DUMPSBOSS.

Palo Alto Networks XSOAR Engineer

Palo Alto Networks XSOAR-Engineer

Version Demo

Total Demo Questions: 23

Total Premium Questions: 233

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Planning and Deploying Cortex XSOAR	13
Topic 2, Cortex XSOAR Integrations	31
Topic 3, Cortex XSOAR Incidents and Indicators	99
Topic 4, Cortex XSOAR Playbooks	52
Topic 5, Cortex XSOAR Administration and Maintenance	38
Total	233

QUESTION NO: 1

A playbook task generates a report as HTML in the context data.

An engineer creates a custom indicator field of type "HTML" and adds the field to a section in a custom indicator layout. How can the engineer populate the HTML field in the indicator layout?

A. Populate the custom indicator field with the built-in !SetIndicator command.

B. Add HTML to a list using !setList and use it as an HTML template to populate the custom indicator field.

C. Create a custom Indicator Mapper and populate the custom indicator field.

D. Use the Mapping option in the playbook task that generates the HTML report to populate the custom indicator field.

ANSWER: D

Explanation:

Use the Mapping option in the playbook task that generates the HTML report to populate the custom indicator field. Task mapping is designed to take a task output—such as the HTML report value placed in the playbook context—and map it directly to a destination field. In this case, the destination is the custom indicator field whose type is HTML. When an analyst opens an indicator using the custom layout, XSOAR renders the stored HTML value in that field, allowing the report to be displayed as part of the indicator details.

The mapping should be configured on the task that produces the report, selecting the appropriate context path for the generated HTML and the relevant custom indicator field as the target. This approach keeps the data flow declarative within the playbook and associates the report with the specific indicator record being processed. It also allows the indicator layout section to present the same field consistently whenever that indicator is viewed. Palo Alto Networks documents playbook task configuration and task input/output handling in the Cortex XSOAR Playbooks documentation and describes indicator configuration concepts in the Indicators documentation.

QUESTION NO: 2

Which two capabilities do Automation script settings include? (Choose two.)

A. Define ‘parameters’

B. Correlate to incident types

C. Define ‘outputs’

D. Set password protection

ANSWER: A C

Explanation:

Define ‘parameters’ and **Define ‘outputs’** are capabilities configured for an XSOAR automation. Parameters, commonly implemented as automation arguments, define the inputs that a playbook task or command supplies when it runs the automation. Argument definitions can include details such as the argument name, whether it is required, a default value, and descriptive information that helps users provide valid input. This makes an automation reusable in different playbooks and incident contexts without changing its code.

Outputs define the context data that the automation returns after execution. Specifying output keys, their types, and descriptions lets playbook authors reliably reference the automation’s results in later tasks, conditions, transformations, and incident-field mappings. Clear output definitions also document the automation contract and make its returned data discoverable to other XSOAR users. Palo Alto Networks’ developer documentation describes handling automation arguments through the command context and returning structured results and context data from an automation. See the Cortex XSOAR code conventions and the CommonServerPython API reference for the expected argument and output patterns.

QUESTION NO: 3

An Engineer wants to filter a csvList value according to a dynamic value saved under the test context key.

Which three values would save the test context key? (Choose three.)

A. Get csvList.value where csvList.value equals test [from previous tasks]

B. Get csvList.value where csvList.value equals ${test} [from previous tasks]

C. Get csvList.value where csvList.value equals test {}[from previous tasks]

D. Get csvList.value where csvList.value equals test [as value]

E. Get csvList.value where csvList.value equals ${test} [as value]

ANSWER: A B E

Explanation:

In Cortex XSOAR, task-input mapping can obtain a value dynamically from context produced by earlier playbook tasks, or it can resolve an embedded context expression when a value is entered directly. `Get csvList.value where csvList.value equals test [from previous tasks]` uses the prior-task/context selection mechanism to supply the value stored under the `test` key. `Get csvList.value where csvList.value equals ${test} [from previous tasks]` likewise identifies the same context value using XSOAR's variable-expression syntax. `Get csvList.value where csvList.value equals ${test} [as value]` is also valid because the expression is resolved at runtime, allowing the filter to compare each `csvList.value` item to the dynamically stored context value rather than to fixed text. This behavior is useful when playbook results must drive downstream filters without hard-coding a comparison value. XSOAR context data is the shared runtime data store for playbook tasks, and task inputs can reference that data through context paths and variable expressions. See the Cortex XSOAR playbooks documentation and the context and outputs documentation for details on context paths, task outputs, and dynamic value references.

QUESTION NO: 4

When mapping incoming data to incident fields, which statement is correct?

A. Data that is not mapped is placed under labels

B. Only text fields are classified

C. Classification cannot be used if mapping is enabled

D. Every incoming field must be mapped

ANSWER: A

Explanation:

Data that is not mapped is placed under labels is correct. In Cortex XSOAR, incident classification and mapping determine how fields from an incoming event are evaluated, classified into an incident type, and assigned to defined incident fields. A mapping does not need to enumerate every source field for the event to be ingested successfully. XSOAR preserves source data that has not been explicitly mapped by storing it as incident labels. These labels are key-value pairs associated with the incident and make the original incoming attributes available for investigation, filtering, automation, and troubleshooting. This behavior lets an engineer map the fields that are important to a specific incident layout or workflow while retaining additional event context without first creating a dedicated incident field for every possible attribute. Labels can subsequently be viewed on the incident and used by playbooks or queries when needed. Palo Alto Networks documents that fields not included in the mapping are saved as labels as part of the incident ingestion and mapping process. See the Cortex XSOAR incident classification and mapping documentation and the Common Server Python reference for details on working with incident data and labels.

QUESTION NO: 5

An automation executes an integration command by using `demisto.executeCommand()`. Before processing the returned entries, the automation must determine whether the command execution returned an error.

Which CommonServerPython helper should the engineer use?

A. `is_error()`

B. `demisto.debug()`

C. `return_results()`

D. `tableToMarkdown()`

ANSWER: A

Explanation:

`is_error()` is used to determine whether the result returned from `demisto.executeCommand()` represents an error. If an error is detected, the automation can stop processing unsuccessful output and return an actionable failure, for example by calling `return_error()`.

`demisto.debug()` writes diagnostic output but does not evaluate command results. `return_results()` returns normal results to XSOAR, while `tableToMarkdown()` formats data for presentation.

QUESTION NO: 6

Which three support types are included in the Marketplace Content Packs? (Choose three.)

A. Customer supported

B. Contex XSOAR supported

C. Community supported

D. Partner supported

E. Prisma Cloud supported

ANSWER: B C D

Explanation:

Marketplace content packs are categorized by the organization responsible for maintaining and supporting the content. **Cortex XSOAR supported** identifies packs developed and maintained by Palo Alto Networks, providing vendor-managed content and support. **Partner supported** identifies packs supplied and maintained by an approved technology partner; the partner owns the associated support responsibility. **Community supported** identifies content contributed and maintained by the broader XSOAR community, enabling integrations and automations that may not be maintained directly by Palo Alto Networks or a formal partner. These support-type labels help administrators understand the appropriate support path and maintenance ownership before installing a Marketplace pack. In particular, support status is a content-pack classification shown in Marketplace metadata, rather than a statement about the customer's deployed product edition or another Palo Alto Networks product. Palo Alto Networks documents the Marketplace support classifications and their ownership model in its Content Pack Support Types documentation. For Marketplace installation and content-management context, see the Marketplace overview.

QUESTION NO: 7

A large number of incidents were deleted by mistake.

Which two architecture components can be used to recover the lost data? (Choose two.)

A. Live backup

B. Engine

C. Distributed database

D. Local backup

ANSWER: A D

Explanation:

Live backup and **Local backup** are the recovery mechanisms for restoring incident data that was deleted accidentally. Cortex XSOAR backups include database information, which contains incident records and their associated data. Live Backup is intended for managed, ongoing backup and disaster-recovery workflows: the system creates backups and stores them in the configured remote backup location, allowing an administrator to restore a prior database state when data loss occurs. Local backup provides a backup artifact stored locally that can likewise be used during the database restoration process. In either case, recovery depends on selecting a backup created before the accidental deletion and following the documented restore procedure appropriate to the deployment. Regularly maintaining both a reliable backup schedule and retention period is important because restoration returns the relevant database content to the state captured by that backup. Palo Alto Networks documents database backup and restore operations, including backup configurations and recovery considerations, in its Cortex XSOAR administration guidance. See the Cortex XSOAR database backup documentation and the disaster recovery and live backup overview.

QUESTION NO: 8

What can you use to assign a layout, field, and playbook to an incoming incident?

A. Playbook

B. Classification and mapping

C. Incident type

D. Pre-processing

ANSWER: B

Explanation:

Classification and mapping is used to process incoming incidents and make them usable in Cortex XSOAR. A classifier evaluates the incoming data and determines the appropriate incident type. That incident type supplies the incident's configured context, including its associated layout and playbook. The mapper then extracts values from the source event and places them into the appropriate XSOAR incident fields, including both standard and custom fields. This lets XSOAR consistently normalize alerts from different integrations before analysts begin triage. For example, data such as severity, source IP address, user name, or alert identifiers can be mapped from a vendor-specific payload into the selected incident type's fields, while the type establishes the relevant analyst view and response automation. As a result, Classification and mapping provides the intake mechanism that connects an incoming event to the correct incident configuration and operational workflow. Palo Alto Networks documents this process in its Classification and Mapping documentation and describes incident configuration concepts in the Incident Types documentation.

QUESTION NO: 9

An engineer is configuring incident ingestion for a source that sends phishing, malware, and informational alerts through the same integration instance.

Which two statements accurately describe classifiers and mappers in this workflow? (Choose two.)

A. A classifier can determine the incident type from values in the raw event.

B. An incoming mapper can populate incident fields from values in the raw event.

C. An incoming mapper determines which incident type is assigned to the event.

D. An outgoing mapper is used to populate fields when an incident is first fetched.

E. A classifier is used to synchronize XSOAR incident updates to an external system.

ANSWER: A B

Explanation:

A classifier evaluates the raw incoming event and determines the incident type that should be assigned. This enables a single integration instance to route different categories of source events to different XSOAR incident types.

An incoming mapper then maps relevant raw event values into the fields of the selected incident type. For example, it can populate a custom sender field for phishing incidents or an affected-host field for malware incidents. A mapper does not determine the incident type, and an outgoing mapper is reserved for sending updates to an external system during mirroring.

QUESTION NO: 10

What are the three ways to add/mark entries as evidence inside the Evidence Board? (Choose three.)

A. Manually directly from the War Room with the Actions drop-down

B. From the Notes section (mark as entry icon)

C. Manually from the playbook task (mark as entry icon)

D. Automatically from playbook tasks when the option is selected on the Advanced tab

E. By running the command !MarkAsEvidence

ANSWER: A B D

Explanation:

The Evidence Board provides a curated view of the incident artifacts that an analyst considers important for investigation, reporting, or later review. An analyst can explicitly promote an existing War Room entry by using the Actions drop-down, which preserves the relevant command output or context as an evidence item. Analysts can also promote information entered in the incident Notes section by selecting the mark-as-entry control; this makes analyst-authored observations available alongside collected artifacts on the board.

For repeatable enrichment and investigation workflows, playbooks can mark results as evidence automatically. The setting on a task's Advanced tab instructs XSOAR to add that task's produced entry to the Evidence Board when the task runs, avoiding a separate manual action and ensuring important task output is consistently captured. These methods support both analyst-driven curation and automation-driven collection, which is particularly useful when building an incident record that must show the supporting artifacts behind decisions and conclusions. Palo Alto Networks documents the Evidence Board and incident investigation workflow in the Cortex XSOAR documentation portal and provides additional implementation guidance in the Cortex XSOAR Developer Documentation.

QUESTION NO: 11

Which two reasons would lead an engineer to create a custom widget? (Choose two.)

A. To visualize server configuration keys

B. To visualize XSOAR list data

C. To visualize complex incident data calculations

D. To visualize context data

E. To visualize a custom query

ANSWER: D E

Explanation:

To visualize context data and to visualize a custom query are valid reasons to create a custom XSOAR dashboard widget. Context data is the structured data produced by integrations, automations, playbooks, and commands during an investigation. A custom widget can use that context to present operationally useful values, tables, or charted results directly on a dashboard, giving analysts a concise view of the information generated by XSOAR workflows.

A custom query is also an appropriate widget data source when the dashboard needs to present selected incident information or calculated results that are not available through an existing out-of-the-box widget configuration. Queries enable an engineer to define the relevant incident dataset and display it in a useful dashboard format, such as a number, table, or graph. Together, context-backed widgets and query-backed widgets allow dashboards to be tailored to the organization's investigation processes, operational metrics, and reporting requirements. See the Cortex XSOAR Administrator Guide and the Cortex XSOAR documentation portal for dashboard and widget configuration guidance.

QUESTION NO: 12

What must happen before a pre-process rule can be applied to a potential incident?.

A. Mapping.

B. Playbook execution.

C. Ingestion.

D. Classification.

ANSWER: C

Explanation:

Ingestion. is correct because a pre-process rule acts on an incident candidate only after Cortex XSOAR has received it from an integration or other ingestion source. The incoming event must first be fetched or otherwise ingested into XSOAR so that the platform has the raw incident data against which it can evaluate the rule's filter conditions. At this stage, the item is still a potential incident rather than a fully created and processed XSOAR incident.

Pre-processing is part of the inbound incident-handling flow. It enables XSOAR to inspect the source data early and take configured actions such as dropping unwanted events, modifying incident fields, or routing the potential incident for subsequent handling. This early placement is important because it lets an organization control which incoming events become incidents and what data is available as they proceed through the incident-processing pipeline. Therefore, ingestion is the prerequisite: without an incoming event being collected by XSOAR, there is no potential incident on which a pre-process rule can operate. See Palo Alto Networks' documentation on incident pre-processing and the incident classification and mapping workflow.

QUESTION NO: 13

An incident field is created having the display name as Source_IP. How can the field be accessed?

A. ${incident.sourceip}

B. ${incident.Source_IP}

C. ${incident.srcip}

D. ${incident.Source IP}

ANSWER: A

Explanation:

`${incident.sourceip}` is the correct way to access this incident field. In Cortex XSOAR, incident fields are referenced in playbooks, scripts, and automation inputs through the `incident` context object using the field's machine-readable name, not its display label. XSOAR normalizes a field display name such as `Source_IP` into a lowercase field key without display formatting, resulting in `sourceip`. Therefore, the field can be interpolated as `${incident.sourceip}`. This syntax enables XSOAR to resolve the value from the active incident consistently, including when the value is used in task inputs, conditional logic, layouts, or automation arguments. When creating or reviewing custom fields, administrators should verify the field's generated machine name in the incident-field configuration because that identifier, rather than the human-facing display name, is the value used in context paths. Palo Alto Networks documents incident fields as configurable objects that become available on incident records and in playbook context. See the Cortex XSOAR guide for creating incident fields and the context data documentation for the context-based field-reference model.

QUESTION NO: 14

An engineer is preparing a content pack that will distribute a complete investigation workflow to several XSOAR environments.

Which three content items can be included in the content pack? (Choose three.)

A. Integrations

B. Playbooks

C. Automations

D. Server log bundles

E. Individual fetched incident records

ANSWER: A B C

Explanation:

Content packs package related Cortex XSOAR content so that workflows and their dependencies can be installed and versioned together. **Integrations**, **playbooks**, and **automations** are supported content-pack items. For example, a pack can include an integration that retrieves data, an automation that transforms its results, and a playbook that orchestrates both.

Server log bundles and individual fetched incident records are operational data, not distributable content-pack objects.

QUESTION NO: 15

Whar are possible war room result (entry) types?

A. Context, file, error, image

B. Note, indicator, error, image

C. Video, file, error, image

D. Note, file, error, image

ANSWER: B

Explanation:

Note, indicator, error, image is correct because Cortex XSOAR can create War Room entries that present human-readable notes, indicator-related results, error messages, and rendered images. These entry types enable an automation, integration

command, or playbook task to communicate meaningful operational results directly in the investigation timeline. A note is used for readable analyst-facing output, while an indicator entry represents indicator-related data that can be investigated and enriched. Error entries communicate execution failures or processing issues in a form visible to the analyst. Image entries allow visual artifacts, such as charts, screenshots, or generated visual output, to be displayed in the War Room. XSOAR's result-handling mechanisms support returning structured command results and selecting the appropriate entry type so that returned data is displayed and handled correctly in the investigation. This makes entry typing important both for analyst usability and for ensuring that content is represented appropriately in the War Room. See the Cortex XSOAR context and outputs reference and the CommonServerPython API reference for result and entry-return implementation guidance.

QUESTION NO: 16 - (SIMULATION)

Arrange these steps in the order that they occur during an incident fetch.

ANSWER: See the explanation for the answer

Explanation:

Correct incident-fetch order:

1. The **integration fetches** the incident data from the external source.
2. **Pre-process rules** are evaluated and applied (for example, to drop, tag, or modify an incoming incident before it is created).
3. The incident **classification** is applied to determine the incident type.
4. The applicable **mapping** is applied to map source fields into XSOAR incident fields.
5. The **incident is created** in Cortex XSOAR.

In short: `Integration fetch → Pre-process rules → Classification → Mapping → Incident creation.`

QUESTION NO: 17

Which two functions in XSOAR are incident types used for? (Choose two.)

A. To run dedicated playbooks for different event types

B. To classify events ingested from various sources into the relevant types

C. To classify indicators extracted in XSOAR incidents to their respective types

D. To facilitate role based access to XSOAR incidents

ANSWER: A B

Explanation:

Incident types provide the classification layer that XSOAR applies to incoming events. Integrations and classifiers use source data to identify the appropriate incident type, ensuring that alerts from different products or event categories are represented consistently as the relevant incident type in the platform. This classification supports the correct handling of incidents from ingestion onward.

An incident type can also be configured with a default playbook. Consequently, assigning an incident to a particular type enables XSOAR to use the dedicated workflow designed for that kind of event, such as phishing, malware, or an access-related alert. This lets teams apply purpose-built investigation, enrichment, containment, and response steps according to the nature of the incident. Incident types may additionally be associated with layouts and fields, helping present the information and workflow appropriate to that event category.

Palo Alto Networks documents incident types as the mechanism used to categorize incidents and associate them with playbooks and related incident configuration. See the Cortex XSOAR Incident Types documentation and the Classifiers and Mappers documentation.

QUESTION NO: 18

If a known malicious domain is no longer associated with a specific IP address, which action will make the association inactive?.

A. Revoke the relationship.

B. Update the relationship type.

C. Expire the IP address indicator.

D. Update the indicator relationship description.

ANSWER: A

Explanation:

Revoke the relationship. is the correct action because Cortex XSOAR models an association between two indicators, such as a domain and an IP address, as an indicator relationship with its own lifecycle and state. When intelligence shows that the domain no longer resolves to, communicates with, or is otherwise associated with that specific IP address, revoking the relationship changes the relationship's state to inactive. This accurately records that the association was once observed but is no longer current.

Keeping the relationship record while marking it inactive is important for threat-intelligence operations. Historical domain-to-IP mappings can remain valuable during incident investigations, timeline analysis, hunting, correlation, and audit review. Revocation therefore preserves the prior intelligence context without allowing the relationship to continue to represent an active connection between the indicators. The domain and IP indicators themselves remain available for their own independent reputation, metadata, expiration, and lifecycle management.

This use of relationships enables XSOAR to maintain structured intelligence that distinguishes active observations from historical ones, rather than requiring analysts to remove prior evidence. See the Cortex XSOAR documentation for indicator relationships and the Cortex documentation portal.

QUESTION NO: 19

Which configuration is a valid distributed database (DB) implementation?

A. 2 main DBs, 1 application server, 2 node servers

B. 1 main DB, 1 application server, 3 node servers

C. 2 application servers, 1 main DB, 1 node server

D. 1 application server, 2 main DBs, 1 node server

ANSWER: B

Explanation:

1 main DB, 1 application server, 3 node servers is a valid distributed DB implementation because Cortex XSOAR's distributed architecture separates the application-service role from the database tier and requires a database cluster that can maintain availability and quorum. The main DB coordinates the database deployment, while the three node servers provide the minimum cluster membership needed to tolerate a single node failure without losing quorum. The application server is deployed separately to run the Cortex XSOAR application services and communicate with the database tier. This topology therefore provides the required functional roles—one application server, one main database component, and three database nodes—in a supported distributed layout. Palo Alto Networks documents the distributed deployment model and its database-node requirements in the Cortex XSOAR Administrator Guide. See the Cortex XSOAR documentation portal and the Cortex documentation portal for the applicable version's distributed architecture and installation guidance.

QUESTION NO: 20

In order to automatically run a playbook on the indicators fetched by an integration, what would an XSOAR Administrator setup?

A. Cron job

B. Time triggered job

C. Feed triggered job

D. REST API job

ANSWER: C

Explanation:

Feed triggered job is correct because it is specifically designed to launch an XSOAR playbook when a feed integration fetches new indicators. The administrator configures the job with the relevant feed and selects the playbook to run. When the feed retrieves indicators, XSOAR triggers the configured playbook and supplies the fetched indicator data as the job's input context. This supports automated enrichment, scoring, tagging, reputation checks, or any other indicator-processing workflow immediately after indicator ingestion.

Feed-triggered jobs are therefore event-driven by the arrival of indicator data from an integration, rather than requiring an analyst to manually initiate a playbook. This is useful for threat-intelligence operations where newly received indicators should consistently pass through the same processing and enrichment pipeline. The job can also be configured so that the playbook receives the indicators fetched during the feed run, allowing the playbook to act directly on the relevant indicator set. Palo Alto Networks documents jobs as a mechanism for automating playbook execution and includes feed-triggered execution for processing indicators fetched by feeds. See the Cortex XSOAR Jobs documentation and the Indicator Feeds documentation.

QUESTION NO: 21

An engineer must create a playbook task which asks a user a single question to determine the next step in the playbook flow.

Which type of task will accomplish this goal?.

A. Standard task using manual task settings.

B. Data collection task using the task option.

C. Conditional task using the ask option.

D. Data collection task using the generated link option.

ANSWER: C

Explanation:

Conditional task using the ask option is correct because it is designed to obtain a user decision and use that decision to control which path the playbook follows next. In Cortex XSOAR, a conditional task evaluates outcomes to determine routing; when configured with the Ask capability, it can present a question to an analyst and use the selected response as the condition that drives the subsequent branch. This is useful when automation cannot safely make a decision on its own and an analyst must choose the appropriate investigation, remediation, escalation, or closure path. The task therefore combines the user interaction and flow-control requirements in one playbook element: the user supplies the decision, and the conditional task routes execution according to that result. Configure clear response choices that correspond to the outgoing task branches so the playbook remains predictable, auditable, and easy for analysts to operate. Palo Alto Networks describes conditional tasks and their role in directing playbook execution in the Cortex XSOAR Administrator Guide. Additional playbook-design guidance is available in the Cortex XSOAR Playbooks documentation.

QUESTION NO: 22

An engineer would like to present a trend using widgets to compare to a previous week's data. Which two methods will allow the engineer to meet the requirement? (Choose two.)

A. Create widget of type Line, check 'Display Trend' and define as 7 days ago

B. Create a custom widget using a new incident query

C. Create widget of type Number, check 'Display Trend' and define as 7 days ago

D. Create a custom widget using a script

ANSWER: C D

Explanation:

A Number widget can display a calculated incident metric and its relative trend. Enabling *Display Trend* and configuring the comparison interval as seven days ago provides a direct comparison between the current value and the corresponding value from the prior week. This is appropriate when the engineer needs a concise dashboard indicator, such as the change in incident volume, SLA breaches, or a query-based count over the selected timeframe. The trend display supplies the visual direction and comparison without requiring the dashboard viewer to calculate the difference manually.

A custom widget implemented with a script also meets the requirement because dashboard scripts can retrieve and process incident data, calculate values for both the current and prior-week periods, and return the result in the format required by the widget. This approach is useful where the comparison requires logic beyond a standard incident query, such as custom filtering, derived metrics, normalization, or a specialized visualization. Palo Alto Networks documents dashboard-widget configuration and the supported widget model in the Cortex XSOAR Widgets reference; script development and execution capabilities are described in the CommonServerPython API reference.

QUESTION NO: 23

An engineer is adding diagnostic logging to an automation while investigating inconsistent responses from an external API. The automation should continue execution after writing the diagnostic information.

Which two functions can be used for this purpose? (Choose two.)

A. `demisto.debug()`

B. `demisto.error()`

C. `return_error()`

D. `return_results()`

ANSWER: A B

Explanation:

`demisto.debug()` writes diagnostic-level information to the XSOAR logs, which is useful for recording request flow, parsed values, and decision paths during troubleshooting. **`demisto.error()`** writes error-level information to the logs and is appropriate for recording an unexpected condition while allowing the automation to handle that condition as designed.

`return_error()` is used when the automation should return a failed result to XSOAR and terminate the normal execution path. `return_results()` returns command output to the War Room and context; it is not a diagnostic logging function.