict returned by the source with the highest reliability. In cases where multiple sources with the same reliability score return a different verdict for the indicator, the worst verdict is taken. Indicators are assigned the following verdicts:

- 0: Unknown
- 1: Benign
- 2: Suspicious
- 3: Malicious

You can set the verdict manually by editing the indicator. If you manually changed the indicator's verdict and want to recalculate it according to enrichment integrations, set the verdict to Unknown and then enrich the indicator. If after manually setting the indicator's verdict you run indicator enrichment without setting the verdict to Unknown, the indicator is enriched but the manually set verdict is not changed.

#### Source reliability

The reliability of an intelligence data source influences the verdict of an indicator and the values for indicator fields when merging indicators. Indicator fields are merged according to the source reliability hierarchy, which means that when there are two different values for a single indicator field, the field will be populated with the value provided by the source with the highest reliability score.

In rare cases, two sources with the same reliability score might return different values for the same indicator field. In these cases, the field is populated with the most recently provided source, unless the field is verdict. If two sources have the same reliability score and return different values for the verdict field, the worse verdict is used.

For the field types Tags and Multi-select, all values are appended, and nothing is overridden.

Source	Reliability Score	Notes
Manual	A+++	A user manually updates the verdict of an indicator.
Reputation script	A++	A script with the <b>reputation</b> tag calculates the verdict of an indicator. For example, the DataDomainReputation script evaluates the verdict of a URL or domain.

Source	Reliability Score	Notes
Third-party enrichment	A+	An integration or service that evaluates the verdict of an indicator. For example, the urlscan.io integration evaluates the verdict of a URL.
Feed	A: Completely reliable	The feed reliability is applied at the integration instance level.
	B: Usually reliable	
	C: Fairly reliable	
	D: Not usually reliable	
	E: Unreliable	
	F: Reliability cannot be judged	

#### Different verdicts from integrations

In this example, two third-party integrations, VirusTotal and AlienVault, return a different verdict for the same indicator. The indicator's verdict will be Malicious because VirusTotal's reliability score is higher than AlienVault.

Integration	Reliability	Verdict	Final Verdict
VirusTotal	C - Fairly reliable	Malicious	Malicious
AlienVault	D- Not usually reliable	Benign	

In this example, two sources with the same verdict score return a different verdict for the same indicator. The indicator's verdict will be Malicious because when two sources have the same reliability, the worse verdict applies.

Integration	Reliability	Verdict	Final Verdict
TAXII Feed	B - Usually reliable	Malicious	Malicious
CSV Feed	B - Usually reliable	Benign	

#### 16.6.2 | Extract and enrich an indicator

##### Abstract

How to extract and enrich an indicator in Cortex XSOAR.

Indicator extraction identifies indicators from different text sources in the system (such as War Room entries), extracts them, and creates indicators in Cortex XSOAR. After extraction, the indicators are enriched. An administrator can set up indicator extraction automatically in an incident type or a playbook. For more information, see Indicator Extraction.

Indicator enrichment takes the extracted indicator and provides detailed information about the indicator (WHOIS information for example), using third-party integrations such as VirusTotal and IPInfo.

If you want to extract an indicator manually, you can do the following:

- Run indicator extraction in the CLI by running one of the following commands:

Command	Description
extractIndicators	<p>If you want to extract indicators from non-War-Room-entry sources (such as extracting from files), use the !extractIndicators command from the CLI. Use the command to do the following:</p> <ul style="list-style-type: none"> <li>Validate regex: Test a specific string to see if the relevant indicators are extracted correctly, such as a URL.</li> <li>In a playbook or script. The command extracts indicators in a playbook or a script (non War Room source), and also creates and enriches them.</li> </ul> <p>You can extract the following:</p> <ul style="list-style-type: none"> <li>A specified entry (an entry ID)</li> <li>Investigation (Investigation ID)</li> <li>Text</li> <li>File path</li> </ul> <p>For example, type !extractIndicators text="some text 1.1.1.1 something" auto-extract=inline. The entry text contains the text of the indicators, which is extracted and enriched.</p> <p>You can also extract indicators by adding the auto-extract parameter with the script and the mode for which you are setting it up. For example: !ReadFile entryId=826@101 auto-extract=inline.</p> <p>Usually, when using the CLI, you want to disable indicator extraction. For example, if you return internal/private data to the War Room, and you do not want it to be extracted and enriched in third-party services, add auto-extract=none to your CLI command.</p>
enrichIndicators	<p>The enrichIndicators command is usually used when you want to batch enrich indicators. This command works on existing indicators only (it does not create them on its own). When running the command, the relevant enrichment command is triggered (such as !ip), which is based on the indicator type that is found. The data is saved to context and the indicator.</p> <p><b>NOTE:</b></p> <p>Triggering enrichment on a substantial number of indicators can take time (because it's activating all enrichment integrations per indicator) and can result in performance degradation.</p>
Reputation commands	<p>Reputation commands such as !ip, can be run for new indicators and indicators already in the system. If extraction is on, the data is saved both to the indicator and the incident's context. If not, then the data is saved only to the context because the mapping flow is always triggered in enrichment commands. The default configuration is set to none in playbook tasks for extraction.</p> <p><b>NOTE:</b></p> <p>Reputation commands, such as !ip, !domain can only be used when you configure and enable a reputation integration instance, such as VirusTotal and WHOIS.</p>

- Use the Enrich indicator button in the indicator layout. This is the same effect as running a reputation command.

You must have a TIM license to access the indicator layout.

- Run indicator enrichment in the Quick View window

If there is an enhancement script attached to the indicator type, in the indicator Quick View window, you can run a script to enrich an indicator. For example, the Domain indicator type uses the DomainReputation enhancement script. In an incident that contains a domain indicator type, click Quick View. In the Indicators tab, click Domain → Actions → DomainReputation.

You can also run the enhancement script in the CLI.

### 16.6.3 | Expire an indicator

#### Abstract

Expire an indicator in the CLI or in the UI.

Indicators can have the Expiration Status field set to Active or Expired. When indicators expire, they still exist in Cortex XSOAR, meaning they are still displayed and you can still search for them. You may want to expire an indicator to filter out less relevant alerts, allowing analysts to focus on active threats. Expiring IoCs that are no longer relevant helps ensure that security systems remain focused on current threats.

You can set up expiration in the indicator type, integration feed, or in a script. For more information, see Configure indicator expiration. When you manually expire an indicator, this overrides indicator extraction rules set in scripts, indicator types, and feeds.

You can expire indicators using the following methods:

- In the indicator layout by clicking Expire indicator.

You need a TIM license to access the indicator layout.

- Use the `expireIndicators` command to change the expiration status to Expired for one or more indicators. This command accepts a comma-separated list of indicator values and supports multiple indicator types. For example, you can set the expiration status for an IP address, domain, and file hash: `!expireIndicators value=1.1.1.1,safeurl.com,45356A9DB614ED7161A3B9192E2F318D0AB5AD10`.
- Use the `!setIndicator` or for multiple indicators use the `!setIndicators` command to reset the indicators' expiration value. The value can also be set to `Never`, so that the indicators never expire. For example, `!setIndicators indicatorsValues=watson.com expiration=Never`.

### 16.6.4 | Manage indicator relationships

#### Abstract

How to use and create indicator relationships in Cortex XSOAR and how it benefits an investigation.

Indicator relationships are connections between different indicators. These relationships can be IP addresses related to one another, domains impersonating legitimate domains, etc. These relationships enable you to enhance investigations with information about indicators and how they might be connected to other incidents or indicators. For example, if you have a phishing incident with several indicators, one of those indicators might lead to another indicator, which is a malicious threat actor. Once you know the threat actor, you can investigate to see the incidents it was involved in, its known TTPs (tactics, techniques, and procedures), and other indicators that might be related to the threat actor. The initial incident which started as a phishing investigation immediately becomes a true positive and relates to a specific malicious entity.

Relationships are created from threat intel feeds and enrichment integrations that support the automatic creation of relationships, such as AlienVault OTX v2 and URLhaus, by selecting Create relationships in the integration settings. Based on the information that exists in the integrations, the relationships are formed.

You can view indicator relationships by clicking on the indicator from an incident, and then from the Quick View window click the Relationships tab.

The Threat Intel Management system in Cortex XSOAR includes a feed that brings in a collection of threat intel objects as indicators. These indicators are stored in the Cortex XSOAR threat intel library and include Malware, Attack Patterns, Campaigns, and Threat Actors. When you add or update an indicator from Unit 42 Intel, a relationship is formed in the database between the relevant threat intel object and the new, or updated, indicator.

#### Create indicator relationships

You can also manually create and modify relationships, which is useful when a specific threat report comes out. For example, Unit 42's SolarStorm report contains indicators and relationships that might not exist in your system, or you might not be aware of their connection.

If a relationship is no longer relevant, you can revoke it. This might be relevant, for example, if a known malicious domain is no longer associated with a specific IP address.

#### NOTE:

To create and modify indicator relationships, you must have the TIM license.

When you create a relationship, you can set the relationship type such as whether the indicator is related, attached, applied, etc. For example, a file is attached-to an email. The email communicated-with the file.

You can create relationships by adding them in a playbook, in the CLI using the `CreateIndicatorRelationship` command, or when investigating an indicator in the Threat Intel tab.

#### How to add an indicator relationship from an Indicator

1. Open an indicator and in the RELATIONSHIPS section add a relationship.
2. In the New Relationships window, in Step 1, add a query by which to search for the relevant indicators.

You can optionally limit the time range for the search.

3. Select the indicators you want to create a relationship to.

4. In Step 2 set the relationship type.

By default, the relationship is related-to. For example, IP address x.x.x.x is related-to IP address y.y.y.y.

5. Save the relationship.

#### NOTE:

You can also add an indicator relationship from the Quick View when selecting an indicator from an incident.

Investigate an indicator using indicator relationships

In this example, you can see how to use the relationships feature to further your investigation.

1. When opening the incident, although you can see that the severity is low, the incident has two indicators.

The screenshot shows the 'Incident Info' tab selected. Under 'CASE DETAILS', the severity is listed as 'Low'. Under 'INDICATORS (2)', there are two entries:

Type	Value	Verdict	First Seen	Last Seen	Source Time Stamp	Related Incidents	Feed
File	867670c365b5132b8c15ffec18ea609c	Unknown	Aug 15H 2022 11:32:43	Aug 15H 2022 11:32:43	Aug 15H 2022 11:32:43	1	Unit 42 Intel
IP	122.200.106.44	Unknown	Aug 15H 2022 11:32:43	Aug 15H 2022 11:32:43	Aug 15H 2022 11:32:43	1	Unit 42 Intel

2. When you click the file hash indicator, neither the Info nor Relationships tabs have any additional details. This seems to indicate that the file is harmless.

The screenshot shows the 'Info' tab selected for the file hash indicator. The 'Relationships' tab is also visible. The indicator details include:

- File: MD5
- Value: 867670c365b5132b8c15ffec18ea609c
- Verdict: Unknown
- First Seen: Aug 15H 2022 11:32:43
- Last Seen: Aug 15H 2022 11:32:43
- Source Time Stamp: Aug 15H 2022 11:32:43
- Related Incidents: 1
- Feed: Unit 42 Intel

3. Click on the IP address indicator.

Under the Info tab, you can see that the indicator was ingested from a threat intel feed. This already bears further investigation.

The screenshot shows the 'Info' tab selected for the IP indicator. The 'Relationships' tab is also visible. The indicator details include:

- Set by Indicator Type IP on May 25, 2021 at 1:12 PM
- Reputation: Unknown
- Source: Set by @CBot
- Verdict: Unknown
- Source Time Stamp: May 25, 2021, 1:12 PM

4. Go to the Relationships tab.

You can see that this indicator is related to a campaign.

The screenshot shows the 'Relationships' tab selected for the IP indicator. The 'Info' tab is also visible. The relationships table shows one entry:

Relationship	Related Indicator	Indicator Type	Modified
Indicated-by	Campaign 1 - Hangover BackConfig	Campaign	May 25, 2021, 1

What started as a low severity incident, has become a lot more threatening.

## 16.6.5 | Delete and exclude indicators

### Abstract

Indicators added to an exclusion list are disregarded by the system. Add indicators to an exclusion list in Cortex XSOAR.

Indicators added to an exclusion list are disregarded by the system and are not created or involved in automated flows such as indicator extraction. You can still manually enrich IP addresses and URLs that are on the exclusion list, but the results are not posted to the War Room.

Add indicators to the exclusion list either in the Indicators table or in the Exclusion List page.

#### Delete and exclude indicators in the Indicators table

Select one or more indicators from the Indicators table and click the Delete and Exclude button. The indicators are deleted from the Indicators table and added to the exclusion list. You can associate these indicators with one or more indicator types.

If you delete the indicator it is removed from Cortex XSOAR. This option should be used mainly for correcting errors in ingestion, and not as part of your regular workflow.

#### Add indicators in the Exclusion List page

From the Exclusion List page, you can view the list of excluded indicators, add an indicator to the exclusion list, or define indicator values to be excluded using a regular expression (regex) or CIDR.

1. Select Settings & Info → Settings → Object Setup → Indicators → Exclusion List → New excluded indicator.
2. Add the indicator value. For example, example.com (for a domain).

#### **CAUTION:**

Ensure you are using the correct syntax when defining the values for your exclusion lists.

3. Select whether to use Regex.

A regular expression enables you to identify a sequence of characters in an unknown string. The following example would identify www.demisto.com:  
`[A-Za-z0-9!@#$%\&]*demisto[A-Za-z0-9!@#$%\&]*`

Classless inter-domain routing (CIDR) enables you to define a range of IP addresses. For example, the IPv4 block 192.168.100.0/22 represents the 1024 IPv4 addresses from 192.168.100.0 to 192.168.103.255.

4. Add a reason as to why you are excluding the indicator.
5. Add the indicator types that apply.
6. Save the excluded indicator.

#### Exclusion list examples

Exclusion	Description	Settings
Domain, URLs, and subdomains	Excludes a specific domain, and all subdomains and URLs associated with the domain.	<p>Define two entries to cover all URLs and subdomains associated with a specific domain.</p> <p>Entry one:</p> <ul style="list-style-type: none"> <li>• Value: Subdomains and URLs. Example: <code>\.example\..com</code></li> <li>• Select Use Regex.</li> <li>• Do not select any indicator types.</li> </ul> <p>Entry two:</p> <ul style="list-style-type: none"> <li>• Value: The specific domain. Example: <code>example.com</code></li> <li>• <b>Do NOT</b> select Use Regex.</li> <li>• Do not select any indicator types.</li> </ul>

Exclusion	Description	Settings
Subdomain (and URLs) specifically	Excludes any subdomains and URLs of a domain, but the domain is still extracted.	<ul style="list-style-type: none"> <li>Value: Subdomains and URLs. Example: <code>\.example\..com</code></li> <li>Select Use Regex.</li> <li>Do not select any indicator types.</li> </ul>
Specific domain only	Excludes a specific domain. Subdomains and URLs are still extracted.	<ul style="list-style-type: none"> <li>Value: The specific domain. Example: <code>example.com</code></li> <li><b>Do NOT</b> select Use Regex.</li> <li>Select indicator type: Domain.</li> </ul>
URL with wildcards	Excludes any indicators of type URL matching the regex. Indicators <code>example.com</code> and <code>examplesub.example.com</code> of type Domain would still be extracted. Start the regex with <code>https://</code> to exclude both HTTP and HTTPS URLs.	<ul style="list-style-type: none"> <li>Value: The URL with wildcard added at the end. Example: <code>http://examplesub.example.com</code></li> <li>Select Use Regex.</li> <li>Select indicator type: URL.</li> </ul>
Specific URL	Excludes a specific URL, but the domain and subdomains are still extracted.	<ul style="list-style-type: none"> <li>Value: The specific URL. Example: <code>http://examplesub.example.com/myexample</code></li> <li><b>Do NOT</b> select Use Regex.</li> <li>Select indicator type: URL.</li> </ul>
URLs, domain, and subdomains, case-insensitive, anchored to start	Excludes domain <code>example.com</code> , its subdomains, and its URLs. Case-insensitive. Anchors regex match to the start of the indicator value, so indicators that contain but do not start with a match (e.g., <code>example.net?param=example.com</code> ) are not excluded.	<ul style="list-style-type: none"> <li>Value: Domain, subdomains and URLs, case insensitive and anchored to the start of the indicator. Example: <code>(?i)^(\https?:\/\/)?(([a-zA-Z0-9\-\_]+\.)+?)example\..com</code></li> <li>Select Use Regex.</li> <li>Select indicator types: URL, Domain.</li> </ul>
All URLs	Excludes all URLs for a specific domain that have a path (even an empty path), but the domain and subdomains are still extracted.	<ul style="list-style-type: none"> <li>Value: URLs with or without a path. Example: <code>example\..com/</code></li> <li>Select Use Regex.</li> <li>Do not select any indicator types.</li> </ul>

## 16.6.6 | Investigate files using sample analysis

### Abstract

View static and dynamic analysis of file samples to identify malware, investigate trends, and create reports.

Unit 42 Intel's Sample Analysis tools enable you to conduct in-depth investigations and analyses of file samples. If the file indicator is found in the Unit 42 Intel service, you have access to a full report on activities, properties, and behaviors associated with the file. File samples are run and analyzed using Palo Alto Networks' WildFire cloud-based threat analysis service, so you can view dynamic analysis of observed behavior, static analysis of the file contents, and related sessions and submissions. For example, when investigating a malicious file found in your network, you want to understand what the file did locally and in the network.

You can search for file samples, either in the Indicators tab or the Sample Analysis tab. If using the Sample Analysis tab, you can search for the following samples:

- Public Samples

Searches for samples that have been submitted by firewalls or sample sources other than those associated with your CSP account.

- My Samples

The My Samples option is only available for users with a Palo Alto Networks Firewall, WildFire, Cortex XDR, Prisma SaaS, or Prisma Access license. It takes data from devices in the same CSP account where your tenant is registered. My Samples data is not available for multi-tenant deployments.

- All Samples

Searches for both public and your samples.

**NOTE:**

When searching on the Sample Analysis page for relationships -relationships "", some results may appear without their specific relationships listed, due to internal relationship permissions.

In the Sample Analysis tab, you can search for samples based on the sample hash and it compares all historical and new samples to the search conditions and filters the search results accordingly.

**Investigate a file sample**

In the Sample Analysis tab, locate a file you want to investigate and click the SHA256 section to start the investigation.

In the Unit 42 Intel tab, you can see the following sections:

Section	Description
General Section	<p>In the top half of the page, you can see the Verdict, a summary of the file, when it was first and last seen by Wildfire, and any relationships.</p> <p>You can download a WildFire report in PDF format, which includes information such as File Information, Static Analysis, and Dynamic Analysis.</p>
WildFire Dynamic Analysis - Observed Behavior	<p>A high-level overview of the behavior observed when the file was run in the WildFire sandbox. Examples might include potentially malicious behaviors such as connecting to a potentially vulnerable port or creating an executable file in the Windows folder, as well as behaviors frequently performed by legitimate software, such as scheduling a task in Windows Task Scheduler.</p>
WildFire Dynamic Analysis - Sections	<p>Dynamic analysis provides a granular view of file activity, process activity, registry activity, connection activity, etc. Files run in a custom-built, evasion-resistant virtual environment in which previously unknown submissions are detonated to determine real-world effects and behavior. Behavior can be observed in one or more operating system environments. It is broken down into the machines it was simulated and the activity itself. For example, Process Activity lists files that started a parent process, the process name, the action the process performed, and whether they are malicious, suspicious, etc. It shows not only the observed behavior of the file sample, but also how many times the behavior was observed in other Unit 42 samples (malicious samples, suspicious samples, and unknown samples).</p> <p>In the following example, you can see that the parent process <code>sample.exe</code> wrote to file <code>kernel32=E02A3B57EA8B393408FF782866A1D342DD8C6B5F5925BA527981DBB21B6A4080</code>. The same behavior occurred in 3.57m samples that had a verdict of malicious.</p>
WildFire Static Analysis	<p>The WildFire Static analysis detects known threats by analyzing the characteristics of a sample before execution in the WildFire sandbox. Static analysis can provide instant identification of malware variants and includes dynamic unpacking to analyze threats attempting to evade detection using packer tools. You can analyze files such as Portable Executable (PE) files and any suspicious files.</p>

Section	Description
Related Sessions & Submissions	Shows any related sessions and submissions where the file was seen in your Firewall. Related sessions and submissions data are available if you have one of the following products: Palo Alto Networks Firewall, WildFire, Cortex XDR, Prisma SaaS, or Prisma Access.

You have the option to add the file sample (without enriching) to Cortex XSOAR or to add and enrich the indicator to Cortex XSOAR.

- Add to XSOAR

The indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (ingested through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. No third-party enrichments are run on the indicator. We recommend using this option if, for security reasons, you do not want to expose the indicator to any third-party services.

- Add to XSOAR & Enrich

The indicator is added to Cortex XSOAR. If the indicator is related to one or more Unit 42 threat intel objects already in Cortex XSOAR (ingested through the Unit 42 Feed integration), relationships are created in the database between the Unit 42 threat intel objects and the file indicator. Your configured third-party enrichments are run on the indicator.

When you add indicators to the Cortex XSOAR threat intel library from Unit 42 Intel, the indicators are available for use in scripts and playbooks.

#### Sample Analysis Advanced Search

You can use Unit 42 Intel data to build complex searches for file samples with similar characteristics. In some sections, you can search for specific characteristics. For example, in the WILDFIRE DYNAMIC ANALYSIS - OBSERVED BEHAVIOR, section you can add Behavior to a search. In the WILDFIRE DYNAMIC ANALYSIS - SECTIONS, you can add PARENT PROCESS, ACTION, or **PARAMETERS** or all characteristics of the file activity to a search.

To investigate further you can build a new search that contains this specific behavior and view the relevant samples. When selecting the relevant column, you can do the following:

- **Add to Sample Analysis Search**

Adds selected information from a column to a Sample Analysis search (in the WILDFIRE DYNAMIC ANALYSIS - SECTIONS, you can add a whole row to the search).

- **Create New Sample Analysis Search**

Clears any search characteristics you have already added and starts a new Sample Analysis search with the selected characteristics.

After selecting the relevant option, a message appears. You can do the following:

- Run the query now, by clicking the link.

You pivot to the Sample Analysis tab where you can edit or run your search for samples that exhibited the same behavior.

- If you want to add additional items to the search, ignore the message.

To run the search, go to the Threat Intel page and click the Sample Analysis tab.

You can save the search query for future use.

In the following example, you have an incident with an extracted file indicator. The Unit 42 Intel tab shows the file's behavior. You scroll through the sample's behavior and see a suspicious behavior: Powershell1.exe written to a file in the Administrator's User folder, named 443.exe. You want to find other samples with the same behavior and determine if they are related to a known adversary or malware, so you add that specific behavior to your search.

#### 16.6.7 | Use sessions and submissions in your investigation

##### Abstract

Use firewall sessions and submissions to products such as Prisma Cloud, and Prisma Access with Cortex XSOAR, to find threats and protect your network.

The Sessions & Submissions tab enables you to use your firewall sessions and submissions data for investigation and analysis.

Sessions refer to firewall sessions that show connections from one endpoint to another. A firewall can forward information about network sessions for an investigation. Cortex XSOAR TIM uses session information to learn more about the context of the suspicious network event, indicators of compromise related to the malware, affected hosts and clients, and applications used to deliver the malware.

Submissions refer to sample logs reported to Wildfire from Palo Alto Networks products, such as Cortex XDR. While Sessions data shows connections from one endpoint to another, submissions data shows if a file was found on a specific endpoint.

Sessions & Submissions data is available for users with at least one of the following products:

- Palo Alto Networks Firewall
- WildFire
- Cortex XDR
- Prisma Cloud
- Prisma Access

You can take steps to block external IP addresses that are the sources of malicious files and threat campaigns. You can find compromised machines within your network, isolate them as needed, and take remediation steps. For example, search for a file hash in Sessions & Submissions. If the file appeared in one or more sessions or submissions, you can see when and where that occurred. A firewall session data enables you to view the source IP and the destination IP for each session that includes the file.

If you are using Cortex XDR, you can see which XDR agent reported the file and which computers are affected.

#### NOTE:

When searching on the Sessions & Submissions page for relationships -relationships "", some results may appear without their specific relationships listed, due to internal relationship permissions.

(Multi-tenant) Sessions & Submissions data is not available for Multi-tenant deployments.

#### Investigate Sessions and Submissions

From Sessions & Submissions in the ID column click an ID to start an investigation.

In the Session Summary tab you can see the following information:

Section	Description
Basic Information	Includes general information such as the session Timestamp, destination IP, and source country.
Sample Information	Includes file information, such as the file name, SHA, File URL, and Status. The Status for blocked samples is Blocked, while the status for allowed samples is blank.
NOTE:	The Application is matched to the type of application traffic detected in a session. For example, a search for the Application web-browsing returns sessions during which web browsing over HTTP occurred. See Applipedia for an updated list of applications that Palo Alto Networks identifies.
Metadata	Includes metadata, such as the source, region, and Device Hostname.
Related Sessions and Submissions	Lists any related sessions and submissions for further investigation

#### Sessions & Submissions advanced search

You can use Unit 42 Intel data to build complex searches for sessions and submissions with similar characteristics. From within the Session Summary tab, any of the items listed in the Basic Information, Sample Information, or Metadata sections can be used to create a new search for similar sessions and submissions. For example, you can create a new search that includes a specific destination IP and a specific file name that you found together in a session.

To build a new search, hover your cursor over the end of the desired row. You can submit the following search:

Basic Information	
Session Timestamp	December 1, 2021 1:51 PM
Destination IP	23.103.140.138
Destination Port	25
Destination Country	India
Source IP	144.0.11.253
Source Port	4937
Source Country	China

- **Add to Sessions & Submissions Search**

Adds selected information to a Sessions & Submissions search.

- **Create New Sessions & Submissions Search**

Clears any search characteristics you have already added and starts a new Sessions & Submissions search.

After selecting the relevant option, a message appears. You can do the following:

- Run the query now, by clicking the link.

You pivot to the Sessions & Submissions tab where you can edit or run your search for sessions and submissions that exhibited the same behavior.

- If you want to add additional items to the search, ignore the message.

To run the search without clicking on the popup link, go to the Threat Intel page and click on the Sessions & Submissions tab.

## 16.7 | Manage Threat Intel Reports

### Abstract

An overview of working with threat intel reports in Cortex XSOAR.

Threat Intel Reports gives you the ability to create, review, publish, and generate threat intelligence reports.

Threat intel reports summarize and share threat intelligence research conducted within your organization by threat analysts and threat hunters. Threat intelligence reports help you communicate the current threat landscape to internal and external stakeholders, whether in the form of high-level summary reports for C-level executives, or detailed, tactical reports for the SOC and other security stakeholders.

#### NOTE:

If users are unable to see the Threat Intel page, ensure that users have access, by verifying that their user role is assigned the Threat Intel permission (Page Access).

The Threat Intel Reports page shows all the types of reports created. You can do the following:

- Create a report

After you create a report, edit the report as required. The core of the report is the Overview/Summary section, which is used to enter freeform text. By default, users with Administrator or Analyst roles have read/write access to the reports. When creating a report, you can restrict the report to specific user roles. When you finish a section, select the checkmark to save. If you navigate away and return to the Threat Intel Reports page, the report appears in the Threat Intel Reports table. Select the report to continue working on it. When finished, you can send it for review, publish it, and generate a PDF version. When published, it creates a read-only version of the report for you to share.

- Edit a report

You can edit the report when you create the report or from the Threat Intel Reports table (if you navigate away and return to the Threat Intel Reports page).

- Delete a report

### Rule-based Access Control

By default, all roles have read/write access to the reports. To grant read and read/write access only to specific roles, you can define access to reports by doing one of the following:

- When you create a report, choose one or more roles in the Permissions section of the new report dialog.
- After you create a report, choose one or more roles in the Access section of the report layout.

If a role has not been added to either the Access or Permissions section, the role does not have read and read/write access to the Threat Intel report.

### Create a Threat Intel report

You can create a threat intel report by choosing a type and defining other basic report information. To customize the threat Intel report such as creating new types and layouts, see Customize Threat Intel Reports.

When you create a report, Cortex XSOAR creates a blank report based on the type you choose. Once created, edit the report to populate it with relevant content before generating or sharing a report.

In the Overview/Summary section, enter freeform text using the Markdown editor, which enables you to apply formatting options to the body text, including text sizing, coloring, formatting, pictures/icons, logo, and section headers.

1. Select Threat Intel → Threat Intel Reports → New Threat Intel Report

2. Enter a Name and configure any other relevant fields.

You can edit any fields after you create the report.

3. Create new Threat Intel Report.

The report is automatically in draft report status.

4. Edit report fields as needed and add any information about the specific report.

5. In the Overview/Summary section, to use the Markdown editor, click M.

When finished, select Preview and then save.

6. (Optional) Change the status as required.

For example, if you want to send to another user to check before you publish, select Review.

7. Publish and generate a report.

#### Publish a Threat Intel report

When you have finished drafting a report, you can publish the report, which means all user roles have read-only access to the report to prevent other users from making changes. If you unpublish a report, that access is reverted. Publish/unpublish does not revert any read/write access that you granted to a specific role.

1. Navigate to the Threat Intel Reports table and click the Name of the report you want to share.

2. From the Access section, Publish the report.

Once published, anyone you give the report link can see the report (provided they have access to your Cortex XSOAR tenant). To remove read-only access, unpublish the report.

#### Generate a Threat Intel report

If you want to send the report to a larger audience other than Cortex XSOAR users, you can generate a report in PDF. Before generating the report you can save the report as a template, so you don't need to define the settings again. To see a TIM report use case, go to Threat Intel Management use cases.

1. Navigate to the Threat Intel Reports table and click the Name of the report you want to generate.

2. Click the vertical ellipsis icon at the top right of the report and click Report.

3. (Optional) If you want to generate a report from a specific tab, Select a tab to generate report from and then select the relevant tab..

4. From the Properties section, choose a Format, Orientation, and Paper Size for the report.

5. (Optional) Save the report as a template.

6. Generate the report.

## 17 | Troubleshoot

### Abstract

Troubleshoot errors in Cortex XSOAR On-prem.

Troubleshoot errors, view, and take action on the System Diagnostics page. View integration and management audit logs and configure management audit notification forwarding., view integration and management audit logs, set up a Syslog Server, and configure management audit notification forwarding.

### 17.1 | System diagnostics

#### Abstract

View errors and take action on the System Diagnostics page for Cortex XSOAR On-prem.

The System Diagnostics page enables you to identify and fix potential issues before they become system-critical. By default, the System Diagnostics page shows trends from the last 24 hours, but you can also select the last hour, 6 hours, 12 hours, 3 days, or 7 days.

#### NOTE:

Only administrators can view the system diagnostics page.

## Download log bundles

To help with debugging issues, you can download the log bundle by clicking  in the upper right hand corner. The log bundle contains information about the system from the current state up to the past ten days, and it should be included when opening a support ticket.

## Nodes

Four widgets present information regarding nodes.

Node	Description
Nodes - CPU	<p>Trend graph showing CPU consumption. The trend graph shows an increase as system usage increases. Temporary peaks might correlate with system delays or slowness.</p> <p>We recommend increasing CPU resources when you reach system limits.</p>
Nodes - Memory	<p>Trend graph showing memory consumption. The trend graph shows an increase as memory usage increases. Temporary peaks might correlate with system delays or slowness.</p> <p>We recommend increasing memory resources when you reach system limits.</p>
Nodes - Storage	<p>Trend graph showing storage usage. The trend graph shows an increase as storage usage increases. Temporary peaks might correlate with system delays or slowness.</p> <p>We recommend increasing storage resources when you reach system limits.</p>
Active Nodes Snapshot	Shows a list of all active nodes and their status - Connected or Disconnected.

## Storage Groups

Storage Groups display a graph illustrating storage group utilization. The trend graph shows an increase as storage usage grows. A rapid surge in storage utilization might indicate a change in system usage.

We recommend increasing storage capacity or performing a data cleanup when utilization reaches 80%.

## Playbooks in Queue

The Playbooks in Queue widget shows a graph that includes manually and automatically triggered playbooks and displays how many playbooks were waiting in the queue over the displayed period. The playbook queues are designed to manage playbook executions efficiently and prevent system overload. A rapid surge in the graph values might indicate a temporary peak of triggered playbooks and cause playbooks to take longer to execute and may slow UI performance.

If the queue count is constantly higher than 0, contact Customer Support to discuss scaling options.

## Cortex Connectivity Snapshot

The Connectivity Snapshot shows the connection status between your Cortex XSOAR tenant and the external gateway. If the status is Disconnected you cannot upgrade Cortex XSOAR, access the Marketplace, or update Docker images.

## Components Snapshot

The Components Snapshot shows the status of a Cortex XSOAR component.

Status	Action
Healthy	None

Status	Action
Warning	<ul style="list-style-type: none"> <li>Check cluster health graphs for temporary peaks or high resources utilization.</li> <li>Check storage utilization graphs.</li> </ul>
Error	<ul style="list-style-type: none"> <li>If you have recently made changes to your system, verify if these changes might have impacted system components.</li> <li>Open a support case if you cannot find the source of the issue.</li> </ul>

**NOTE:**

For some components, such as storage, if the system reaches a critical level, Cortex XSOAR will no longer function, and you will not be able to access the System Diagnostics page.

We recommend monitoring system components on an ongoing basis to avoid critical-level issues.

The components include:

Component	Description
API	The API request handlers
Storage	System storage and files
Databases	System databases
Telemetry	System telemetry collection
Automation layer	Automation resources and components handler
Playbook Engines	Task queue and priority handling
System Scheduler	System scheduled tasks and prioritization handlers
External Gateways	External resource and connection handling
System Orchestrators	System initialization
Execution Environments	Task execution

## 17.2 | View service limit errors and warnings in the Guard Rails page

### Abstract

Use the Cortex XSOAR Guard Rails page to see details about service limit errors or warnings.

The Cortex XSOAR Guard Rails page provides a list of usage limitation errors and warnings that occur during incident ingestion, investigation, and response. It helps to keep your environment stable and prevent actions that can cause major performance degradation or instability.

Cortex XSOAR has service rate limits for the number of incidents and indicators that can be ingested and stored. The Guard Rails page indicates when incident or indicator size exceeds predefined service limits and may affect performance.

### Cortex XSOAR service rate limits for incident and indicators

Cortex XSOAR supports one or more tenants per customer: One for production, and one or more for development. The development tenant allows you to develop and test components (such as playbooks, automation scripts, and screen layouts) before they are deployed to production.

Indicator volume support differs between customers who own a TIM license and those who do not own a TIM license.

#### Production tenant service limits

Feature	Without A TIM License	With A TIM License
Incidents per day	10,000 Rate limit of 100 incidents ingested per minute	10,000 Rate limit of 100 incidents ingested per minute
Total indicators stored	3,000,000	100,000,000

#### Development tenant service limits

Feature	Without A TIM License	With A TIM License
Incidents per day	2000 Rate limit of 100 incidents ingested per minute	5000 Rate limit of 100 incidents ingested per minute
Total indicators stored	500,000	10,000,000

The development tenant has different technical specifications and should not be used for a production environment or stress testing.

#### NOTE:

For multi-tenant deployments, the same service limits apply to each child tenant.

#### Cortex XSOAR Guard Rails page

The Cortex XSOAR Guard Rails page displays a table with a list of service limit errors and warnings and their details.

An error occurs when a service limit is exceeded. For example, an error can be generated for exceeding the size limit of an attachment or for exceeding the number of entries per incident.

A warning occurs when approaching the service limit. For example, a warning can be generated when the number of entries per incident is approaching the service limit or the number of linked incidents is approaching the service limit.

The service limits are defined out-of-the-box. Contact Cortex XSOAR support if you need to change the values for your service limits.

Access the Guard Rails page from Cortex XSOAR Settings & Info → Settings → System.

The table shows the following information:

- ID: (by default hidden) The log number.
- Timestamp: The date time the error or warning occurred.
- Type: The object type the error or warning occurred on, for example incident or indicator.
- Subtype: The object sub type (N/A if it doesn't exist), for example entries or attachments.
- Severity: Whether the item is an error or a warning.
- Object ID: The ID of the restricted object.
- Count: The number of times a specific item occurred in the last calendar day.
- Description: A short description of the error or warning.

#### NOTE:

Identical messages generated within the same day are not duplicated in the table, only the Count is updated and the Timestamp shows the date time the error or warning occurred for the first time. A count greater than one indicates an identical error or warning occurred more than once within the same day.

## 17.3 | Logs

### Abstract

Logs are a valuable tool in troubleshooting issues that might arise in your Cortex XSOAR environment.

Logs provide information about events that occur in the system. These logs are a valuable tool in troubleshooting issues that might arise in your Cortex XSOAR environment.

## 17.4 | Integration logs

### Abstract

View and export integration logs in Cortex XSOAR. Integration logs record integration details in Cortex XSOAR for troubleshooting.

The Integration Logs table helps monitor, troubleshoot, and analyze performance. It provides visibility into interactions between Cortex XSOAR and external systems, facilitating effective integration management and ensuring the operational integrity of security operations.

Logs are generated for integrations developed with Python or non-python code, for example, JavaScript or PS.

To view integration logs, navigate to Settings & Info → Settings → Integrations → Integration Logs.

The integration log table displays log details sorted by date.

### NOTE:

All Integration logs are located in the integration logs table in your Cortex XSOAR tenant. If you have an engine, verbose logs (including integration logs) are also stored in the log file on your engine machine.

You can view the following data:

Field	Description
Timestamp	The date and time that the integration created the log.
Engine name	The engine server name that the integration was running on.
Log level	The log level (Debug, Error, or Info).
Command	The command that the integration ran.
Source brand	The integration name.
Source instance	The integration instance name that was set in the integration instance settings.
Bundle ID	A unique identifier for logs generated from a specific integration execution. For example, a bundle ID is assigned for all logs generated from a specific integration command or fetch incidents.
Message	A text message indicating the integration status or error.

You can do the following:

- Update the log table

To update the log table, click the Refresh button.

- Filter by field

You can filter by log table column, such as log level, brand, or instance name, and you can save filters for later use. You can also adjust the width of the columns and add or remove columns.

- Export logs

To export the integration log as a **.tsv** file, click the Export to file button.

## 17.5 | Management audit logs

### Abstract

View, export, extract, and purge the audit trail in Cortex XSOAR. The audit trail logs all administrative user actions in Cortex XSOAR.

The management audit logs display a log of all administrative user interactions within Cortex XSOAR. By default, the logs are sorted by Timestamp and cover which users interacted in what way with system objects, and associated data.

#### NOTE:

The audit logs do not include actions performed in the War Room. These actions are documented in the War Room.

You can filter by field, such as email, ID, user name, type, etc., and you can save filters for later use. In addition, you can adjust the appearance of the columns and add or remove columns.

To view the audit logs, go to Settings & Info → Management Audit Logs.

To export the management audit logs as a **.tsv** text-based file, click the Export to file button. You can also forward management audit notifications to a syslog server or an email distribution list.

The following table describes the log types and sub types.

### API Keys

Includes the following subtypes:

#### Subtypes

- Add New Key
- Edit Key
- Delete Key

### Authentication

Includes the following subtypes:

#### Subtypes

- Login
- Logout

### Licensing

Includes details about the license such as expiration and ingestion violation.

### Permissions

Includes user role permissions such as:

#### User role permissions

- Role created
- User permissions assigned
- User Group created
- Role deleted
- User permissions revoked

## Cortex Automation

### Types and subtypes

Type	Sub Type
Classifier	Incident and indicator classifier subtypes, such as add, copy, and edit subtypes.
Content	Includes content bundle subtypes, such as install and download.
ContentPack	Includes content pack subtypes, such as install, and delete.
ContributionPack	Includes contribution content pack subtypes, such as add, edit, and delete.
Credentials	Includes integration credential subtypes, such as add, edit, and delete.
Dashboard	Includes dashboard subtypes such as add, edit, and delete.
Engine	Includes engine subtypes such as add, edit, and delete.
Entry	Includes the following subtypes in an incident investigation: <ul style="list-style-type: none"> <li>• Delete</li> <li>• RemoveEntryPermanently</li> <li>• Edit</li> </ul>
HyperProcess	Includes the add and delete subtypes for the Hyper Process.
Incident	Includes incident subtypes, such as add, edit, close, execute, and duplicate.
IncidentField	Includes incident field subtypes, such as add, edit, delete, and export
Incident Layout	Includes incident layout subtypes, such as add, duplicate, edit, attach, and detach.
IncidentType	Includes incident type subtypes, such as attach, detach, enabled, disabled, and delete.
Indicator	Includes indicator subtypes, such as edit, add, and delete.
IndicatorBulkEdit	Includes the indicator bulk edit subtype, such as edit.

Type	Sub Type
Integration permissions	Includes the indicator permissions edit subtype.
Integrations	Includes integration subtypes, such as add, edit, and delete.
IntegrationsConfig	Includes integration configuration subtypes, such as add, edit, and upload.
Investigation	Includes investigation subtypes, such as add, edit, and reopen.
Jobs	Includes job subtypes, such as add, edit, delete, pause, and abort.
Layout	Includes indicator layout subtypes, such as copy, detach, edit, attach, and detach.
List	Includes list subtypes such as add, edit, and delete.
Playbook	Includes playbook subtypes such as add, edit, upload, and delete.
PreprocessRule	Includes pre-processing rule subtypes such as add and edit.
Script	Includes script subtypes such as copy, upload, edit, and delete.
ServerConfiguration	Includes the server configuration edit subtype.
ThreatIntelReport	Includes the Threat Intel Report subtypes such as create, edit, and delete.
Whitelist	Includes the whitelist subtypes such as delete, batchcreate, and add.
Widget	Includes the widget subtypes such as edit add and reset.

## XSOAR Migration

Includes audit information about the migration from Cortex XSOAR 6 to 8, such as whether users were migrated, the cutoff date, and whether content and integrations were resynced.

## 17.6 | Configure management audit notification forwarding

### Abstract

Send Management Audit logs to an email distribution list.

You can forward management audit notifications to an email distribution list.

By default, all management audit notifications are forwarded.

1. Navigate to Settings & Info → Settings → System → Audit Notifications → Add Forwarding Configuration.
2. Enter a name and a description for the configuration and click Next.
3. Define the Management Audit log scope.

To select a subset of the management audit notifications, click the filter button, select the relevant filters, and perform a search. For example, if you want to forward only notifications related to API keys, click the filter button,, select Type and then select the Api Key value.

4. Click Next.

5. Update the following fields:

Field	Description	Mandatory
Distribution List	Add at least one email address to receive management audit notifications.	Yes
Notification Timezone	Change the notification timezone. The notification timezone only affects the time listed in email notifications. You can use the timezone configured in Cortex XSOAR or select Coordinated Universal Time (UTC).	No
Grouping Timeframe	Change the grouping time frame. The grouping time frame specifies how often Cortex XSOAR sends notifications. Every 30 notifications aggregated within this time frame are sent together. To send every notification as soon as it is generated, set the time frame to 0.  By default, the grouping time frame is 10 minutes.	No
Subject	Select to generate the subject automatically or deselect and enter the email subject. By default, this field is selected.	Optional

6. Click Done to send the notification.

## 18 | Reference

### Abstract

Includes reference topics, such as a list of server configurations, and user details and preferences for Cortex XSOAR.

This section provides comprehensive reference information, empowering you to manage Cortex XSOAR effectively. Access essential details like server configurations and explore powerful in-app search functionalities to quickly find the information you need.

### 18.1 | Cortex XSOAR concepts

#### Abstract

Common concepts in Cortex XSOAR.

#### Commands

Cortex XSOAR uses the following commands:

- System commands: Commands that enable you to perform Cortex XSOAR operations, such as clearing the playground or closing an incident. These commands are not specific to an integration. System commands are entered in the command line using a /.
- External commands: Integration-specific commands that enable you to perform actions specific to an integration. For example, you can quickly check the reputation of an IP address. External commands are entered in the command line using a !. For example, !ip.

#### Content packs

All Cortex XSOAR content is organized in packs. Packs are groups of artifacts that implement use cases in the product. Content packs are created by Palo Alto Networks, technology partners, consulting companies, MSSPs, customers, and individual contributors. Content packs may include a variety of different components, such as integrations, scripts, playbooks, and widgets.

#### Content repository

The content repository functionality built into Cortex XSOAR allows you to sync content between development and production machines using a private repository.

## Context data

Different commands and playbook tasks are tied together by the Cortex XSOAR context. Every incident and playbook has a place to store data called the context. The context stores the results from every integration command and every automation script that is run. It is a JSON storage for each incident. Whether you run an integration command from the CLI or from a playbook task, the output result is stored in the JSON context in the incident or the playground. For example, the command `!whois query="paloaltonetworks.com"` returns the data and stores the results in the context.

## Dashboards

Dashboards include visualized data, including Cortex XSOAR incident, indicator, and system data, displayed for a rolling, relative time frame. Dashboards enable you to track metrics, analyze trends that appear in your Cortex XSOAR data, and identify areas of concern. Dashboards can be customized with widgets that focus on the data points most relevant to your organization.

## Engines

An engine is a proxy server application that is installed on a remote machine and enables communication between the remote machine and the Cortex XSOAR tenant. You can run playbooks, scripts, commands, and integrations on the remote machine and the results are returned to the tenant.

You mainly use engines for the following:

- Integration instances for On-prem applications. For example, the GitLab v2 integration enables you to run commands on GitLab instances.
- Execute scripts and commands that require access to On-prem resources. For example, the Active Directory v2 integration enables you to run commands in Active Directory.
- Generic Indicator export service. In Cortex XSOAR, you can configure an EDL to share a list of Cortex XSOAR indicators with other products in your network, such as a firewall or SIEM. For example, your Palo Alto Networks firewall can add IP address and domain data from the EDL to block or allow lists.
- Load balancing which enables the distribution of the command execution load.

## Incidents

Incidents are potential security data threats that SOC administrators identify and remediate. There are several incident triggers, including:

- SIEM alerts
- Mail alerts
- Security alerts from third-party services, such as SIEM, mailboxes, data in CSV format, or from the API

Cortex XSOAR includes several out-of-the-box incident types, and users can add custom incident types with custom fields, as necessary.

## Incident fields

Incident fields are used for accepting or populating incident data. You create incident fields to hold information received from third-party integrations, manual input, or via the API.

## Incident lifecycle

You can define integrations with your third-party security and incident management vendors. You can then trigger events from these integrations that become incidents. After the incidents are created, you can run playbooks on these incidents to enrich them with information from other products in your system, which helps you complete the picture. In most cases, you can use rules and scripts to determine if an incident requires further investigation or can be closed based on the findings. This enables your analysts to focus on the minority of incidents that require further investigation.

## Indicators and Indicator types

DBot can simplify your incident investigation process by collecting and analyzing information and artifacts found in War Room entries. Cortex XSOAR analyzes indicators to determine whether they are malicious. Using indicator types reveals predefined, regular expressions in the War Room.

Hits are indicators that are determined to have a malicious verdict and were previously identified in the network. The verdict is the indicator's level of maliciousness, determined manually or by hypersearch scripts. If a hypersearch script identifies an indicator, the source is DBot.

There are many out-of-the-box indicator types, but you can add custom indicator types as necessary. The following is a list of some of the indicator types, but the list is not exhaustive:

- IP address (IP4, IP6)
- Registry key
- URL
- Email
- File hash (SHA-1, MD5)
- Domains
- CIDR

When you add an indicator type, you can add formatting, enhancement, and reputation scripts, as well as reputation commands. Formatting scripts modify how the indicator is displayed in the War Room and reports. Enhancement scripts enable you to gather additional data about the highlighted entry in the War Room. Reputation scripts calculate the reputation score for an entry that DBot analyzed, for example, DataIPReputation, which calculates the reputation of an IP address. Reputation commands (such as !ip for IP addresses) are an alternate way to calculate an indicator's reputation score (verdict) and gather additional data about the indicator. Reputation commands and reputation scripts are executed when enriching a specific indicator type (for example, when the indicator is extracted from an incident).

## Integrations

Integrations are third-party tools and services that the Cortex XSOAR platform works with to orchestrate and automate SOC operations.

### NOTE:

Cortex XSOAR 8 currently does not support the IoT Security Third-party Integrations Add-on. For more information, see the IoT Security documentation.

In addition to third-party tools, you can create your own integration using the Bring Your Own Integration (BYOI) feature.

The following lists some of the integration categories available in Cortex XSOAR. The list is not exhaustive, and highlights the main categories:

- Analytics and SIEM
- Authentication
- Case Management
- Data Enrichment & Threat Intelligence
- Database
- Endpoint
- Forensics and Malware Analysis
- IT Services
- Messaging
- Network Security
- Vulnerability Management

## Integration instance

A configuration of an integration. You can have multiple instances of an integration, for example, to connect to different environments. If you are an MSSP and have multiple tenants, you can configure a separate instance for each tenant.

## Jobs

You can create scheduled events using jobs. Jobs are triggered either by time-triggered events or feed-triggered events. For example, you can define a job to trigger a playbook when a specified TIM feed finishes a fetch operation that included a modification to the list.

## Marketplace

Marketplace is the central location for installing, exchanging, contributing, and managing all of your content, including playbooks, integrations, scripts, fields, layouts, and more.

When a content pack is available for update, you see an Updates waiting in Marketplace notification in the side menu. You can update to the latest content version or to a specific version. All dependent content packs update automatically with the content pack. We recommend periodically reviewing your installed Marketplace packs for any available updates, and updating, as required.

## Playbooks

Playbooks are self-contained, fully documented prescriptive procedures that query, analyze, and take action based on the gathered results. Playbooks enable you to organize and document security monitoring, orchestration, and response activities. There are several out-of-the-box playbooks that cover common investigation scenarios. You can use these playbooks as-is, or customize them according to your requirements. Playbooks are written in YAML file format using the COPS standard.

A key feature of playbooks is the ability to structure and automate security responses, which were previously handled manually. You can reuse playbook tasks as building blocks for new playbooks, saving you time and streamlining knowledge retention.

## Playground

The Playground is a non-production environment where you can safely develop and test scripts, APIs, commands, and more. It is an investigation area that is not connected to a live (active) investigation.

To erase a playground and create a new one, in the Cortex XSOAR CLI run the `/playground_create` command.

## Reports

Reports include visualized data, including Cortex XSOAR incident, indicator, and system data, which can be run for a specific time frame and automatically sent via email to internal or external stakeholders.

## Scripts

The Scripts page is where you manage, create, and modify scripts. These scripts perform a specific action, and are comprised of commands associated with an integration. You write scripts in either Python or JavaScript. Scripts are used as part of tasks, which are used in playbooks and commands in the War Room.

The Scripts section includes a Script Helper, which provides a list of available commands and scripts, ordered alphabetically.

## War Room

The War Room is a collection of all investigation actions, artifacts, and collaboration pieces for an incident. It is a chronological journal of the incident investigation. You can run commands and playbooks from the War Room and filter the entries for easier viewing.

## 18.2 | How to search in Cortex XSOAR

### Abstract

Search Cortex XSOAR using Lucene query syntax, the search box, or general search.

Cortex XSOAR comes with a very powerful search capability. You can search for data using the following:

- The search query
- The search box
- Free text search
- General search

### The Search Query

The search follows the Bleve query syntax. Bleve query syntax is similar to Lucene query syntax, but with some differences, such as query syntax for numeric ranges and date ranges. The search is performed on certain pages such as incidents, indicator, or the entire data (such as titles, entries, chats).

To explicitly use the following characters in a search query, place them within double quotes. An escape character \ is not required.

`&&, ||, !, {, }, [, ], (, ), ~, *, ?`

To explicitly use the following characters in a search query, place them within double quotes and use an escape character \.

`\, \n, \t, \r, ", ^, :, comma, and space`

### Basic syntax of the search

You can add some of the following inputs, when searching for data:

Input	Description
Add text	Type any text. The results show all data where one of the words appears. For example, the search <b>low virus</b> returns all data where either the string, <b>low</b> or the string, <b>virus</b> appears.
<b>and</b>	Searches for data where all conditions are met. For example, <b>status:Active and severity:High</b> finds all incidents with an active status that have a high severity.
<b>or</b>	Searches for data where either conditions are met. For example, <b>status:Pending or severity:Critical</b> finds all incidents with a pending status and with severity high or critical.
<b>*</b> <b>?</b>	Wildcard search: <b>*</b> and <b>?</b> should be used when searching for partial strings. For example, when searching for all scripts that start with AD, use <b>AD**</b> . If you need to search for a script which contains "get", search for <b>*get*</b> .
<b>""</b>	An empty value.
<b>-</b>	Excludes from any search. For example in the Incidents page the <b>-status:closed -category:job</b> searches for all incidents that are not closed and for categories other than jobs.
<b>"me"</b>	Filters incidents by a user's account. For example, <b>owner:{me}</b> will display all incidents where I am the owner. It can also be used for other fields such as <b>createdBy:{me}</b> which will display all incidents I created.
Relative time. For example, "today", "half an hour ago", "1 hour ago", "5 minutes ago", "10 days ago", "5 seconds ago", "five days ago", "a month ago", "in 1 year".	<p>Relative time in natural language can be used in search queries. Time filters <b>- &lt; and &gt;</b> can be used when referring to a specified time, such as <b>dueDate:&gt;="2018-03-05T00:00:00 +0200"</b>.</p> <p><b>NOTE:</b></p> <p>The timezone for searches is UTC. The system timezone is not used.</p> <p>When adding some fields, such as <b>Occurred</b> you can enter the date from the calendar. You can also filter the date when the results are displayed.</p>

You can also search using Regex. To use Regex, you need to use the value **"//"**. For example, to search for indicator values that contain www and end with .com, type: **value: "/w{3}..\*.com/"**. This returns values such as [www.namecheap.com](http://www.namecheap.com), [www.kloshpro.com](http://www.kloshpro.com).

To search for indicator values that contain lower-upper a-z letters and 0-9 numbers with a length of 32, type: **value: "/[a-zA-Z0-9]{32}/"**. This returns values such as **775A0631FB8229B2AA3D7621427085AD**, **87798e30ca72f77abe624073b7038b4e**.

### The search box

The search box searches for incidents, investigations, and indicators. The search box appears in the top right-hand corner on most pages. You can either type free text or search using the search query format (use the arrow keys to assist you in the search). For example, **incident.severity:Low** searches for all incidents that have **low** in the severity category.

#### NOTE:

For precise results when searching for all long text, phase, name, reason, details or type, set the Server Configuration, **incident.search.exact.match.only** to true. For example, when doing a search for **type:Phish Mail**, if the server configuration is set to true, the results returned include the exact text **Phish Mail** and not each word separately. Another option to return exact text, just for name, type and phase, is to add the term "raw" preceding the query in your search. For example, rather than just entering **type:Phish Mail**, type **rawType:"Phish Mail"**.

### Free Text

A free text search is used in the Playbooks and Scripts pages. You can search using part or all of the component's name. The component tag or description is included in the search. You can also search for an exact match of the component name by putting quotation marks around the search text. For example, searching for **"AddEvidence"** returns the script with that name. You can search for more than one exact match by including the logical operator **"or"** in-between your search texts in quotation marks. For example, searching for **"AddEvidence" or "AddKeyToList"** returns the two scripts with those names. Wildcards are not supported in free text search.

## General Search

Use a general search. For example, when searching for a table in the Users tab, searching for a widget, or a task in a playbook.

## 18.3 | How to use markdown in Cortex XSOAR

### Abstract

Use markdown to add basic formatting to text in multiple contexts within Cortex XSOAR.

You can use Markdown in many places within Cortex XSOAR. Some of the more common places are:

- Threat intel reports
- Command line interface (CLI)
- Scripts
- Playbook tasks
- Widgets
- Incident fields
- Lists

In most contexts where Markdown is supported, a Markdown editor is available to help you apply styles and view a preview of how those styles will look.

### Markdown Syntax

Most Markdown syntax elements within Cortex XSOAR are identical to those used in basic and extended Markdown syntax. For more information about markdown syntax, see <https://www.markdownguide.org/>.

The following Markdown elements used in Cortex XSOAR and exposed in the Markdown editor follow the same syntax as basic/extended Markdown:

- Bold
- Italics
- Strike-through
- Headings
- Lists (unordered/ordered)
- Links
- Code

#### NOTE:

Using the Insert code button in the Markdown editor adds three backtick quotes, which allows inclusion of a literal backtick character within the code snippet.

- Tables

#### NOTE:

You can use the Insert table button in the Markdown editor to easily create a table with up to five rows/columns.

- Images
- Blockquote

Additional elements not exposed in the Markdown editor can also be applied, such as: `letter-spacing`, `text-shadow`, `font-weight`, `font-size`.

### Cortex XSOAR markdown elements

Cortex XSOAR supports additional elements not found in basic/extended Markdown that provide useful functionality when working with Cortex XSOAR. For example:

Markdown Element	Syntax/Description
Underline	+This text will be underlined+
Text alignment	<--->Left Aligned Text<--->Center Aligned Text<--->Right Aligned Text
Highlight text	==This text will be highlighted==
Text color	<pre data-bbox="488 518 1429 550">{{color:#fd0800}}(This text will be in red) OR {{color:red}}(This text will be in red)</pre> <p data-bbox="504 563 572 590"><b>NOTE:</b></p> <p data-bbox="504 608 1139 635">You can use the name of the color, or the color code (hex triplet format)</p>
Text background color	<pre data-bbox="488 707 1421 761">{{background:#fd0800}}(This text will have red background) OR {{background:red}}(This text will have red background)</pre> <p data-bbox="504 774 572 801"><b>NOTE:</b></p> <p data-bbox="504 819 1472 961">You can use either the name of the color, or the color code (hex triplet format). You can use text color and text background color in parallel. For example (using the editor buttons): {{background:red}} ({{color:blue}})(This text will be in blue with red background)). Or, alternatively, if you are manually applying attributes, you can include both types in a single bracket: {{background:red;color:blue}}(This text will be in blue with red background)). For both text color and text background color, you can select Show custom options in the Markdown editor to select or enter a specific color code (hex triplet format).</p>
Keyboard input style	<pre data-bbox="488 1021 564 1048">[[kbd]]</pre> <p data-bbox="504 1066 572 1093"><b>NOTE:</b></p> <p data-bbox="504 1111 901 1138">Denotes textual user input from a keyboard.</p>
Linking to other Cortex XSOAR incidents	#Incident ID Number
Upload a local image	<p data-bbox="488 1329 1466 1414">You can upload a local image that is not available on the internet to the Markdown editor. Copy/paste or drag a local image into the Markdown editor, which automatically applies the standard image syntax and adds a relative path to the image.</p> <p data-bbox="504 1437 572 1464"><b>NOTE:</b></p> <p data-bbox="504 1482 1417 1536">Within the War Room, when the Markdown editor is open, you will only be able to drag images into the Markdown editor. To drag images into the War Room, first close the Markdown editor.</p>
Button	<pre data-bbox="488 1596 1461 1650">%%{"message": "This is a Button", "action":"Command Name", "params": { "param1": "val1", "param2": "val2"}}%%</pre> <p data-bbox="504 1673 572 1700"><b>NOTE:</b></p> <ul data-bbox="541 1724 1445 1821" style="list-style-type: none"> <li>Buttons are available only in entries of incidents.</li> <li>If data between the two sets of %% is not parsed as JSON, all of the data is taken as a command to render. For example, %%!Print value='test'%% causes the button to run !Print.</li> </ul>

Some extended Markdown syntax may not be supported. For example, checkboxes and footnotes.

## 18.4 | User details and preferences

### Abstract

Cortex XSOAR users can control user details and preferences, and notifications.

Each user can define their own details and preferences. To configure, click your username, and select User Preferences. These preferences do not affect other users.

## Details

Edit your first and last name and reset your password. Your password must meet the requirements of your organization's password policy.

### User preferences

Section	Description
General	<ul style="list-style-type: none"> <li>Enable or disable keyboard shortcuts</li> <li>Choose a default landing page that appears when you log in.</li> </ul>
My avatar	Upload an image to display as your avatar.
Keyboard Shortcuts	Change the shortcut letter used to open the GoTo bar to search, investigate, and initiate actions. To change the shortcut, click the letter in the box, type a letter, and then save. The shortcut value must be a keyboard letter (A to Z).
Timezone	Select the timezone to display your Cortex XSOAR data, which affects the timestamps displayed in Cortex XSOAR, such as auditing logs, and exported files.
Timestamp Format	The timestamp format is displayed in data tables, auditing logs, and exported files.

#### NOTE:

If Keyboard Shortcuts, Timezone, and Timestamp Format do not appear in the Preferences tab, the default keyboard shortcut can only be set on the Server Settings page.

## Notifications

Each user can define their notifications. These settings do not affect other users.

You can configure which notifications to receive and via what channels. Notifications are presented by categories:

- My Incidents
- My Playbook Tasks
- My To-Do Tasks
- Other Notifications

By default, all of the categories and available channels are selected.

Email notifications are enabled by default. Every email notification from which you can unsubscribe includes a link at the bottom of the email to bring you directly to the User Preferences page where you can edit your notification settings.

If your organization has integrations with Slack or other messaging applications, you can choose to enable or disable notifications through those applications.

#### NOTE:

To set your active/away status, click your username and go to Set Yourself as Away. Once you are set as away, a Zz icon appears next to your name. To change the status, select Set Yourself as Active. Other users see you as active or away in dropdown lists, such as when assigning an owner to an incident. Your status can also be set by entering !setYourselfAs in the command line.

## 18.5 | Server configurations

### Abstract

Customize and troubleshoot Cortex XSOAR with server configuration settings.

Cortex XSOAR provides custom server configuration settings that enable you to customize your Cortex XSOAR on the tenant level. You can also use custom server configuration settings in situations where you experience issues or need to troubleshoot situations in your environment.

To modify or add server configurations:

1. Navigate to Settings & Info → Settings → System → Server Settings → Server Configuration.
2. Click Add Server Configuration or edit an existing configuration.
3. Enter the key and value.
4. Click Save.

#### Engines

Key	Description	Default
engine.test.command.timeout<brand-name>	<p>Increases the timeout, in seconds, for a specific integration when using an engine. For example, change it to 300 seconds. Type in this format adding the brand name:</p> <pre>engine.test.command.timeoutTaniumengine.test.command.timeoutTanium</pre>	60
engines.notification.users	<p>Specifies which users receive an email notification when an engine disconnects. A comma-separated list of Cortex XSOAR users. For example:</p> <pre>user1,user2,user3user1,user2,user3</pre>	N/a

#### Google API

Key	Description	Default
UI.google.api.key	<p>Entities that have Geo-location information (latitude and longitude) can be displayed on a Google map, by utilizing the Google Map API (which is required). For example, if you want to see the physical location of a computer that was attacked by Malware. To display the physical location of an entity on a map, run this command with the value: Google Maps API Key. For more information, see Set up Google Maps in Cortex XSOAR to use map automations.</p>	N/a

#### Incidents

Key	Description	Default
incident.closereasons	<p>Customizes close reasons in a comma-separated list. For example:</p> <pre>false positive, resolved, duplicate, low priority, invalid, other</pre>	<pre>false positive, resolved, duplicate, other</pre>

Key	Description	Default
inline.edit.on.blur	<p>By default, when editing the following inline values in an incident/indicator/threat intel report, the changes are not saved until you confirm your changes (clicking the checkmark icon in the value field).</p> <ul style="list-style-type: none"> <li>Dropdown values, such as Owner, Severity, etc.</li> <li>Text values, such as Asset ID. (You can only edit when you click the pencil in the value field).</li> </ul> <p>These icons are designed to let you have an additional level of security before you make changes to the fields in incidents/indicators.</p> <p>Set this configuration to true, to enable you to make changes to the inline fields without clicking the checkmark. The changes are automatically saved when clicking anywhere on the page or when navigating to another page. For text values, you can also click anywhere in the value field to edit.</p>	false
investigation.prevent.modify.closed	Whether to add chats and notes to the closed investigation (set to false to allow).	true
module.health.notification.users	List of names in CSV format to receive notifications when an integration experiences a fetch error. For more information, see Receive notifications on an incident fetch error.	N/a
Export.utf8bom	Whether to export incidents and indicators to CSV using the UTF8-BOM format. For more information, see Export an incident to CSV using the UTF8-BOM format.	False

**Indicators**

Key	Description	Default
enrichment.reputationScript.reliability	The reliability of the score from a reputation script.	A++
indicator.timeline.auto.extract.enabled	Enables the indicator timeline in the indicator extraction flow. For more information, see Configure the indicator timeline.	true
indicator.timeline.enabled	Enables the indicator timeline in all flows. For more information, see Configure the indicator timeline.	true

**Integrations**

Key	Description	Default
<integration_name>.<command_name>.timeout	Timeout in minutes for specific integration commands.	3
sync.mirror.job.delay	The interval for the job in minutes. For more information, see Special Server Configurations.	1

Key	Description	Default
sync.mirror.job.enable	Enable or disable the mirroring job. For more information, see Special Server Configurations.	enable

**Notifications**

Key	Description	Default
content.notification.enabled	Set to true to enable notification for new content updates.	false
content.notification.users	Notifies all users by email when there is a content update available (comma-separated user names in Cortex XSOAR).	N/a
message.ignore.failedFetchIncidents	Whether to ignore failed fetch incident messages. For more information, see Receive notifications on an incident fetch error.	false
message.ignore.incidentChanged	Whether to disable notifications when an incident is changed.	false
message.ignore.incidentOpenedincidentOpened	Whether to disable notifications, when an incident is opened.	false
message.ignore.incidentAssigned	Whether to disable notifications when an incident is assigned.	false
message.ignore.investigationClosedinvestigationClosed	Whether to disable notifications when an incident is closed.	false
module.health.notification.users	List of names in CSV format. For example, <b>user1,user2,user3</b> . For more information, see Receive notifications on an incident fetch error.	N/a
server.notification.using.send-mail	Select which email sender should send the notification. For more information, see .	

**Playbooks**

Key	Description	Default
soc.name	Customizes the SOC name in the survey header for an Ask task. For more information, see Customize the SOC name.	N/a
comm.ask.linktocontext.enabled	Whether to display the links generated for an Ask task in the Context Data of the Work Plan.	true
comm. datacollection.linktocontext.disabled	Whether to display the links generated for a Data Collection task in the Context Data of the Work Plan.	true

Key	Description	Default
ignore.default.in.playbooks	Whether to allow the Do Not Use By Default checkbox to affect playbooks. By default the Cortex XSOAR playbook does not take Do not use by default into account (only for CLI Commands). For example, if you have 3 mail sender instances, 2 of them are set to not use by default, when running the playbook without specifying an instance, it sends with all 3 instances. After you set this configuration to true, it only sends from the one that is not marked as do not use by default.	false

**Proxy**

Key	Description	Default
condition.ask.external.link	The address (including the HTTPS prefix) of the proxy used for external user communication in a conditional task.	N/a

**Remote Repository**

Key	Description	Default
UI.version.control.admin.only	<p>Set to true to restrict access for pushing content to a remote repository to administrators only.</p> <p><b>NOTE:</b></p> <p>When set to true this key also removes the Save Version feature. This prevents users who don't have administration permissions from pushing content changes to the remote repository.</p> <p>For more information, see Remote repository management.</p>	false

**Reports**

Key	Description	Default
reports.time.zone	Configure the timezone for widgets in a report. For more information, see Configure the timezone in a report.	Local time/Location

**Scripts**

Key	Description	Default
script.timeout	The timeout, in minutes, to prevent blank pages when running a script. If you generate a report that runs a script and has blank pages you can Troubleshoot the script timeout. For more information, see Troubleshoot script timeout for reports.	3

## System Settings

Key	Description	Default
UI.show.timezone.in.server.settings	If set to true, settings for Keyboard Shortcuts, Timezone, and Timestamp format appear on the Server Settings page. By default, these settings instead appear on the Preferences tab of the User Details page.	false

## SLA

Key	Description	Default
sla.risk.threshold	Change the SLA risk threshold.	72 hours

## Widgets

Key	Description	Default
ROI.Cost.Monitor	Amount in Dollars. Relevant for ROI widget. For more information, see Saved By Dbot (ROI) Widget.	60

## 18.6 | New user FAQ

## Abstract

New User FAQ for Cortex XSOAR.

The following are frequently asked questions for new Cortex XSOAR users.

How do I import custom content into Cortex XSOAR?

If you have a full content bundle (.tar.gz file), navigate to Settings & Info → Settings → System → Server Settings and scroll to Custom content. Browse for the file or dragging into the Upload custom content box.

You can also import specific content types, such as playbooks, by navigating to that section of Cortex XSOAR and then clicking the upload button in the upper right corner of the page.

How do I export custom content from Cortex XSOAR so I can share it?

Navigate to Settings & Info → Settings → System → Server Settings and scroll to Custom content. Click Export all custom content to download a compressed file containing all of the custom content from your instance.

You can also export individual content items, such as playbooks, by selecting the content item, clicking the triple dot menu in the upper right corner of the page, and clicking the Download button.

How do I configure my Cortex XSOAR notification settings?

Navigate to your username and select Username → User Preferences → Notifications. By default, all notifications are enabled. De-select the checkboxes for notifications you don't want to receive.

How do I get notifications via Slack/Teams/other chat applications?

Configure an integration instance for that chat application. As long as the integration instance implements the **send-notification** command, it appears on the Notifications tab.

What is the difference between a dashboard and a report?

Dashboards show data from a rolling, relative time frame from a certain time in the past (for example, 7 days ago) through the present and are shown when you log into Cortex XSOAR. Reports allow you to share similar data outside of Cortex XSOAR via email. Reports can be scheduled to run at a specific time to capture data where the start/end time is important. For example, if management requests a report on the incidents that occurred between 08:00 yesterday and 08:00 today.

How do I access the playground?

The link to the playground appears at the bottom of the My Incidents menu item in the left sidebar. You can also access the shortcut option using **ctrl+alt+k** and type **playground** or go directly to <https://<tenant>/WarRoom/playground/>

How do I update all of my installed content packs at the same time?

Navigate to Marketplace → Installed Content Packs. From the Show list, select Update available. Click the checkbox to select all, then click the Update button.

How do I search for incidents/indicators/playbooks/scripts and more in Cortex XSOAR?

Cortex XSOAR comes with a powerful search capability that uses the Lucene query syntax. For example, to search playbooks:

- Search for the playbook with the exact name "Phishing - Generic v3": `name:"Phishing - Generic v3"`
- Search for playbooks where the word "Phishing" appears anywhere in supported system objects: `Phishing`
- Search for playbooks where the playbook name contains "Phishing": `name:"Phishing"`

## 18.7 | Telemetry in Cortex XSOAR

Abstract

Cortex XSOAR uses telemetry to collect specific usage data. The data is analyzed and used to improve Cortex XSOAR.

Cortex XSOAR uses telemetry to collect specific usage data. This data is analyzed and used to improve Cortex XSOAR and to identify common usage to help drive the product roadmap.

You can limit or turn off telemetry (apart from essential information according to your license type). For more information, see Configure server settings.

## 18.8 | Product support lifecycle

Abstract

Cortex XSOAR product support lifecycle.

Cortex XSOAR's support and End-of-Life (EoL) policies depend on your version.

Cortex XSOAR 8 support and EoL

Cortex XSOAR On-prem versions are generally available for upgrade every 3 months.

Cortex XSOAR Version	EOL
8.5	April 1, 2025
8.6	July 14, 2025

Engines

Cortex XSOAR maintains backward compatibility with N-2 engine versions. For example, when Cortex XSOAR 8.6 is GA and deployed, we have backward compatibility for Cortex XSOAR engine versions 8.4, 8.5, and 8.6. For more information about upgrading your engine, see Upgrade an engine.

Also, make sure that the Cortex XSOAR tenant version is not EoL. For example, if you are running Cortex XSOAR 8.6, but 8.4 is EoL, the 8.4 engine is not supported.

### NOTE:

It is highly recommended to run engines on the latest version. New features, performance improvements, and bug fixes are only provided on the tenant's version. For example, if the tenant is running on 8.6 but the engine is running on 8.4 and a bug is found, the fix for the bug will require you to upgrade the engine to 8.6.

Cortex XSOAR 6 On-prem support and EoL

For Cortex XSOAR 6 EoL, see Product Support Lifecycle.

You can continue using a version that is EoL.

## 18.9 | Keyboard shortcuts

### Abstract

Keyboard shortcuts to navigate and manage playbooks, scripts, CLI, and incident pages.

The following keyboard shortcuts enable you to quickly navigate and manage Cortex XSOAR.

### Playbooks, Scripts, and CLI

Description	Mac	Windows	Available In Pages
Fast Navigation	Command-K	Ctrl + K	All
Focus on the CLI	Command-;	Ctrl + ;	All
Save the Playbook	Command-S	Ctrl + S	Playbooks (edit mode)
Auto-Align the Playbook	Command-L	Ctrl + L	Playbooks (edit mode)
Copy Task(s)	Command-C	Ctrl + C	Playbooks (edit mode)
Paste Task(s)	Command-V	Ctrl + V	Playbooks (edit mode)
Cut Task(s)	Command-X	Ctrl + X	Playbooks (edit mode)
Select task(s)	Shift + drag with the mouse	Shift + drag with the mouse	Playbooks (edit mode)
Undo	Command-Z	Ctrl + Z	Playbooks (edit mode)
Delete task(s)	Delete	Delete	Playbooks (edit mode)
Save the script or integration that is currently being edited	Command-S	Ctrl + S	Scripts, Integrations
Create a New Script	Command-I	Ctrl + I	Scripts
Create a New Incident	Command-I	Ctrl + I	Home, Incidents
Open the Markdown Box and the Toolbar in CLI	Command-M	Ctrl + M	CLI area in all pages
Show or hide the CLI	Option-F	Alt + F	CLI area in all pages

## Open and Close Side Panels on the Incident Page

Description	Mac	Windows
Open/Close the Incident Summary Quick View	Option-Q	Alt + Q
Open/Close the Systems Windows	Option-W	Alt + W
Open/Close the Incident Team Window	Option-E	Alt + E
Open/Close the Incident Context View	Option-R	Alt + R

## 18.10 | Cortex XSOAR navigation cheat sheet

### Abstract

Learn about commonly used features of Cortex XSOAR.

The main menu for Cortex XSOAR includes:

Feature	Description
My Incidents	Includes your favorites, incidents you own, and incidents you have participated in.
Dashboards & Reports	<p>Dashboards include visualized data, including Cortex XSOAR incident, indicator, and system data, displayed for a rolling, relative time frame. Dashboards enable you to track metrics, analyze trends that appear in your Cortex XSOAR data, and identify areas of concern. Dashboards can be customized with widgets that focus on the data points most relevant to your organization.</p> <p>Reports also contain visualized data, but can be run for a specific time frame and automatically sent via email to internal or external stakeholders.</p>
Incidents	<p>On the Incidents page, you can search for and interact with incidents that have been ingested from third-party integrations or manually created in Cortex XSOAR.</p> <p>Incidents enable you to organize your investigation and response work. Each incident is a self-documenting IR workbench where you can view incident details in a custom layout, run scripts and playbooks on the incident, create notes, tag evidence items, and more.</p>

Feature	Description
Threat Intel (Indicators)	<p>The Threat Intel page displays a table or summary view of all indicators.</p> <p><b>NOTE:</b></p> <p>If you do not have a TIM license, the page is titled Indicators. Most Threat Intel features are available only with a Cortex XSOAR Threat Intelligence license.</p> <p>Includes the following:</p> <ul style="list-style-type: none"> <li>• Indicators: Indicators database. Search, review, and interact with indicators including IPs, domains, URLs, hashes. Research threats and correlate indicators of compromise across multiple incidents. Track indicator properties such as their verdict and add tags to apply your own indicator classification and grouping logic.</li> <li>• Sample Analysis (TIM license only): View detailed file sample analysis results from PANW WildFire. Conduct in-depth research and analysis of file sample behaviors and characteristics based on WildFire's sandboxed detonation of the file.</li> <li>• Sessions &amp; Submissions (TIM license only): For users of PANW firewalls, WildFire, Cortex XDR, Prisma SaaS, and/or Prisma Access, search and view firewall sessions and file sample submission data from these products. Correlate file hashes observed in firewall sessions or submitted through other PANW products with hashes in Cortex XSOAR.</li> <li>• Threat Intel Reports (TIM license only): Build and share rich threat intelligence reports. Share threat intelligence reports with stakeholders either within or outside of Cortex XSOAR.</li> </ul>
Playbooks	<p>On the Playbooks page, you can browse, create, and customize Cortex XSOAR playbooks, which are workflows that link together ordered response steps including scripts, manual tasks, and communication tasks.</p> <p>Playbooks enable you to standardize and orchestrate your IR processes. A playbook helps ensure users follow a consistent response process, automates mundane response tasks, ties together your different IR tools, and gathers all relevant incident context and enrichment data in one centralized place.</p> <p><b>NOTE:</b></p> <p>You can copy/paste tasks from one playbook to another by using keyboard shortcuts.</p>
Scripts	<p>On the Scripts page, you can browse, create, and customize Python, PowerShell, and JavaScript scripts for use in Cortex XSOAR. View the code for out-of-the-box scripts in order to troubleshoot, better understand, or build upon them. You can create custom scripts to extend Cortex XSOAR's functionality to achieve your automation goals.</p>
Jobs	<p>Jobs allow you to schedule playbooks to run on a recurring basis, either at a specific time or triggered by new indicators ingested from a feed integration. With jobs, you can automate actions you would normally take on a recurring basis, such as compiling malicious indicators and sending them to the SOC for verification before they are blocked.</p>
Marketplace	<p>The Cortex Marketplace provides access to hundreds of integrations that extend the functionality of Cortex XSOAR and allow communication with third-party services. Includes the following:</p> <ul style="list-style-type: none"> <li>• Browse: The central location for searching and installing Cortex XSOAR content, including playbooks, integrations, and scripts.</li> <li>• Installed content packs: View and manage your installed Cortex XSOAR content packs.</li> <li>• Contributions: Contribute content that you have created, including playbooks, integrations, and scripts.</li> <li>• Deployment Wizard: The Deployment Wizard significantly reduces the time required to set up your use case. It guides you through the process of setting up your content pack for your specific use case. Relevant for phishing and malware content packs.</li> </ul>

Feature	Description
Settings & Info	<p>Includes the following:</p> <ul style="list-style-type: none"> <li>• Cortex Gateway: Cortex Gateway allows you to activate new tenants and view and manage existing tenants and tenants available for activation that are allocated to your Customer Support Portal account.</li> <li>• Cortex XSOAR License: View information about the licenses, expiry dates, and the number of licensed and active users.</li> <li>• Management Audit Logs: View and export a historical audit trail of user actions taken in Cortex XSOAR.</li> <li>• Settings: Access the detailed Settings menu.</li> </ul>
Tenant Navigator	<p>if you have more than one Customer Support Portal account, you can view and pivot to all the tenants that you have access to, by clicking Tenant Navigator. In the <b>Tenant Navigator</b>, you can do the following:</p> <ul style="list-style-type: none"> <li>• View existing tenants The currently chosen tenant is marked by a green Active Session label. The tenants are grouped according to Customer Support Portal accounts.</li> <li>• Pivot to an existing tenant The current tenant is marked by a green Active Session label.</li> <li>• Search for a tenant If there are more than 5 tenants, a search option is available. If there are more than 5 tenants within a specific account, a list of tenants is available for that Customer Support Portal account.</li> <li>• Pivot to Cortex Gateway</li> <li>• Pivot to the Customer Support Portal</li> </ul> <p><b>NOTE:</b> If you do not have more than one account, the Tenant Navigator is unavailable.</p>
User Menu (username)	<ul style="list-style-type: none"> <li>• About: Detailed information on Cortex XSOAR version.</li> <li>• User preferences: Change default landing page and configure notifications via your preferred communication method. Customize your display to suit your preferences. Get notified of Cortex XSOAR events of interest to you, such as being assigned an incident. Disable unwanted notifications.</li> <li>• Set Yourself as Away: Change your away/active status.</li> <li>• Log out: Log out of the Customer Support Portal</li> </ul>