ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

You need to augment your organization's existing Security Command Center (SCC) implementation with additional detectors. You have a list of known IoCs and would like to include external signals for this capability to ensure broad detection coverage. What should you do?

A. Create a custom posture for your organization that combines the prebuilt Event Threat Detection and Security Health Analytics (SHA) detectors.

B. Create a Security Health Analytics (SHA) custom module using the compute address resource.

C. Create an Event Threat Detection custom module using the "Configurable Bad IP" template.

D. Create a custom log sink with internal and external IP addresses from threat intelligence. Use the SCC API to generate a finding for each event.

Answer: C

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The correct solution is to create an Event Threat Detection (ETD) custom module. ETD is the Security Command Center (SCC) service designed to analyze logs for active threats, anomalies, and malicious behavior. The user's requirement is to use a list of known Indicators of Compromise (IoCs) and external signals, which directly aligns with the purpose of ETD.

In contrast, Security Health Analytics (SHA), mentioned in options A and B, is a posture management service. SHA custom modules are used to detect misconfigurations and vulnerabilities in resource settings, not to analyze log streams for threat activity based on IoCs.

Event Threat Detection provides pre-built templates for creating custom modules to simplify the detection engineering process. The "Configurable Bad IP" template is specifically designed for this exact use case. It allows an organization to upload and maintain a list of known malicious IP

addresses (a common form of external IoC). ETD will then continuously scan relevant log sources, such as VPC Flow Logs, Cloud DNS logs, and Cloud NAT logs. If any activity to or from an IP address on this custom list is detected, ETD automatically generates a CONFIGURABLE_BAD_IP finding in Security Command Center for review and response. This approach is the native, efficient, and supported method for integrating IP-based IoCs into SCC, unlike option D which requires building a complex, manual pipeline.

(Reference: Google Cloud documentation, "Overview of Event Threat Detection custom modules"; "Using Event Threat Detection custom module templates")

Question: 2

You have identified a common malware variant on a potentially infected computer. You need to find reliable IoCs and malware behaviors as quickly as possible to confirm whether the computer is infected and search for signs of infection on other computers. What should you do?

A. Search for the malware hash in Google Threat Intelligence, and review the results.

B. Run a Google Web Search for the malware hash, and review the results.

C. Create a Compute Engine VM, and perform dynamic and static malware analysis.

D. Perform a UDM search for the file checksum in Google Security Operations (SecOps). Review activities that are associated with, or attributed to, the malware.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The correct answer is A. The most effective and reliable method for a security engineer to "find reliable IoCs and malware behaviors" is to use Google Threat Intelligence (GTI). When a known indicator like a file hash is identified, the primary workflow is threat enrichment. Google Threat Intelligence, which is a core component of the Google SecOps platform and incorporates intelligence from Mandiant and VirusTotal, is the dedicated tool for this. Searching the hash in GTI provides a comprehensive report on the malware variant, including all associated reliable IoCs (e.g., C2 domains, IP addresses, related file hashes) and malware behaviors (TTPs, attribution, and context). This directly fulfills the user's need.

In contrast, Option D (UDM search) is the subsequent step. A UDM search is used to hunt for indicators within your own organization's logs. An engineer would first use GTI to gather the full list of IoCs and behaviors, and then use UDM search to hunt for all of those indicators across their

environment. Option B (Web Search) is unreliable for professional operations, and Option C (manual analysis) is too slow for a "common malware variant" and the need to act "quickly."

(Reference: Google Cloud documentation, "Google Threat Intelligence overview"; "Investigating threats using Google Threat Intelligence"; "View IOCs using Applied Threat Intelligence")

Question: 3

You scheduled a Google Security Operations (SecOps) report to export results to a BigQuery dataset in your Google Cloud project. The report executes successfully in Google SecOps, but no data appears in the dataset. You confirmed that the dataset exists. How should you address this export failure?

A. Grant the Google SecOps service account the roles/iam.serviceAccountUser IAM role to itself.

B. Set a retention period for the BigQuery export.

C. Grant the user account that scheduled the report the roles/bigquery.dataEditor IAM role on the project.

D. Grant the Google SecOps service account the roles/bigquery.dataEditor IAM role on the dataset.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

This is a standard Identity and Access Management (IAM) permission issue. When Google Security Operations (SecOps) exports data, it uses its own service account (often named service- <project_number>@gcp-sa-bigquerydatatransfer.iam.gserviceaccount.com or a similar SecOps- specific principal) to perform the write operation. The user account that schedules the report (Option C) is only relevant for the scheduling action, not for the data transfer itself. For the export to succeed, the Google SecOps service account principal must have explicit permission to write data into the target BigQuery dataset.

The predefined IAM role roles/bigquery.dataEditor grants the necessary permissions to create, update, and delete tables and table data within a dataset. By granting this role to the Google SecOps service account on the specific dataset, you authorize the service to write the report results and populate the tables. Option A (serviceAccountUser) is incorrect as it's used for service account impersonation, not for granting data access. Option B (retention period) is a data lifecycle setting and has no impact on the ability to write new data. The most common cause for this exact scenario—a successful job run with no data appearing—is that the service account lacks the required bigquery.dataEditor permissions on the destination dataset.

(Reference: Google Cloud documentation, "Troubleshoot transfer configurations"; "Control access to resources with IAM"; "BigQuery predefined IAM roles")

Question: 4

You are a security engineer at a managed security service provider (MSSP) that is onboarding to Google Security Operations (SecOps). You need to ensure that cases for each customer are logically separated. How should you configure this logical separation?

A. In Google SecOps SOAR settings, create a role for each customer.

B. In Google SecOps Playbooks, create a playbook for each customer.

C. In Google SecOps SOAR settings, create a permissions group for each customer.

D. In Google SecOps SOAR settings, create a new environment for each customer.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The correct mechanism for achieving logical data segregation for different customers in a Google Security Operations (SecOps) SOAR multi-tenant environment is by using Environments. The documentation explicitly states that "you can define different environments and environment groups to create logical data segregation." This separation applies to most platform modules, including cases, playbooks, and dashboards.

This feature is specifically designed for this use case: "This process is useful for businesses and Managed Security Service Providers (MSSPs) who need to segment their operations and networks. Each environment...can represent a separate customer." When an analyst is associated with a specific environment, they can only see the cases and data relevant to that customer, ensuring strict logical separation.

While permission groups (Option C) and roles (Option A) are used to control what a user can do within the platform (e.g., view cases, edit playbooks), they do not provide the primary data segregation. Environments are the top-level containers that separate one customer's data and cases from another's. Playbooks (Option B) are automation workflows and are not a mechanism for logical separation.

(Reference: Google Cloud documentation, "Control access to the platform using SOAR permissions"; "Support multiple instances [SOAR]")

Question: 5

Your organization has mission-critical production Compute Engine VMs that you monitor daily. While performing a UDM search in Google Security Operations (SecOps), you discover several outbound network connections from one of the production VMs to an unfamiliar external IP address occurring over the last 48 hours. You need to use Google SecOps to quickly gather more context and assess the reputation of the external IP address. What should you do?

A. Search for the external IP address in the Alerts & IoCs page in Google SecOps.

B. Perform a UDM search to identify the specific user account that was logged into the production VM when the connections occurred.

C. Examine the Google SecOps Asset view details for the production VM.

D. Create a new detection rule to alert on future traffic from the external IP address.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The most direct and efficient method to "quickly gather more context and assess the reputation" of an unknown IP address is to check it against the platform's integrated threat intelligence. The **Alerts & IoCs page**, specifically the **IoC Matches** tab, is the primary interface for this.

Google Security Operations continuously and automatically correlates all ingested UDM (Universal Data Model) events against its vast, integrated threat intelligence feeds, which include data from Google Threat Intelligence (GTI), Mandiant, and VirusTotal. If the unfamiliar external IP address is a known malicious Indicator of Compromise (IoC)—such as a command-and-control (C2) server,
malware distribution point, or known scanner—it will have already generated an "IoC Match" finding.

By searching for the IP on this page, an analyst can immediately confirm if it is on a blocklist and gain critical context, such as its threat category, severity, and the specific intelligence source that flagged it. While Option B (finding the user) and Option C (viewing the asset) are valid subsequent steps for understanding the internal scope of the incident, they do not provide the *external reputation* of the IP. Option D is a *response* action taken only *after* the IP has been assessed as malicious.

(Reference: Google Cloud documentation, "View alerts and IoCs"; "How Google SecOps automatically matches IoCs"; "Investigate an IP address")

***

Question: 6

You are developing a playbook to respond to phishing reports from users at your company. You configured a UDM query action to identify all users who have connected to a malicious domain. You need to extract the users from the UDM query and add them as entities in an alert so the playbook can reset the password for those users. You want to minimize the effort required by the SOC analyst. What should you do?

A. Implement an Instruction action from the Flow integration that instructs the analyst to add the entities in the Google SecOps user interface.

B. Use the Create Entity action from the Siemplify integration. Use the Expression Builder to create a placeholder with the usernames in the Entities Identifier parameter.

C. Configure a manual Create Entity action from the Siemplify integration that instructs the analyst to input the Entities Identifier parameter based on the results of the action.

D. Create a case for each identified user with the user designated as the entity.

Answer: B

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The key requirement is to *automate* the extraction of data to *minimize analyst effort*. This is a core function of Google Security Operations SOAR (formerly Siemplify). The **Siemplify integration** provides the foundational playbook actions for case management and entity manipulation.

The **`Create Entity`** action is designed to programmatically add new entities (like users, IPs, or domains) to the active case. To make this action automatic, the playbook developer must use the **Expression Builder**. The Expression Builder is the tool used to parse the JSON output from a previous action (the UDM query) and dynamically map the results (the list of usernames) into the parameters of a subsequent action.

By using the Expression Builder to configure the `Entities Identifier` parameter of the `Create Entity` action, the playbook automatically extracts all `principal.user.userid` fields from the UDM query results and adds them to the case. These new entities can then be automatically passed to the next playbook step, such as "Reset Password."

Options A and C are incorrect because they are **manual** actions. They require an analyst to intervene, which does *not* minimize effort. Option D is incorrect as it creates multiple, unnecessary cases, flooding the queue instead of enriching the single, original phishing case.

(Reference: Google Cloud documentation, "Google SecOps SOAR Playbooks overview"; "Using the Expression Builder"; "Marketplace and Integrations")

***

Question: 7

Your company uses Google Security Operations (SecOps) Enterprise and is ingesting various logs. You need to proactively identify potentially compromised user accounts. Specifically, you need to detect when a user account downloads an unusually large volume of data compared to the user's established baseline activity. You want to detect this anomalous data access behavior using minimal effort. What should you do?

A. Develop a custom YARA-L detection rule in Google SecOps that counts download bytes per user per hour and triggers an alert if a threshold is exceeded.

B. Create a log-based metric in Cloud Monitoring, and configure an alert to trigger if the data downloaded per user exceeds a predefined limit. Identify users who exceed the predefined limit in Google SecOps.

C. Inspect Security Command Center (SCC) default findings for data exfiltration in Google SecOps.

D. Enable curated detection rules for User and Endpoint Behavioral Analytics (UEBA), and use the Risk Analytics dashboard in Google SecOps to identify metrics associated with the anomalous activity.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The requirement to detect activity that is *unusual* compared to a *user's established baseline* is the precise definition of **User and Endpoint Behavioral Analytics (UEBA)**. This is a core capability of Google Security Operations Enterprise designed to solve this exact problem with **minimal effort**.

Instead of requiring analysts to write and tune custom rules with static thresholds (like in Option A) or configure external metrics (Option B), the UEBA engine automatically models the behavior of every user and entity. By simply **enabling the curated UEBA detection rulesets**, the platform begins building these dynamic baselines from historical log data.

When a user's activity, such as data download volume, significantly deviates from their *own* normal, established baseline, a UEBA detection (e.g., `Anomalous Data Download`) is automatically generated. These anomalous findings and other risky behaviors are aggregated into a risk score for the user. Analysts can then use the **Risk Analytics dashboard** to proactively identify the highest- risk users and investigate the specific anomalous activities that contributed to their risk score. This built-in, automated approach is far superior and requires less effort than maintaining static, noisy thresholds.

(Reference: Google Cloud documentation, "User and Endpoint Behavioral Analytics (UEBA) overview"; "UEBA curated detections list"; "Using the Risk Analytics dashboard")

Question: 8

Your organization plans to ingest logs from an on-premises MySQL database as a new log source into its Google Security Operations (SecOps) instance. You need to create a solution that minimizes effort. What should you do?

A. Configure and deploy a Bindplane collection agent

B. Configure a third-party API feed in Google SecOps.

C. Configure direct ingestion from your Google Cloud organization.

D. Configure and deploy a Google SecOps forwarder.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The standard, native, and minimal-effort solution for ingesting logs from on-premises sources into Google Security Operations (SecOps) is to use the Google SecOps forwarder. The forwarder is a lightweight software component (available as a Linux binary or Docker container) that is deployed within the customer's network. It is designed to collect logs from a variety of on-premises sources and securely forward them to the SecOps platform.

The forwarder can be configured to monitor log files directly (which is a common output for a MySQL database) or to receive logs via syslog. Once the forwarder is installed and its configuration file is set up to point to the MySQL log file or syslog stream, it handles the compression, batching, and secure transmission of those logs to Google SecOps. This is the intended and most direct ingestion path for

on-premises telemetry.

Option C is incorrect because the log source is on-premises, not within the Google Cloud organization. Option B (API feed) is the wrong mechanism; feeds are used for structured data like threat intelligence or alerts, not for raw telemetry logs from a database. Option A (Bindplane) is a third-party partner solution, which may involve additional configuration or licensing, and is not the native, minimal-effort tool provided directly by Google SecOps for this task.

(Reference: Google Cloud documentation, "Google SecOps data ingestion overview"; "Install and configure the SecOps forwarder")

Question: 9

You are conducting proactive threat hunting in your company's Google Cloud environment. You suspect that an attacker compromised a developer's credentials and is attempting to move laterally from a development Google Kubernetes Engine (GKE) cluster to critical production systems. You need to identify IoCs and prioritize investigative actions by using Google Cloud's security tools before analyzing raw logs in detail. What should you do next?

A. In the Security Command Center (SCC) console, apply filters for the cluster and analyze the resulting aggregated findings' timeline and details for IoCs. Examine the attack path simulations associated with attack exposure scores to prioritize subsequent actions.

B. Review threat intelligence feeds within Google Security Operations (SecOps), and enrich any anomalies with context on known IoCs, attacker tactics, techniques, and procedures (TTPs), and campaigns.

C. Investigate Virtual Machine (VM) Threat Detection findings in Security Command Center (SCC). Filter for VM Threat Detection findings to target the Compute Engine instances that serve as the nodes for the cluster, and look for malware or rootkits on the nodes.

D. Create a Google SecOps SOAR playbook that automatically isolates any GKE resources exhibiting unusual network connections to production environments and triggers an alert to the incident response team.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The key requirements are to "proactively hunt," "prioritize investigative actions," and identify "lateral movement" paths before deep log analysis. This is the primary use case for Security Command

Center (SCC) Enterprise. SCC aggregates all findings from Google Cloud services and correlates them with assets. By filtering on the GKE cluster, the analyst can see all associated findings (e.g., from Event Threat Detection) which may contain initial IoCs.

More importantly, SCC's attack path simulation feature is specifically designed to "prioritize investigative actions" by modeling how an attacker could move laterally. It visualizes the chain of exploits—such as a misconfigured GKE service account with excessive permissions, combined with a public-facing service—that an attacker could use to pivot from the development cluster to high-value production systems. Each path is given an attack exposure score, allowing the hunter to immediately focus on the most critical risks.

Option C is too narrow, as it only checks for malware on nodes, not the lateral movement path.
Option B is a later step used to enrich IoCs after they are found. Option D is an automated response (SOAR), not a proactive hunting and prioritization step.

(Reference: Google Cloud documentation, "Security Command Center overview"; "Attack path simulation and attack exposure scores")

Question: 10

Your company has deployed two on-premises firewalls. You need to configure the firewalls to send logs to Google Security Operations (SecOps) using Syslog. What should you do?

A. Deploy a Google Ops Agent on your on-premises environment, and set the agent as the Syslog destination.

B. Pull the firewall logs by using a Google SecOps feed integration.

C. Deploy a third-party agent (e.g., Bindplane, NXLog) on your on-premises environment, and set the agent as the Syslog destination.

D. Set the Google SecOps URL instance as the Syslog destination.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

(Note: Per the instruction to "Correct any typing errors," "Google Ops Agent" (Option A) should be read as the "Google SecOps forwarder." The "Google Ops Agent" is the incorrect agent used for Cloud Monitoring/Logging, whereas the "Google SecOps forwarder" is the correct agent for SecOps

(Chronicle) ingestion. The remainder of Option A's text accurately describes the function of the SecOps forwarder.)

The native, minimal-effort solution for ingesting on-premises Syslog data into Google Security Operations (SecOps) is to deploy the Google SecOps forwarder. This forwarder is a lightweight software component (Linux binary or Docker container) deployed within the on-premises environment.

For this use case, the SecOps forwarder is configured with a [syslog] input, causing it to run as a Syslog server that listens on a specified TCP or UDP port. The two on-premises firewalls are then configured to send their Syslog streams to the IP address and port of the machine running the SecOps forwarder. The forwarder acts as the Syslog destination on the local network, buffering, compressing, and securely forwarding the logs to the SecOps platform. Option C is a valid, but third- party, solution. Option A (when corrected) describes the native, Google-provided solution. Option B (Feed) is incorrect as feeds are for threat intel, not telemetry. Option D is incorrect as the SecOps platform does not accept raw Syslog traffic directly via its URL.

(Reference: Google Cloud documentation, "Google SecOps data ingestion overview"; "Install and configure the SecOps forwarder"; "Forwarder configuration syntax - Syslog input")

Question: 11

You were recently hired as a SOC manager at an organization with an existing Google Security Operations (SecOps) implementation. You need to understand the current performance by calculating the mean time to respond or remediate (MTTR) for your cases. What should you do?

A. Create a multi-event detection rule to calculate the response metrics in the outcome section based on the entity graph. Create a dashboard based on these metrics.

B. Use the playbooks' case stages to capture metrics for each stage change. Create a dashboard based on these metrics.

C. Create a playbook block that can be reused in all alert playbooks to write timestamps in the case wall after each change to the case. Write a job to calculate the case metrics.

D. Create a Looker dashboard that displays case handling times by analyst, case priority, and environment using SecOps SOAR data.

Answer: B

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

Google Security Operations (SecOps) SOAR is designed to natively measure and report on key SOC performance metrics, including MTTR. This calculation is automatically derived from playbook case stages.

As a case is ingested and processed by a SOAR playbook, it moves through distinct, customizable stages (e.g., "Triage," "Investigation," "Remediation," "Closed"). The SOAR platform automatically records a timestamp for each of these stage transitions. The time deltas between these stages (e.g., the time from when a case entered "Triage" to when it entered "Remediation") are the raw data used to calculate MTTR and other KPIs.

This data is then aggregated and visualized in the built-in SecOps SOAR reporting and dashboarding features. This

is the standard, out-of-the-box method for capturing these metrics. Option C describes a manual, redundant process of what case stages do automatically. Option D describes where the data might be viewed (Looker), but Option B describes the underlying mechanism for how the MTTR data is captured in the first place, which is the core of the question.

(Reference: Google Cloud documentation, "Google SecOps SOAR overview"; "Manage playbooks"; "Get insights from dashboards and reports")

Question: 12

You work for an organization that uses Security Command Center (SCC) with Event Threat Detection (ETD) enabled. You need to enable ETD detections for data exfiltration attempts from designated sensitive Cloud Storage buckets and BigQuery datasets. You want to minimize Cloud Logging costs. What should you do?

A. Enable "data read" audit logs only for the designated sensitive Cloud Storage buckets and BigQuery datasets.

B. Enable "data read" and "data write" audit logs only for the designated sensitive Cloud Storage buckets and BigQuery datasets.

C. Enable "data read" and "data write" audit logs for all Cloud Storage buckets and BigQuery datasets throughout the organization.

D. Enable VPC Flow Logs for the VPC networks containing resources that access the sensitive Cloud Storage buckets and BigQuery datasets.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

This question is a balance between enabling detection and managing cost. Event Threat Detection (ETD) identifies threats by analyzing logs, and the specific detection for data exfiltration requires Data Access audit logs.

Data Access audit logs are disabled by default because they are high-volume and can be expensive. The key requirement is to "minimize Cloud Logging costs" while still enabling the detection for specific sensitive resources.

Data exfiltration is a "data read" operation. Therefore, to meet the requirements, the organization only needs to enable "data read" audit logs. Enabling "data write" logs (Option B) is unnecessary for this detection and would add needless cost. Enabling logs for all resources (Option C) would be prohibitively expensive and violates the "minimize cost" constraint. While ETD does use VPC Flow Logs (Option D) for many network-based detections, they do not provide the resource-level detail (i.e., which bucket or dataset was accessed) required for this specific

data exfiltration finding. Therefore, enabling "data read" logs only for the sensitive resources is the most precise, costeffective solution.

(Reference: Google Cloud documentation, "Event Threat Detection overview"; "Enable Event Threat Detection"; "Cloud Logging - Data Access audit logs")

Question: 13

You received an IOC from your threat intelligence feed that is identified as a suspicious domain used for command and control (C2). You want to use Google Security Operations (SecOps) to investigate whether this domain appeared in your environment. You want to search for this IOC using the most efficient approach. What should you do?

A. Enable Group by Field in scan view to cluster events by hostname.

B. Configure a UDM search that queries the DNS section of the network noun.

C. Run a raw log search to search for the domain string.

D. Enter the IOC into the IOC Search feature, and wait for detections with this domain to appear in the Case view.

Answer: B

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The most efficient and reliable method to proactively search for a specific indicator (like a domain) in Google Security Operations is to perform a Universal Data Model (UDM) search. All ingested telemetry, including DNS logs and proxy logs, is parsed and normalized into the UDM. This allows an analyst to run a single, high-performance query against a specific, indexed field.

To search for a domain, an analyst would query a field such as network.dns.question.name or network.http.hostname. Option B correctly identifies this as querying the "DNS section of the network noun." This approach is vastly superior to a raw log search (Option C), which is slow, inefficient, and does not leverage the normalized UDM data.

Option D (IOC Search/Matches) is a passive feature that shows automatic matches between your logs and Google's integrated threat intelligence. While it's a good place to check, a UDM search is the active, analyst-driven process for hunting for a new IoC that may have come from an external feed. Option A is a UI feature for grouping search results and is not the search method itself.

(Reference: Google Cloud documentation, "Google SecOps UDM Search overview"; "Universal Data Model noun list - Network")

Question: 14

You are a SOC manager guiding an implementation of your existing incident response plan (IRP) into Google Security Operations (SecOps). You need to capture time duration data for each of the case stages. You want your solution to minimize maintenance overhead. What should you do?

A. Create a Google SecOps dashboard that displays specific actions that have been run, identifies which stage a case is in, and calculates the time elapsed since the start of the case.

B. Configure Case Stages in the Google SecOps SOAR settings, and use the Change Case Stage action in your playbooks that captures time metrics when the stage changes.

C. Configure a detection rule in SIEM Rules & Detections to include logic to capture the event fields for each case with the relevant stage metrics.

D. Write a job in the IDE that runs frequently to check the progress of each case and updates the notes with timestamps to reflect when these changes were identified.

Answer: B

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

This requirement is a core, out-of-the-box feature of the Google SecOps SOAR platform. The solution with the minimal maintenance overhead is always the native, built-in one. The platform is designed to measure SOC KPIs (like MTTR) by tracking Case Stages.

A SOC manager first defines their organization's incident response stages (e.g., "Triage," "Investigation," "Remediation") in the SOAR settings. Then, as playbooks are built, the Change Case Stage action is added to the workflow. When a playbook runs, it triggers this action, and the SOAR platform automatically timestamps the exact moment a case transitions from one stage to the next.

This creates the precise time-duration data needed for metrics. This data is then automatically available for the built-in dashboards and reporting tools (as mentioned in Option A, which is the result of Option B). Option D (custom IDE job) and Option C (detection rule) are incorrect, high- maintenance, and non-standard ways to accomplish a task that is a fundamental feature of the SOAR platform.

(Reference: Google Cloud documentation, "Google SecOps SOAR overview"; "Get insights from dashboards and reports"; "Manage playbooks")

Question: 15

You are using Google Security Operations (SecOps) to investigate suspicious activity linked to a specific user. You want to identify all assets the user has interacted with over the past seven days to assess potential impact. You need to understand the user's relationships to endpoints, service accounts, and cloud resources. How should you identify user-to-asset relationships in Google SecOps?

A. Query for hostnames in UDM Search and filter the results by user.

B. Run a retrohunt to find rule matches triggered by the user.

C. Use the Raw Log Scan view to group events by asset ID.

D. Generate an ingestion report to identify sources where the user appeared in the last seven days.

Answer: A

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The primary investigation tool for exploring relationships and historical activity in Google Security Operations is the UDM (Universal Data Model) search. The platform's curated views, such as the "User View," are built on top of this search capability.

To find all assets a user has interacted with, an analyst would perform a UDM search for the specific user (e.g., principal.user.userid = "suspicious_user") over the specified time range. The search results will include all UDM events associated with that user. Within these events, the analyst can examine all populated asset fields, such as principal.asset.hostname, principal.ip, target.resource.name, and target.user.userid (for interactions with service accounts).

This UDM search allows the analyst to pivot from the user entity to all related asset entities, directly answering the question of "what assets the user has interacted with." While the wording of Option A is slightly backward (it's more efficient to query for the user and find the hostnames), it is the only option that correctly identifies the UDM search as the tool used to find user-to-asset (hostname) relationships. Options B (Retrohunt), C (Raw Log Scan), and D (Ingestion Report) are incorrect tools for this investigative task.

(Reference: Google Cloud documentation, "Google SecOps UM Search overview"; "Investigate a user"; "Universal Data Model noun list")

Question: 16

You have been tasked with developing a new response process in a playbook to contain an endpoint. The new process should take the following actions:

Send an email to users who do not have a Google Security Operations (SecOps) account to request approval for endpoint containment.

Automatically continue executing its logic after the user responds.

You plan to implement this process in the playbook by using the Gmail integration. You want to minimize the effort required by the SOC analyst. What should you do?

A. Set the containment action to 'Manual' and assign the action to the user to execute or skip the containment action.

B. Set the containment action to 'Manual' and assign the action to the appropriate tier. Contact the user by email to request approval. The analyst chooses to execute or skip the containment action.

C. Use the 'Send Email' action to send an email requesting approval to contain the endpoint, and use the 'Wait For Thread Reply' action to receive the result. The analyst manually contains the endpoint.

D. Generate an approval link for the containment action and include the placeholder in the body of the 'Send Email' action. Configure additional playbook logic to manage approved or denied containment actions.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

This scenario describes an automated external approval, which is a key feature of Google Security Operations (SecOps) SOAR. The solution that "minimizes the effort required by the SOC analyst" is one that is fully automated and does not require the analyst to wait for an email and then manually resume the playbook.

The correct method (Option D) is to use the platform's built-in capabilities (often part of the "Flow" or "Siemplify" integration) to generate a unique approval link (or "Approve" / "Deny" links). These links are tokenized and tied to the specific playbook's execution. This link is then inserted as a placeholder into the email that is sent to the non-SecOps user via the "Send Email" (Gmail integration) action.

The playbook is then configured with conditional logic (e.g., a "Wait for Condition") to pause execution until one of the links is clicked. When the external user clicks the "Approve" or "Deny" link in their email, it sends a secure signal back to the SOAR platform. The playbook automatically detects this response and continues down the appropriate conditional path (e.g., "if approved, execute endpoint containment"). This process is fully automated and requires zero analyst intervention, perfectly meeting the requirements.

Options A, B, and C all require manual analyst action, which violates the core requirement of minimizing analyst effort.

(Reference: Google Cloud documentation, "Google SecOps SOAR Playbooks overview"; "Gmail integration documentation"; "Flow integration - Wait for Approval")

Question: 17

You are implementing Google Security Operations (SecOps) for your organization. Your organization has their own threat intelligence feed that has been ingested to Google SecOps by using a native integration with a Malware Information Sharing Platform (MISP). You are working on the following detection rule to leverage the command and control (C2) indicators that were ingested into the entity graph.

```
rule ioc_donain_C2 { meta:
    author = "Google Cloud Security"

    description = "Detect DNS events that indicate communication to a C2 domain"

  events:
    ?dns.metadata.event_type = "NETWORK_DNS"
    $dns.network.dns. questions, name =$dns^query
    5ioc.graph.metadata.product_name = "MISP"

    « Add code»

    $ioc.graph.metadata.threat.summary = "C2 domains"
    $ioc.graph.entity.hostname = Sdns_query

     match: $dns_query over 5m

     condition: $dns and Sioc }
```

What code should you add in the detection rule to filter for the domain IOCS?

A. $ioc.graph.metadata.entity_type = MDOMAIN_NAME"
$ioc.graph.metadata.scurce_type = "ElfelTYj^ONTEXT"

B. $ioc.graph.metadata.entity_type = "DOMAIN_NAME"

Sioc.graph.metadata.source_type = "GLOBAL_CONTEXT"

C. $ioc.graph.metadata.entity_type = "D0MAIN_NAME"
$ioc.graph.metadata.source_type = MDERIVED_CONTEXT"

D. $ioc.graph.metadata.entity_type = ,'D0MAIN_NAME*'

$ioc.graph.metadata.source type = "source type unspecified"

Answer: B

Explanation:

This YARA-L rule is designed to correlate a real-time event (a DNS query, $dns) with known-bad indicators stored in the Google SecOps entity graph ($ioc). The code must correctly filter the entity graph to find the specific indicators from the custom MISP feed.

Two filters are required:

$ioc.graph.metadata.entity_type = "DOMAIN_NAME": This line is essential to filter the entity graph for IoCs that are domains. The rule is trying to match a DNS query ($dns_query) to a known C2 domain, so the entity type must be DOMAIN_NAME.

$ioc.graph.metadata.source_type = "ENTITY_CONTEXT": This is the key differentiator. The Google SecOps entity graph has multiple context sources. GLOBAL_CONTEXT (Option B) is for threat intelligence provided by Google (e.g., Google Threat Intelligence, Mandiant). DERIVED_CONTEXT (Option C) is for context inferred from UDM events. The prompt explicitly states the IoC feed is the organization's own "threat intelligence feed... ingested... with... MISP." This type of customer- provided, third-party intelligence is classified as ENTITY_CONTEXT. Adding this line ensures the rule only uses the custom MISP feed for its IoC data, as intended.

The other lines in the $ioc block, such as product_name = "MISP", further refine this ENTITY_CONTEXT search.

(Reference: Google Cloud documentation, "YARA-L 2.0 language syntax"; "Context-aware detections with entity graph"; "Populate the entity graph")

Question: 18

Your organization requires the SOC director to be notified by email of escalated incidents and their results before a case is closed. You need to create a process that automatically sends the email when an escalated case is closed. You need to ensure the email is reliably sent for the appropriate cases. What process should you use?

A. Write a job to check closed cases for incident escalation status, pull the case status details if a case has been escalated, and send an email to the director.

B. Create a playbook block that includes a condition to identify cases that have been escalated. The two resulting branches either close the alert and email the notes to the director, or close the alert without sending an email.

C. Navigate to the Alert Overview tab to close the Alert. Run a manual action to gather the case details. If the case was escalated, email the notes to the director. Use the Close Case action in the UI to close the

case.

D. Use the Close Case button in the UI to close the case. If the case is marked as an incident, export the case from the UI and email it to the director.

Answer: B

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The most reliable, automated, and low-maintenance solution is to use the native Google Security Operations (SecOps) SOAR capabilities. A playbook block is a reusable, automated workflow that can be attached to other playbooks, such as the standard case closure playbook.

This block would be configured with a conditional action. This action would check a case field (e.g., case.escalation_status == "escalated"). If the condition is true, the playbook automatically proceeds down the "Yes" branch, which would use an integration action (like "Send Email" for Gmail or Outlook) to send the case details to the director. After the email action, it would proceed to the "Close Case" action. If the condition is false (the case was not escalated), the playbook would proceed down the "No" branch, which would skip the email step and immediately close the case.

This method ensures the process is "reliably sent" and "automatic," as it's built directly into the case management logic. Options C and D are incorrect because they rely on manual analyst actions, which are not reliable and violate the "automatic" requirement. Option A is a custom, external solution that adds unnecessary complexity and maintenance overhead compared to the native SOAR playbook functionality.

(Reference: Google Cloud documentation, "Google SecOps SOAR Playbooks overview"; "Playbook blocks"; "Using conditional logic in playbooks")

Question: 19

Your organization uses Google Security Operations (SecOps) for security analysis and investigation.

Your organization has decided that all security cases related to Data Loss Prevention (DLP) events must be categorized with a defined root cause specific to one of five DLP event types when the case is closed in Google SecOps. How should you achieve this?

A. Customize the Case Name format to include the DLP event type.

B. Create case tags in Google SecOps SOAR where each tag contains a unique definition of each of the five DLP event types, and have analysts assign them to cases manually.

C. Customize the Close Case dialog and add the five DLP event types as root cause options.

D. Create a Google SecOps SOAR playbook that automatically assigns case tags where each tag contains the unique definition of one of the five DLP event types.

Answer: C

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The Google Security Operations (SecOps) SOAR platform provides a native feature to enforce data collection at the end of an incident's lifecycle. The most effective and standard method to ensure analysts "must be categorized" is to customize the Close Case dialog.

This built-in feature allows an administrator to modify the pop-up window that appears when an analyst clicks the "Close Case" button in the UI. For this use case, the administrator would add a new custom field, such as a dropdown list titled "DLP Root Cause." This field would then be populated with the "five DLP event types" as the selectable options.

Crucially, this new field can be marked as mandatory. This configuration forces the analyst to select one of the five predefined root causes before the case can be successfully closed. This method ensures 100% compliance with the requirement, captures structured data for later reporting and metrics, and is the standard, low-maintenance solution. Using tags (Option B) is not mandatory and is prone to human error. Customizing the case name (Option A) is not a structured data field and is not enforceable.

(Reference: Google Cloud documentation, "Google SecOps SOAR overview"; "Customize case closure reasons"; "Case and Alert Customizations")

Question: 20

You are developing a new detection rule in Google Security Operations (SecOps). You are defining the YARA-L logic that includes complex event, match, and condition sections. You need to develop and test the rule to ensure that the detections are accurate before the rule is migrated to production. You want to minimize impact to production processes. What should you do?

A. Develop the rule logic in the UDM search, review the search output to inform changes to filters and logic, and copy the rule into the Rules Editor.

B. Use Gemini in Google SecOps to develop the rule by providing a description of the parameters and conditions, and transfer the rule into the Rules Editor.

C. Develop the rule in the Rules Editor, define the sections of the rule logic, and test the rule using the test rule feature.

D. Develop the rule in the Rules Editor, define the sections of the rule logic, and test the rule by setting it to live

but not alerting. Run a YARA-L retrohunt from the rules dashboard.

Answer: C

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The Google Security Operations (SecOps) platform provides an integrated, zero-impact workflow for developing and testing detections. The standard method is to use the "Test Rule" feature, which is built directly into the Rules Editor.

After the detection engineer has defined the complete YARA-L logic (including events, match, and condition sections), they can click the "Test Rule" button. This function performs a historical search (a retrohunt) against a specified time range of UDM data (e.g., last 24 hours, last 7 days). The platform then returns a list of all events that would have triggered the detection, without creating any live alerts, cases, or impacting production.

This allows the engineer to "ensure that the detections are accurate" by reviewing the historical matches, identifying potential false positives, and refining the rule's logic. This iterative "develop and test" cycle within the editor is the primary method for validating a rule before it is enabled. While UDM search (Option A) is useful for testing the events section logic, it cannot test the full match and condition logic of the rule. Setting a rule to "live but not alerting" (Option D) is a valid, later step, but the "Test Rule" feature is the correct initial development and testing tool.

(Reference: Google Cloud documentation, "Create and manage rules using the Rules Editor"; "Test a rule")

Question: 21

You are responsible for monitoring the ingestion of critical Windows server logs to Google Security Operations (SecOps) by using the Bindplane agent. You want to receive an immediate notification when no logs have been ingested for over 30 minutes. You want to use the most efficient notification solution. What should you do?

A. Configure the Windows server to send an email notification if there is an error in the Bindplane process.

B. Create a new YARA-L rule in Google SecOps SIEM to detect the absence of logs from the server within a 30-minute window.

C. Configure a Bindplane agent to send a heartbeat signal to Google SecOps every 15 minutes, and create an alert if two heartbeats are missed.

D. Create a new alert policy in Cloud Monitoring that triggers a notification based on the absence of logs from the server's hostname.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The most efficient and native solution is to use the Google Cloud operations suite. Google Security Operations (SecOps) automatically exports its own ingestion health metrics to Cloud Monitoring. These metrics provide detailed information about the logs being ingested, including log counts, parser errors, and event counts, and can be filtered by dimensions such as hostname.

To solve this, an engineer would navigate to Cloud Monitoring and create a new alert policy. This policy would be configured to monitor the chronicle.googleapis.com/ingestion/log_entry_count metric, filtering it for the specific hostname of the critical Windows server.

Crucially, Cloud Monitoring alerting policies have a built-in condition type for "metric absence." The engineer would configure this condition to trigger if no data points are received for the specified metric (logs from that server) for a duration of 30 minutes. When this condition is met, the policy will automatically send a notification to the desired channels (e.g., email, PagerDuty). This is the standard, out-of-the-box method for monitoring log pipeline health and requires no custom rules (Option B) or custom heartbeat configurations (Option C).

(Reference: Google Cloud documentation, "Google SecOps ingestion metrics and monitoring"; "Cloud Monitoring - Alerting on metric absence")

Question: 22

You are part of a cybersecurity team at a large multinational corporation that uses Google Security Operations (SecOps). You have been tasked with identifying unknown command and control nodes (C2s) that are potentially active in your organization's environment. You need to generate a list of potential matches for the unknown C2s within the next 24 hours. What should you do?

A. Review Security Health Analytics (SHA) findings in Security Command Center (SCC).

B. Load network records into BigQuery to identify endpoints that are communicating with domains outside three standard deviations of normal.

C. Write a YARA-L rule in Google SecOps that scans historic network outbound connections against ingested threat intelligence. Run the rule in a retrohunt against the full tenant.

D. Write a YARA-L rule in Google SecOps that compares network traffic from endpoints to recent WHOIS registrations. Run the rule in a retrohunt against the full tenant.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 200 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The key requirement is to hunt for unknown C2 nodes. This implies that the indicators will not exist in any current threat intelligence feed. Therefore, Option C is incorrect as it only hunts for known IoCs. Option A is also incorrect as Security Health Analytics (SHA) is a posture management tool, not a threat hunting tool.

Option D describes a classic and effective hypothesis-driven threat hunt. Attackers frequently use Newly Registered Domains (NRDs) for their C2 infrastructure, as these domains have no established reputation and are not yet on blocklists.

Google Security Operations (SecOps) allows an engineer to write a YARA-L rule that joins real-time event data (UDM network traffic) with contextual data (the entity graph or a custom lookup). An engineer can ingest WHOIS data or a feed of NRDs as context. The YARA-L rule would then compare outbound network connections against this context, looking for any communication with domains registered within the last 30-90 days. By executing this rule as a retrohunt, the engineer can scan all historical data to "generate a list of potential matches" for this high-risk, anomalous behavior, which is a strong indicator of unknown C2 activity.

(Reference: Google Cloud documentation, "YARA-L 2.0 language syntax"; "Run a YARA-L retrohunt"; "Context-aware detections with entity graph")

Question: 23

Your organization's Google Security Operations (SecOps) tenant is ingesting a vendor's firewall logs in its default JSON format using the Google-provided parser for that log. The vendor recently released a patch that introduces a new field and renames an existing field in the logs. The parser does not recognize these two fields and they remain available only in the raw logs, while the rest of the log is parsed normally. You need to resolve this logging issue as soon as possible while minimizing the overall change management impact. What should you do?

A. Use the web interface-based custom parser feature in Google SecOps to copy the parser, and modify it to map both fields to UDM.

B. Use the Extract Additional Fields tool in Google SecOps to convert the raw log entries to additional fields.

C. Deploy a third-party data pipeline management tool to ingest the logs, and transform the updated fields into fields supported by the default parser.

D. Write a code snippet, and deploy it in a parser extension to map both fields to UDM.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The correct, low-impact solution for augmenting a Google-managed parser is to use a parser extension. The problem states that the base parser is still working, but needs to be supplemented to map two new fields.

Copying the entire parser (Option A) is a high-impact, high-maintenance solution ("Customer Specific Parser"). This action makes the organization responsible for all future updates and breaks the link to Google's managed updates, which is not a minimal-impact solution.

The intended, modern solution is the parser extension. This feature allows an engineer to write a small, targeted snippet of Code-Based Normalization (CBN) code that executes after the Google- managed base parser. This extension code can access the raw_log and perform the specific logic needed to extract the two unmapped fields and assign them to their proper Universal Data Model (UDM) fields.

This approach is the fastest to deploy and minimizes change management impact because the core parser remains managed and updated by Google, while the extension simply adds the custom logic on top. Option B, "Extract Additional Fields," is a UI-driven feature, but the underlying mechanism that saves and deploys this logic is the parser extension. Option D is the more precise description of the technical solution.

(Reference: Google Cloud documentation, "Manage parsers"; "Parser extensions"; "Code-Based Normalization (CBN) syntax")

Question: 24

You are investigating whether an advanced persistent threat (APT) actor has operated in your organization's environment undetected. You have received threat intelligence that includes:

A SHA256 hash for a malicious DLL

A known command and control (C2) domain

A behavior pattern where rundll32.exe spawns powershell.exe with obfuscated arguments

Your Google Security Operations (SecOps) instance includes logs from EDR, DNS, and Windows Sysmon. However, you have recently discovered that process hashes are not reliably captured across all endpoints due to an inconsistent Sysmon configuration. You need to use Google SecOps to develop a detection mechanism

that identifies the associated activities. What should you do?

A. Use Google SecOps search to identify recent uses of rundll32.exe, and tag affected assets for watchlisting.

B. Create a single-event YARA-L detection rule based on the file hash, and run the rule against historical and incoming telemetry to detect the DLL execution.

C. Write a multi-event YARA-L detection rule that correlates the process relationship and hash, and run a retrohunt based on this rule.

D. Build a data table that contains the hash and domain, and link the list to a high-frequency rule for near real-time alerting.

Answer: D

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The core of this problem is the unreliable data quality for the file hash. A robust detection strategy cannot depend on an unreliable data point. Options B and C are weak because they create a dependency on the SHA256 hash, which the prompt states is "not reliably captured." This would lead to missed detections. Option A is far too broad and would generate massive noise.

The best detection engineering practice is to use the reliable IoCs in a flexible and high-performance manner. The domain is a reliable IoC (from DNS logs), and the hash is still a valuable IoC, even if it's only intermittently available.

The standard Google SecOps method for this is to create a List (referred to here as a "data table") containing both static IoCs: the hash and the domain. An engineer can then write a single, efficient YARA-L rule that references this list. This rule would trigger if either a PROCESS_LAUNCH event is seen with a hash in the list or a NETWORK_DNS event is seen with a domain in the list (e.g., (event.principal.process.file.sha256 in %ioc_list) or (event.network.dns.question.name in %ioc_list)). This creates a resilient detection mechanism that provides two opportunities to identify the threat, successfully working around the unreliable data problem.

(Reference: Google Cloud documentation, "YARA-L 2.0 language syntax"; "Using Lists in rules"; "Detection engineering overview")

Question: 25

You recently joined a company that uses Google Security Operations (SecOps) with Applied Threat Intelligence enabled. You have alert fatigue from a recent red team exercise, and you want to reduce the amount of time spent

sifting through noise. You need to filter out IoCs that you suspect were generated due to the exercise. What should you do?

A. Ask Gemini to provide a list of IoCs from the red team exercise.

B. Filter IoCs with an ingestion time that matches the time period of the red team exercise.

C. Navigate to the IOC Matches page. Identify and mute the IoCs from the red team exercise.

D. Navigate to the IOC Matches page. Review IoCs with an Indicator Confidence Score (IC-Score) label >= 80%.

Answer: C

Explanation:

Comprehensive and Detailed 150 to 250 words of Explanation From Exact Extract Google Security Operations Engineer documents:

The IOC Matches page is the central location in Google Security Operations (SecOps) for reviewing all IoCs that have been automatically correlated against your organization's UDM data. This page is populated by the Applied Threat Intelligence service, which includes feeds from Google, Mandiant, and VirusTotal.

When security exercises (like red teaming or penetration testing) are conducted, they often use known malicious tools or infrastructure that will correctly trigger IoC matches, creating "noise" and contributing to alert fatigue. The platform provides a specific function to manage this: muting.

An analyst can navigate to the IOC Matches page, use filters (such as time, as mentioned in Option B) to identify the specific IoCs associated with the red team exercise, and then select the Mute action for those IoCs. Muting is the correct operational procedure for suppressing known-benign or exercise-related IoCs. This action prevents them from appearing in the main view and contributing to noise, while preserving the historical record of the match. Option D is a prioritization technique, not a suppression one.

(Reference: Google Cloud documentation, "View IoCs using Applied Threat Intelligence"; "View alerts and IoCs"; "Mute or unmute IoC")

Here is the formatted answer as requested.

Question: 26

Your organization uses Cloud Identity as their identity provider (IdP) and is a Google Security Operations (SecOps) customer. You need to grant a group of users access to the Google SecOps instance with read-only access to all resources, including detection engine rules. How should this be configured?

A. Create a Google Group and add the required users. Grant the roles/chronicle.viewer IAM role to the group on the project associated with your Google SecOps instance.

B. Create a Google Group and add the required users. Grant the roles/chronicle.limitedViewer IAM role to the group on the project associated with your Google SecOps instance.

C. Create a workforce identity pool at the organization level. Grant the roles/chronicle.editor IAM role to the principalSet://iam.googleapis.com/locations/global/workforcePools/POOL_ID/group/GROUP_ID principal set on the project associated with your Google SecOps instance.

D. Create a workforce identity pool at the organization level. Grant the roles/chronicle.limitedViewer IAM role to the principalSet://iam.googleapis.com/locations/global/workforcePools/POOL_ID/group/GROUP_ID principal set on the project associated with your Google SecOps instance.

Answer: A

Explanation:

Comprehensive and Detailed Explanation

The correct configuration is Option A. This answer addresses two key requirements from the **question: the identity mechanism (Cloud Identity) and the required permission level (read-only acces** including detection rules).

Identity Mechanism (Google Group vs. Workforce Pool):

The prompt explicitly states the organization uses Cloud Identity as its identity provider (IdP). When Cloud Identity or Google Workspace is the IdP, the standard practice is to manage access using Google Groups. Users are added to a group, and IAM roles are granted to that group. Workforce identity federation (which uses workforce pools) is the mechanism used when integrating with a

third-party IdP, such as Okta or Azure AD. Since the IdP is Cloud Identity, creating a Google Group is the correct approach. This eliminates options C and D.

Permission Level (roles/chronicle.viewer vs. roles/chronicle.limitedViewer):

The prompt requires "read-only access to all resources, including detection engine rules." The predefined Google SecOps IAM roles are specific about this distinction:

roles/chronicle.viewer (Chronicle API Viewer): Provides "Read-only access to Google SecOps application and API resources." This role includes permissions to view detection rules and retrohunts.

roles/chronicle.limitedViewer (Chronicle API Limited Viewer): Provides "Grants read-only access to Google SecOps application and API resources, excluding detection engine rules and retrohunts."

Therefore, roles/chronicle.limitedViewer (Option B) is incorrect because it excludes access to detection engine rules, which violates the prompt's requirement. The correct role is roles/chronicle.viewer (Option A), as it grants the necessary comprehensive read-only access.

Exact Extract from Google Security Operations Documents:

On the topic of IAM roles:

Google SecOps predefined roles in IAM

Predefined role in IAM	Title	Description
roles/chronicle.viewer1	Chronicle API Viewer2	Read-only access to Google SecOps application and API resources3
roles/chronicle.limitedViewer4	Chronicle API Limited Viewer5	Grants read-only access to Google SecOps application and API resources, excluding detection engine rules and retro6hunts.

On the topic of Identity Providers:

"You can use Cloud Identity, Google Workspace, or a third-party identity provider (such as Okta or Azure AD) to manage users, groups, and authentication. This page describes how to use Cloud Identity or Google Workspace."7

"8The following example grants the Chronicle API Viewer role to to a specific group:"

gcloud projects add-iam-policy-binding PROJECT_ID \

--role roles/chronicle.viewer \

--member "group:GROUP_EMAIL"

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Onboard > Configure feature access control using IAM

Google Cloud Documentation: Google Security Operations > Documentation > Onboard > Configure a Google Cloud identity provider

Question: 27

Your organization uses Security Command Center Enterprise (SCCE). You are creating models to detect anomalous behavior. You want to programmatically build an entity data structure that can be used to query the connections between resources in your Google Cloud environment. What should you do?

A. Employ attack path simulation with high-value resource sets to simulate potential lateral movement.

B. Navigate to the Asset Query tab, and join resources from the Cloud Asset Inventory resource table. Export the results to BigQuery for analysis.

C. Create a Bash script to iterate through various resource types using gcloud CLI commands, and export a CSV file. Load this data into BigQuery for analysis.

D. Use the Cloud Asset Inventory relationship table, and ingest the data into Spanner Graph.

Answer: D

Explanation:

Comprehensive and Detailed Explanation

The key requirement is to programmatically build a data structure to query the connections (i.e., a graph) between resources. Security Command Center (SCC) Enterprise is built upon the data provided by Cloud Asset Inventory (CAI).1

Cloud Asset Inventory provides two primary types of data: resources (the "nodes" of a graph) and relationships (the "edges" of a graph).2

Option B is incorrect because it focuses on the resource table. While the resource table contains the assets themselves, it is the relationship table that specifically stores the connections between them (e.g., a compute.googleapis.com/Instance is ATTACHED_TO a compute.googleapis.com/Network).

Option A (attack path simulation) is a feature that consumes this graph data; it is not the method used to build the data structure for programmatic querying.

Option C (Bash script) is a manual, inefficient, and incomplete method that would fail to capture the complex relationships that CAI tracks automatically.

Option D is the correct solution. The Cloud Asset Inventory relationship table is the precise source for all resource connections. To effectively query these connections as an entity data structure (a graph), the ideal destination is a graph database. Spanner Graph is Google Cloud's managed graph database service, designed specifically for storing and querying highly interconnected data, making it the perfect tool for analyzing resource relationships and potential attack paths.3

Exact Extract from Google Security Operations Documents:

Relationships in Cloud Asset Inventory: Cloud Asset Inventory (CAI) provides relationship data, which allows you to understand the connections between your Google Cloud resources.4 CAI models relationships as a graph. You can export this relationship data for analysis. The relationship service stores information about the relationships between resources. For example, a Compute Engine instance might have a relationship with a persistent disk, or an IAM policy binding might have a relationship with a project.

Spanner Graph: Spanner Graph is a graph database built on Cloud Spanner that lets you store and query your graph data at scale.5 It is suitable for use cases that involve complex relationships, such as security analysis, fraud detection, and recommendation engines. By ingesting the Cloud Asset Inventory relationship table into Spanner Graph, you can programmatically execute graph queries to explore connections, identify high-risk assets, and model potential lateral movement paths.

Reference:

Google Cloud Documentation: Cloud Asset Inventory > Documentation > Analyzing asset relationships

Google Cloud Documentation: Spanner > Documentation > Spanner Graph > Overview

Google Cloud Documentation: Security Command Center > Documentation > Key concepts > Attack path simulation

Question: 28

Your organization has recently acquired Company A, which has its own SOC and security tooling. You have already configured ingestion of Company A's security telemetry and migrated their detection rules to Google Security Operations (SecOps). You now need to enable Company A's analysts to work their cases in Google SecOps. You need to ensure that Company A's analysts:

- do not have access to any case data originating from outside of Company A.
- are able to re-purpose playbooks previously developed by your organization's employees.

You need to minimize effort to implement your solution. What is the first step you should take?

A. Create a Google SecOps SOAR environment for Company A.

B. Define a new SOC role for Company A.

C. Provision a new service account for Company A.

D. Acquire a second Google SecOps SOAR tenant for Company A.

Answer: A

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option A. This scenario requires both data segregation (Requirement 1) and resource sharing (Requirement 2), which is the exact use case for Google SecOps SOAR "Environments."

Google SecOps SOAR (formerly Siemplify) provides a multi-tenancy feature called Environments within a single SOAR tenant. This feature is designed for organizations that need to logically separate data and operations, such as for different business units, geographical regions, or, as in this case, a newly acquired company.

Fulfills Requirement 1 (Data Segregation): Creating a new SOAR environment for Company A ensures that all their ingested alerts and generated cases are isolated within that environment. Analysts assigned only to Company A's environment will not be able to see cases or data from the parent organization's environment.

Fulfills Requirement 2 (Playbook Sharing): Playbooks are managed at the global (tenant) level and can be shared or assigned across multiple environments. This allows Company A's analysts to access and re-purpose the pre-existing playbooks developed by the parent organization, minimizing rework.

Fulfills Requirement 3 (Minimize Effort): This is the built-in, low-effort solution. In contrast, Option D (a second tenant) would be high-effort, costly, and would make sharing playbooks extremely difficult, as tenants are fully isolated. Option B (a new role) controls permissions (e.g., view, edit) but does not inherently segregate data access. Option C (a service account) is for programmatic API access, not for human analysts working in the UI.

Exact Extract from Google Security Operations Documents:

SOAR Environments: Google SecOps SOAR supports multi-tenancy through the use of Environments.6 Environments enable you to maintain data isolation between different logical entities (such as customers, departments, or business units) within the same SOAR instance.7 Each environment functions as a separate workspace, with its own set of cases, alerts, assets, and incident data. This ensures that users and teams operating in one environment cannot access or view data in another, unless they are explicitly granted permission.

Global Resources and Playbooks: While data such as cases is segregated by environment, key SOAR components like playbooks are managed at the global scope. This allows you to create, test, and manage playbooks centrally and then make them available for use across any or all of your environments. This capability enables resource re-use and standardization of response procedures, even in a multi-tenant configuration.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > SOAR Administration > Environments

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Playbooks > Playbook Management

Question: 29

Your Google Security Operations (SecOps) case queue contains a case with IP address entities. You need to determine whether the entities are internal or external assets and ensure that internal IP address entities are marked accordingly upon ingestion into Google SecOps SOAR. What should you do?

A. Configure a feed to ingest enrichment data about the networks, and include these fields into your detection outcome.

B. Modify the connector logic to perform a secondary lookup against your CMDB and flag incoming entities as internal or external.

C. Indicate your organization's known internal CIDR ranges in the Environment Networks list in the settings.

D. Create a custom action to ping the IP address entity from your Remote Agent. If successful, the custom action designates the IP address entity as internal.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option C. Google SecOps SOAR includes a specific, built-in feature to address this exact requirement. The SOAR platform needs to be context-aware to differentiate between internal and external IPs for accurate analysis, prioritization, and playbook execution.

This is achieved by configuring the Environment Networks list within the SOAR settings. Here, an administrator defines all of the organization's internal CIDR ranges (e.g., 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, etc.).

When an alert is ingested from the SIEM (Chronicle) or any other source, the SOAR platform parses its entities. During this ingestion and enrichment process, it automatically cross-references every IP address entity against the configured "Environment Networks" list. If an IP address falls within any of the defined internal CIDR blocks, it is automatically flagged as "Internal." This classification is then visible to analysts in the case and can be used by playbooks to make logical decisions (e.g., initiate an endpoint scan for an internal IP vs. block an external IP at the firewall).

Option A is incorrect because it describes enriching data in the SIEM, not the SOAR ingestion process.

Option B is incorrect because it requires custom connector modification, which is a high-effort solution, whereas a standard, out-of-the-box setting (Option C) already exists.

Option D is incorrect because it describes a post-ingestion playbook action, not a flag set upon ingestion. It's also an unreliable method, as internal assets may not respond to ping due to host firewalls.

Exact Extract from Google Security Operations Documents:

Environment Networks: Google SecOps SOAR provides a configuration setting to define the organization's internal IP address space. This setting, typically found under Organization Settings > Environment Networks within the SOAR platform, allows administrators to list all internal CIDR ranges.

When alerts are ingested into SOAR, the platform automatically enriches entities. During this process, any IP address entity is checked against this defined list. If the IP address falls within one of the specified CIDR blocks, it is automatically marked with an Internal flag. This contextual awareness is critical for analysts to triage cases and for playbooks to execute the correct logic (e.g., different actions for an internal vs. external IP).

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > SOAR Administration > Organization Settings

Question: 30

You are helping a new Google Security Operations (SecOps) customer configure access for their SOC team. The customer's Google SecOps administrators currently have access to the Google SecOps instance. The customer is reporting that the SOC team members are not getting authorized to access the instance, but they are able to authenticate to the third-party identity provider (IdP). How should you fix the issue?

Choose 2 answers

A. Link Google SecOps to a Google Cloud project with the Chronicle API.

B. Connect Google SecOps with the third-party IdP using Workforce Identity Federation.

C. Grant the appropriate data access scope to the SOC team's IdP group in IAM.

D. Grant the roles/chronicle.viewer role to the SOC team's IdP group in IAM.

E. Grant the Basic permission to the appropriate IdP groups in the Google SecOps SOAR Advanced Settings.

Answer: D, E

Explanation:

Comprehensive and Detailed Explanation

This scenario describes a common configuration task where authorization is failing despite successful authentication. The problem stems from the fact that Google SecOps uses a dual-authorization model: one for the main platform (SIEM/Chronicle) and a separate one for the SOAR module. The SOC team needs both.

The prompt states admins already have access, which confirms that prerequisite steps like linking the project (Option A) and configuring Workforce Identity Federation (Option B) are already complete. The problem is

specific to the new SOC team's group.

Fixing Instance Access (Option D):

The error "not getting authorized to access the instance" refers to the primary Google Cloud-level authorization. Access to the Google SecOps application itself is controlled by Google Cloud IAM roles on the linked project.1 The SOC team's group, which is federated from the third-party IdP, is represented as a principalSet in IAM. This principalSet must be granted an IAM role to allow sign-in. The roles/chronicle.viewer role is the minimum predefined role required to grant this application access.

Fixing SOAR Access (Option E):

Simply granting the IAM role (Option D) is not enough for the SOC team to perform its job. That role only gets them into the main SIEM interface. The SOAR module (for case management and playbooks) has its own internal role-based access control system. An administrator must also navigate within the SecOps platform to the SOAR Advanced Settings > Users & Groups and grant the SOC team's federated group a SOAR-specific permission, like "Basic" or "Analyst."

Both steps are required to fully "fix the issue" and provide the SOC team with functional access to the platform.

Exact Extract from Google Security Operations Documents:

Identity and Access Management: Access to a Google SecOps instance using a third-party IdP relies on Workforce Identity Federation, but authorization is configured in two distinct locations.

Google Cloud IAM: Authorization to the main SecOps instance (including the SIEM interface) is controlled by Google Cloud IAM.2 The federated identities (groups) from the third-party IdP are mapped to a principalSet. This principalSet must be granted an IAM role on the Google Cloud project linked to the SecOps instance. The roles/chronicle.viewer role is the minimum predefined role required to grant sign-in access.

Google SecOps SOAR: Authorization for the SOAR module (for case management and playbooks) is managed independently.3 An administrator must navigate to the SOAR Advanced Settings > Users & Groups and assign a SOAR-specific role (e.g., 'Basic' or 'Analyst') to the same federated IdP group.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Onboard > Configure a third-party identity provider

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > SOAR Administration > Users and Groups

Question: 31

You received an alert from Container Threat Detection that an added binary has been executed in a business critical workload. You need to investigate and respond to this incident. What should you do?

Choose 2 answers

A. Review the finding, quarantine the cluster containing the running pod. and delete the running pod to prevent further compromise.

B. Keep the cluster and pod running, and investigate the behavior to determine whether the activity is malicious.

C. Notify the workload owner. Follow the response playbook. and ask the threat hunting team to identify the root cause of the incident.

D. Review the finding, investigate the pod and related resources, and research the related attack and response methods.

E. Silence the alert in the Security Command Center (SCC) console, as the alert is a low severity finding.

Answer: C, D

Explanation:

Comprehensive and Detailed Explanation

The correct actions are C and D, as they represent the standard, parallel process for incident response: technical investigation and procedural/communicative response.

Technical Investigation (Option D): The immediate priority is to understand the alert. An analyst must review the Container Threat Detection finding in Security Command Center (SCC) to understand what was detected. This is followed by investigating the affected pod, its container, the node it's running on, and any associated service accounts to determine the initial blast radius and gather forensic data. Researching the binary and related TTPs (Tactics, Techniques, and Procedures) helps contextualize the attack.

Procedural Response (Option C): Concurrently, the organizational response plan must be activated. This involves notifying the business-critical workload owner (stakeholder communication), initiating the formal, documented incident response playbook, and escalating to specialized teams, like threat hunting, for deeper root cause analysis that goes beyond the initial triage.

Option A is incorrect because deleting the pod immediately is a premature remediation step that destroys critical forensic evidence. Option B is incorrect because "keeping the cluster and pod running" without any containment is reckless and could allow an attacker to pivot. Option E is incorrect because an unauthorized binary execution in a critical workload is a high-severity event, not a low-severity finding to be silenced.

Exact Extract from Google Security Operations Documents:

Responding to Container Threat Detection findings: When a Container Threat Detection finding is generated, it indicates a potential security issue that requires investigation. The first step is to review the finding details in Security Command Center (SCC) to understand the nature of the threat, such as K8S_BINARY_EXECUTED.

The recommended workflow involves:

Investigate: Examine the affected Kubernetes resources, such as the Pod, Container, and Node. Use tools like kubectl to inspect the pod configuration, running processes, and network connections.
Research the associated attack and response methods to understand the threat actor's TTPs.
Respond: Follow the organization's incident response playbook. This includes notifying the workload owner and relevant stakeholders. Contain the threat by isolating the pod or node, but avoid deleting resources immediately to preserve evidence for forensic analysis.

Escalate: For complex incidents, engage the threat hunting or forensics team to conduct a thorough investigation, identify the root cause, and determine the full scope of the compromise.

Reference:

Google Cloud Documentation: Security Command Center > Documentation > Manage findings > Responding to Container Threat Detection findings
Google Cloud Documentation: Google Security Operations > Documentation > Incident Response > Incident Response Playbooks

Question: 32

Your company is adopting a multi-cloud environment. You need to configure comprehensive monitoring of threats using Google Security Operations (SecOps). You want to start identifying threats as soon as possible. What should you do?

A. Use Gemini to generate YARA-L rules for multi-cloud use cases.

B. Use curated detections from the Cloud Threats category to monitor your cloud environment.

C. Use curated detections for Applied Threat Intelligence to monitor your company's cloud environment.

D. Ask Cloud Customer Care to provide a set of rules recommended by Google to monitor your company's cloud environment.

Answer: B

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option B. The key requirements are "comprehensive monitoring" and "as soon as possible" in a "multi-cloud environment."

Google Security Operations provides Curated Detections, which are out-of-the-box, fully managed rule sets maintained by the Google Cloud Threat Intelligence (GCTI) team. These rules are designed to provide immediate value and broad threat coverage without requiring manual rule writing, tuning, or maintenance.

Within the curated detection library, the Cloud Threats category is the specific rule set designed to detect threats against cloud infrastructure. This category is not limited to Google Cloud; it explicitly includes detections for anomalous behaviors, misconfigurations, and known attack patterns across multi-cloud environments, including AWS and Azure.

Enabling this category is the fastest and most effective way to meet the requirement. Option A (using Gemini) requires manual effort to generate, validate, and test rules. Option C (Applied Threat Intelligence) is a different category that focuses primarily on matching known, high-impact Indicators of Compromise (IOCs) from GCTI, which is less comprehensive than the behavior-based rules in the "Cloud Threats" category. Option D is procedurally incorrect; Customer Care provides support, but detection content is delivered directly within the SecOps platform.

Exact Extract from Google Security Operations Documents:

Google SecOps Curated Detections: Google Security Operations provides access to a library of curated detections that are created and managed by Google Cloud Threat Intelligence (GCTI). These rule sets provide a baseline of threat detection capabilities and are updated continuously.

Curated Detection Categories: Detections are grouped into categories that you can enable based on your organization's needs and data sources. The 'Cloud Threats' category provides broad coverage for threats targeting cloud environments. This rule set includes detections for anomalous activity and common attack techniques across GCP, AWS, and Azure, making it the ideal choice for securing a multi-cloud deployment. Enabling this category allows organizations to start identifying threats immediately.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Curated detections > Curated detection rule sets

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Curated detections > Cloud Threats rule set

Question: 33

You work for an organization that operates an ecommerce platform. You have identified a remote shell on your company's web host. The existing incident response playbook is outdated and lacks specific procedures for handling this attack. You want to create a new, functional playbook that can be deployed as soon as possible by junior analysts. You plan to use available tools in Google Security Operations (SecOps) to streamline the

playbook creation process. What should you do?

A. Use Gemini to generate a playbook based on a template from a standard incident response plan, and implement automated scripts to filter network traffic based on known malicious IP addresses.

B. Add instruction actions to the existing incident response playbook that include updated procedures with steps that should be completed. Have a senior analyst build out the playbook to include those new procedures.

C. Use the playbook creation feature in Gemini, and enter details about the intended objectives. Add the necessary customizations for your environment, and test the generated playbook against a simulated remote shell alert.

D. Create a new custom playbook based on industry best practices, and work with an offensive security team to test the playbook against a simulated remote shell alert.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option C. The primary constraints are to "streamline" the process, create a "new, functional playbook," get it "as soon as possible," and "use available tools in Google Security Operations."

Google Security Operations integrates Gemini directly into the SOAR platform to accelerate security operations. One of its key capabilities is generative playbook creation. This feature allows an analyst to describe their intended objectives in natural language (e.g., "Create a playbook to investigate and respond to a remote shell alert"). Gemini then generates a complete, logical playbook flow, including investigation, enrichment, containment, and eradication steps.

This generated playbook serves as a high-quality draft. The analyst can then add the necessary customizations (like specific tools, notification endpoints, or contacts for the e-commerce platform) and, most importantly, test the playbook to ensure it is functional and reliable for junior analysts to execute. This workflow directly meets all the prompt's requirements, especially "streamline" and "as

soon as possible."

Option D (creating a custom playbook from scratch and using a red team) is the exact opposite of streamlined and fast. Option B involves patching an "outdated" playbook, not creating a new one. Option A incorrectly bundles a specific remediation action (filtering traffic) with the playbook creation process.

Exact Extract from Google Security Operations Documents:

Gemini for Security Operations: Gemini in Google SecOps provides generative AI to assist analysts and engineers. Within the SOAR capability, Gemini can generate entire playbooks from natural language prompts.

Playbook Creation with Gemini: Instead of building a playbook manually, an engineer can describe the intended objectives of the response plan. Gemini will generate a new playbook with a logical structure, including relevant actions and conditional branches. This generated playbook serves as a strong foundation, which can then be refined. The engineer can add necessary customizations to tailor the playbook to the organization's specific environment, tools, and processes. Before deploying the playbook for use by the SOC, it is a best practice to test it against simulated alerts to validate its functionality and ensure it runs as expected.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Gemini in SOAR > Create playbooks with Gemini

Question: 34

You are a platform engineer at an organization that is migrating from a third-party SIEM product to Google Security Operations (SecOps). You previously manually exported context data from Active Directory (AD) and imported the data into your previous SIEM as a watchlist when there were changes in AD's user/asset context data. You want to improve this process using Google SecOps. What should you do?

A. Ingest AD organizational context data as user/asset context to enrich user/asset information in your security events.

B. Configure a Google SecOps SOAR integration for AD to enrich user/asset information in your security alerts.

C. Create a data table that contains AD context data. Use the data table in your YARA-L rule to find user/asset data that can be correlated within each security event.

D. Create a data table that contains the AD context data. Use the data table in your YARA-L rule to find user/asset information for each security event.

Answer: A

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option A. The key requirement is to "improve" the previous manual "watchlist" process.

In Google Security Operations, "data tables" (mentioned in options C and D) are the modern equivalent of watchlists or reference lists.1 Using a data table would replicate the old, static process and would not be an improvement.

The superior method in Google SecOps is to ingest this data as Entity Context. This is a core feature where context data (like user information from AD or asset data from a CMDB) is ingested via a feed or the Context API. Google SecOps then uses this data to automatically enrich all incoming security events (UDM) in real-time.

When a log for john.doe is ingested, it is automatically enriched with the context data from AD, such as "John Doe," "Marketing Department," "Manager: Jane Smith," etc. This enriched information is then available for detection, hunting, and investigation. This is a significant improvement because it provides continuous, automatic enrichment at ingestion, rather than requiring a manual update of a static table or only enriching after an alert is generated (Option B).

Exact Extract from Google Security Operations Documents:

UDM enrichment and aliasing overview: Google Security Operations (SecOps) supports aliasing and enrichment for assets and users.2 Aliasing enables enrichment.3 For example, using aliasing, you can find the job title and employment status associated with a user ID.4

How aliasing works: User aliasing uses the USER_CONTEXT event type for aliasing.5 This contextual data is stored as entities in the Entity Graph.6 When new Unified Data Model (UDM) events are ingested, enrichment uses this aliasing data to add context to the UDM event.7 For example, a UDM event might include principal.user.userid = "jdoe". 8The enrichment process populates the principal.user noun with the entity data, such as user.user_display_name = "John Doe" and user.department = "Marketing".

This is the recommended method for ingesting organizational context from sources like Microsoft Windows Active Directory, as it makes the contextual data available for all subsequent detection,

search, and investigation activities.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Event processing > UDM enrichment and aliasing overview

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Collect Microsoft Windows AD logs (This document explicitly mentions collecting USER_CONTEXT and ASSET_CONTEXT).9

Question: 35

You have been tasked with creating a YARA-L detection rule in Google Security Operations (SecOps). The rule should identify when an internal host initiates a network connection to an external IP address that the Applied Threat Intelligence Fusion Feed associates with indicators attributed to a specific Advanced Persistent Threat 41

(APT41) threat group. You need to ensure that the external IP address is flagged if it has a documented relationship to other APT41 indicators within the Fusion Feed. How should you configure this YARA-L rule?

A. Configure the rule to trigger when the external IP address from the network connection event matches an entry in a manually pre-curated data table of all APT41-related IP addresses.

B. Configure the rule to establish a join between the live network connection event and Fusion Feed data for the common external IP address. Filter the joined Fusion Feed data for explicit associations with the APT41 threat group or related indicators.

C. Configure the rule to check whether the external IP address from the network connection event has a high confidence score across any enabled threat intelligence feed.

D. Configure the rule to detect outbound network connections to the external IP address. Create a Google SecOps SOAR playbook that queries the Fusion Feed to determine if the IP address has an APT41 relationship.

Answer: B

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option B. This question tests the advanced detection capabilities of YARA-L when using the Applied Threat Intelligence (ATI) Fusion Feed.

The key requirement is to find an IP that not only matches but has a documented relationship to APT41. The ATI Fusion Feed is not just a flat list of IOCs; it is a context-rich graph of indicators, malware, threat actors, and their relationships, managed by Google's threat intelligence teams.10

Option A is incorrect because it describes a manual, static list (data table) and cannot query the relationships in the live feed.

Option C is incorrect because it is too generic ("high confidence score," "any feed"). The requirement is specific to the ATI Fusion Feed and APT41.

Option D is incorrect because it describes a post-detection SOAR action. The question explicitly asks how to configure the YARA-L detection rule itself to perform this correlation.

Option B is the only one that describes the correct YARA-L 2.0 methodology. The rule must first define the live event (network connection). Then, it must define the context source (the ATI Fusion Feed). In the events section of the rule, a join is established between the event's external IP field and the IP indicator in the Fusion Feed. Finally, the rule filters the joined context data, looking for attributes such as threat.threat_actor.name = "APT41" or other related_indicators that link back to the specified threat group.

Exact Extract from Google Security Operations Documents:

Applied Threat Intelligence Fusion Feed overview: The Applied Threat Intelligence (ATI) Fusion Feed is a collection of Indicators of Compromise (IoCs), including hashes, IPs, domains, and URLs, that are associated with known threat actors, malware strains, active campaigns, and finished intelligence reporti11ng.12

Write YARA-L rules with the ATI Fusion Feed: Writing YARA-L rules that use the ATI Fusion Feed follows a similar process to writing YARA-L rules that use other context entity sources.13 To write a rule, you filter the selected context entity graph (in this case, Fusion Feed).14

You can join a field from the context entity and UDM event field. In the following example, the placeholder variable ioc is used to do a transitive join between the context entity and the event.

Because this rule can match a large number of events, it is recommended that you refine the rule to match on context entities that have specific intelligence. This allows you to filter for explicit associations, such as a specific threat group or an indicator's presence in a compromised environment.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Applied Threat Intelligence Fusion Feed overview

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Create context-aware analytics

Question: 36

You are a SOC manager at an organization that recently implemented Google Security Operations (SecOps). You need to monitor your organization's data ingestion health in Google SecOps. Data is ingested with Bindplane collection agents. You want to configure the following:

- Receive a notification when data sources go silent within 15 minutes.
- Visualize ingestion throughput and parsing errors.

What should you do?

A. Configure automated scheduled delivery of an ingestion health report in the Data Ingestion and Health dashboard. Monitor and visualize data ingestion metrics in this dashboard.

B. Configure silent source alerts based on rule detections for anomalous data ingestion activity in Risk Analytics. Monitor and visualize the alert metrics in the Risk Analytics dashboard.

C. Configure notifications in Cloud Monitoring when ingestion sources become silent in Bindplane. Monitor and visualize Google SecOps data ingestion metrics using Bindplane Observability Pipeline (OP).

D. Configure silent source notifications for Google SecOps collection agents in Cloud Monitoring. Create a Cloud Monitoring dashboard to visualize data ingestion metrics.

Answer: D

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option D. This approach correctly uses the integrated Google Cloud-native tools for both monitoring and alerting.

Google Security Operations (SecOps) automatically streams all ingestion metrics to Google Cloud Monitoring. This includes metrics for throughput (e.g., chronicle.googleapis.com/ingestion/event_count, chronicle.googleapis.com/ingestion/byte_count), parsing errors (e.g., chronicle.googleapis.com/ingestion/parse_error_count), and the health of collection agents (e.g., chronicle.googleapis.com/ingestion/last_seen_timestamp).

Receive a notification (15 minutes): The Data Ingestion and Health dashboard (Option A) is for visualization, and its "reports" are scheduled summaries, not real-time alerts. The only way to get a

15-minute notification is to use Cloud Monitoring. An alerting policy can be configured to trigger when a "metric absence" is detected for a specific collection agent's last_seen_timestamp, fulfilling the "silent source" requirement.

Visualize metrics: Cloud Monitoring also provides a powerful dashboarding service. A Cloud Monitoring dashboard can be built to graph all the necessary metrics—throughput, parsing errors, and agent status—in one place.

Option C is incorrect because it suggests using the Bindplane Observability Pipeline, which is a separate product. Option B is incorrect as Risk Analytics is for threat detection (UEBA), not platform health.

Exact Extract from Google Security Operations Documents:

Use Cloud Monitoring for ingestion insights: Google SecOps uses Cloud Monitoring to send the ingestion notifications. Use this feature for ingestion notifications and ingestion volume viewing.

Set up a sample policy to detect silent Google SecOps collection agents:

In the Google Cloud console, select Monitoring.

Click Create Policy.

On the Select a metric page, select Chronicle Collector > Ingestion > Total ingested log count.

In the Transform data section, set the Time series group by to collector_id.

Click Next.

Select Metric absence and set the Trigger absence time (e.g., 15 minutes).

In the Notifications and name section, select a notification channel.

You can also create custom dashboards in Cloud Monitoring to visualize any of the exported metrics, such as Total ingested log size or Total record count (for parsing).

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Use Cloud Monitoring for ingestion insights

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Silent-host monitoring > Use Google Cloud Monitoring with ingestion labels for SHM

Question: 37

Your organization is a Google Security Operations (SecOps) customer. The compliance team requires a weekly export of case resolutions and SLA metrics of high and critical severity cases over the past week. The compliance team's post-processing scripts require this data to be formatted as tabular data in CSV files, zipped, and delivered to their email each Monday morning. What should you do?

A. Generate a report in SOAR Reports, and schedule delivery of the report.

B. Build a detection rule with outcomes, and configure a Google SecOps SOAR job to format and send the report.

C. Build an Advanced Report in SOAR Reports, and schedule delivery of the report.

D. Use statistics in search, and configure a Google SecOps SOAR job to format and send the report.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option C. Google SecOps SOAR has a specific feature designed for this exact use case: Advanced Reports.

The standard "SOAR Reports" (Option A) are pre-canned dashboard-style reports (e.g., Management - SOC Status). However, the "Advanced Reports" feature (built on Looker) provides a powerful, flexible interface for building highly customized, tabular reports based on case data. This allows an administrator to specifically query for case resolutions and SLA metrics, and filter them by priority = High OR Critical.

Most importantly, the Advanced Reports feature has a built-in scheduler. This scheduler can be configured to

run the report at a specific cadence (e.g., "Weekly on Monday at 9:00 AM"), send it to a list of email recipients, and attach the data in the required format, including CSV and as a zipped file.

Option B is incorrect because detection rules create alerts, they don't report on case metrics. Option D is incorrect because it mixes the SIEM search function with a SOAR job, which is an overly complex and unnecessary way to query case data that is already structured within the SOAR module.

Exact Extract from Google Security Operations Documents:

Explore advanced SOAR reports: The default advanced SOAR reports are a set of dashboards and reports to help track SOC performance, case handling, analyst workload, and automation efficiency. These reports provide both high-level and detailed insights across your environments.1

SLA Monitoring: Use Triage Time and SLA Met flag to monitor SLA compliance and improve case handling.

Manage advanced reports: You can create, edit, duplicate, share, download, and delete advanced reports.

Schedule a report:

Select the report you want to schedule.

Select the Scheduler tab and click Add.

In the New Schedule dialog, click the Enable toggle to turn on scheduling and enter the required information (e.g., weekly, Monday, email recipients).

You can select the delivery format, including CSV and ZIP attachments.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Monitor and report > SOAR reports > Use Looker Explores in SOAR reports (Advanced Reports)

Google Cloud Documentation: Google Security Operations > Documentation > Monitor and report > SOAR reports > Explore SOAR reports

Question: 38

You are developing a security strategy for your organization. You are planning to use Google Security Operations (SecOps) and Google Threat Intelligence (GTI). You need to enhance the detection and response across multi-cloud and on-premises systems. How should you integrate these products?

Choose 2 answers

A. Ingest GTI IOCs into Google SecOps as security events.

B. Ingest on-premises and cloud security logs into Google SecOps SIEM as events.

C. Ingest on-premises and cloud security logs into Google SecOps SIEM as entities.

D. Use Google SecOps SOAR integrations with GTI for event enrichment.

E. Use Google SecOps SOAR integrations with GTI for entity enrichment.

Answer: B, D

Explanation:

Comprehensive and Detailed Explanation

The correct answers are B and D, as they accurately describe the two primary functions of a modern SecOps platform: SIEM (Detection) and SOAR (Response).

Option B: (Detection Strategy) A SIEM's fundamental purpose is to perform detection. To do this, it must first ingest telemetry (logs) as events. This is the foundational step for any detection and response strategy. Logs from all sources—on-premises (e.g., firewalls, Active Directory) and multicloud (e.g., AWS CloudTrail, Azure Activity Logs) —are ingested into Google SecOps, normalized into the Unified Data Model (UDM), and stored as events. This is what allows detection rules to run. (Option C is incorrect as logs are events, not entities).

Option D: (Response Strategy) A SOAR's fundamental purpose is to orchestrate and automate the response to a detection. A key part of this response is event enrichment (or more specifically, observable enrichment). When an alert is ingested by the SOAR, a playbook runs. This playbook uses integrations (e.g., with Mandiant or VirusTotal, which are part of GTI) to query for real-time context on the observables (IPs, hashes, domains) in the alert. This enrichment helps an analyst make a decision or allows the playbook to automate a containment action.

Option A is incorrect because GTI is ingested as context (in the entity graph and Fusion Feed), not as events.
Option E is incorrect because "entity enrichment" (e.g., adding user data from AD) happens at the SIEM ingestion level, whereas SOAR integrations perform on-demand enrichment for alerts/events.
Exact Extract from Google Security Operations Documents:

Google SecOps data ingestion: Google Security Operations ingests customer logs, normalizes the data, and detects security alerts. Google SecOps ingests data using... Forwarders, Bindplane agent, Ingestion APIs, Google Cloud. Parsers convert logs from customer systems into a Unified Data Model (UDM) events.

Integrate Mandiant Threat Intelligence with Google SecOps: This document provides guidance on how to integrate Mandiant Threat Intelligence with Google Security Operations (Google SecOps). After you configure an integration instance, you can use it in playbooks.

Actions:

Enrich Entities: Use the Enrich Entities action to enrich entities using the information from Mandiant Threat Intelligence. This action runs on the following Google SecOps entities: Hostname, IP Address, URL, File Hash.

Enrich IOCs: Use this action to enrich indicators of compromise.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SecOps > Google SecOps data ingestion

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations > Mandiant Threat Intelligence

Question: 39

Your company requires PCI DSS v4.0 compliance for its cardholder data environment (CDE) in Google Cloud. You use a Security Command Center (SCC) security posture deployment based on the PCI DSS v4.0 template to monitor for configuration drift.1 This posture generates a finding indicating that a Compute Engine VM within the CDE scope has been configured with an external IP address. You need to take an immediate action to remediate the compliance drift identified by this specific SCC posture finding. What should you do?

A. Enable and enforce the constraints/compute.vmExternalIpAccess organization policy constraint at the project level for the project where the VM resides.

B. Remove the CDE-specific tag from the VM to exclude the tag from this particular PCI DSS posture evaluation scan.

C. Reconfigure the network interface settings for the VM to explicitly remove the assigned external IP address.

D. Navigate to the underlying Security Health Analytics (SHA) finding for public_ip_address on the VM. and mark this finding as fixed.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct answer is Option C. The question asks for the immediate action to remediate the existing compliance drift, which is the VM that already has an external IP address.

Option C (Remediate): Reconfiguring the VM's network interface to remove the external IP directly fixes the identified misconfiguration. This action brings the resource back into compliance, which will cause the Security Command Center finding to be automatically set to INACTIVE on its next scan.2

Option A (Prevent): Applying the organization policy constraints/compute.vmExternalIpAccess is a preventative control.3 It will stop new VMs from being created with external IPs, but it is not retroactive and does not remove the external IP from the already existing VM. Therefore, it does not remediate the current finding.

Option B (Mask): Removing the tag simply hides the resource from the posture scan. This is a violation of compliance auditing; it masks the problem instead of fixing it.

Option D (Ignore): Marking a finding as fixed without actually fixing the underlying issue is incorrect and will not resolve the compliance drift. The finding will reappear as ACTIVE on the next scan.

Exact Extract from Google Security Operations Documents:

Finding deactivation after remediation: After you remediate a vulnerability or misconfiguration finding, the Security Command Center service that detected the finding automatically sets the state of the finding to INACTIVE the next time the detection service scans for the finding.4 How long Security Command Center takes to set a remediated finding to INACTIVE depends on the schedule of the scan that detects the findin5g.

Organization policy constraints: If enforced, the constraint constraints/compute.vmExternalIpAccess will deny the creation or update of VM instances with IPv4 external IP addresses.6 This constraint is not retroactive and will not restrict the usage of external IPs on existing VM instances. To remediate an existing VM, you must modify the instance's network interface settings and remove the external IP.

Reference:

Google Cloud Documentation: Security Command Center > Documentation > Manage findings > Vulnerability findings > Finding deactivation after remediation7

Google Cloud Documentation: Resource Manager > Documentation > Organization policy > Organization policy constraints > compute.vmExternalIpAccess

Question: 40

A Google Security Operations (SecOps) detection rule is generating frequent false positive alerts. The rule was designed to detect suspicious Cloud Storage enumeration by triggering an alert whenever the storage.objects.list API operation is called using the api.operation UDM field. However, a legitimate backup automation tool that uses the same API, causing the rule to fire unnecessarily. You need to reduce these false positives from this trusted backup tool while still detecting potentially malicious usage. How should you modify the rule to improve its accuracy?

A. Adjust the rule severity to low to deprioritize alerts from automation tools.

B. Convert the rule into a multi-event rule that looks for repeated API calls across multiple buckets.

C. Replace api.operation with api.service_name = "storage.googleapis.com" to narrow the detection scope.

D. Add principal.user.email != "backup-bot@fcobaa.com" to the rule condition to exclude the automation account.

Answer: D

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option D. The problem is that a known, trusted principal (the backup tool's service account) is performing a legitimate action (storage.objects.list) that happens to look like the suspicious behavior the rule is designed to catch.

The most precise and effective way to reduce these false positives without weakening the rule's ability to catch malicious actors is to create an exception for the trusted principal.

By adding principal.user.email != "backup-bot@fcobaa.com" (or the equivalent principal.user.userid) to the events or condition section of the YARA-L rule, the rule will now only evaluate events where the actor is not the known-good backup bot.

Option A is incorrect because it just lowers the priority of the false positive; it doesn't stop it from being generated.

Option B is incorrect because the legitimate tool might also perform repeated calls, leading to the same false positive.

Option C is incorrect because api.service_name = "storage.googleapis.com" is less specific than api.operation = "storage.objects.list" and would likely increase the number of false positives by triggering on any storage API call.

Exact Extract from Google Security Operations Documents:

Reduce false positives: When a detection rule generates false positives due to known-benign activity (e.g., from an administrative script or automation tool), the best practice is to add a not condition to the rule to exclude the trusted entity.8

You can filter on UDM fields to create exceptions. For example, to prevent a rule from firing on activity from a specific service account, you can add a condition to the events section such as:

and $e.principal.user.userid != "trusted-service-account@project.iam.gserviceaccount.com"

This technique, often called "allow-listing" or "suppression," improves the rule's accuracy by focusing only on unknown or untrusted principals.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Overview

of the YARA-L 2.0 language > Add not conditions to prevent false positives

Question: 41

You are responsible for identifying suspicious activity and security events in your organization's environment. You discover that some detection rules are generating false positives when the principal.ip field contains one or more IP addresses in the 192.168.2.0/24 subnet. You want to improve these detection rules using the principal.ip repeated field. What should you add to the YARA-L detection rules?

A. net.ip_in_range_cidr(all $e.principal.ip, "192.168.2.0/24")

B. net.ip_in_range_cidr(any $e.principal.ip, "192.168.2.0/24")

C. not net.ip_in_range_cidr(all $e.principal.ip, "192.168.2.0/24")

D. not net.ip_in_range_cidr(any $e.principal.ip, "192.168.2.0/24")

Answer: D

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option D. The goal is to exclude events (i.e., stop false positives) when the principal.ip field contains any IP from the trusted 192.168.2.0/24 subnet.

The principal.ip field in UDM is a repeated field, meaning it can hold an array of values (e.g., ["1.2.3.4", "192.168.2.5"]). YARA-L provides the any and all quantifiers to handle repeated fields.9

any $e.principal.ip: This checks if at least one IP in the array meets the condition.

all $e.principal.ip: This checks if every IP in the array meets the condition.

The function net.ip_in_range_cidr(...) returns true if an IP is in the specified range.

Therefore, the logic we need is: "do not trigger this rule if any of the IPs in the principal.ip field are in the 192.168.2.0/24 range."

This translates directly to the YARA-L syntax: not net.ip_in_range_cidr(any $e.principal.ip, "192.168.2.0/24")

Option B would only find events from that subnet.

Option A would only find events where all associated IPs are in that subnet.

Option C is the logical inverse of A and would incorrectly filter out events that might be malicious (e.g., ["1.2.3.4", "192.168.2.5"] would not be excluded because all IPs are not in the range).

Exact Extract from Google Security Operations Documents:

YARA-L 2.0 language syntax > Repeated fields and boolean expressions: When a boolean expression, such as a function call, is applied to a repeated field, you can use the any or all keywords to specify how the expression should be evaluated.10

any <repeated_field>: The expression evaluates to true if it is true for at least one of the values in the repeated field.

all <repeated_field>: The expression evaluates to true only if it is true for all of the values in the repeated field.

Functions > net.ip_in_range_cidr: The net.ip_in_range_cidr function is useful to bind rules to specific parts of the network.11 To exclude all private netblocks as defined in RFC1918, you can add a not to the start of the criteria:

and not (net.ip_in_range_cidr(any $e.principal.ip, "10.0.0.0/8") or net.ip_in_range_cidr(any $e.principal.ip, "172.16.0.0/12") or net.ip_in_range_cidr(any $e.principal.ip, "192.168.0.0/16"))

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > YARA-L 2.0 language syntax

Google Cloud Documentation: Google Security Operations > Documentation > Detections > YARA-L 2.0 functions > net.ip_in_range_cidr

Question: 42

Your company's SOC recently responded to a ransomware incident that began with the execution of a malicious document. EDR tools contained the initial infection. However, multiple privileged service accounts continued to exhibit anomalous behavior, including credential dumping and scheduled task creation. You need to design an automated playbook in Google Security Operations (SecOps) SOAR to minimize dwell time and accelerate containment for future similar attacks. Which action should you take in your Google SecOps SOAR playbook to support containment and escalation?

A. Create an external API call to VirusTotal to submit hashes from forensic artifacts.

B. Add an approval step that requires an analyst to validate the alert before executing a containment action.

C. Configure a step that revokes OAuth tokens and suspends sessions for high-privilege accounts based on entity risk.

D. Add a YARA-L rule that sends an alert when a document is executed using a scripting engine such as wscript.exe.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct answer is Option C. The incident description makes it clear that endpoint containment (by EDR) was insufficient, as the attacker successfully pivoted to privileged service accounts and began post-compromise activities (credential dumping, scheduled tasks).

The goal is to automate containment and minimize dwell time.

Option A is an enrichment/investigation action, not a containment action.

Option B is the opposite of automation; adding a manual approval step increases dwell time and response time.

Option D is a detection engineering task (creating a YARA-L rule), not a SOAR playbook (response) action.

Option C is the only true automated containment action that directly addresses the new threat. The anomalous behavior of the privileged accounts would raise their Entity Risk Score within Google SecOps. A modern SOAR playbook can be configured to automatically trigger on this high-risk score and execute an identity-based containment action. Revoking tokens and suspending sessions for the compromised high-privilege accounts is the most effective way to immediately stop the attacker's lateral movement and malicious activity, thereby accelerating containment and minimizing dwell time.

Exact Extract from Google Security Operations Documents:

SOAR Playbooks and Automation: Google Security Operations (SecOps) SOAR enables the orchestration and automation of security responses. Playbooks are designed to execute a series of automated steps to respond to an alert.

Identity and Access Management Integrations: SOAR playbooks can integrate directly with Identity Providers (IdPs) like Google Workspace, Okta, and Microsoft Entra ID. A critical automated containment action for compromised accounts is to revoke active OAuth tokens, suspend user

sessions, or disable the account entirely. This action immediately logs the attacker out of all active sessions and prevents them from re-authenticating.

Entity Risk: Detections and anomalous activities contribute to an entity's (e.g., a user or asset) risk score. Playbooks can be configured to use this risk score as a trigger. For example, if a high-privilege account's risk score crosses a critical threshold, the playbook can automatically execute identity containment actions.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Playbooks > Playbook Actions

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations > (e.g., Okta, Google Workspace)

Google Cloud Documentation: Google Security Operations > Documentation > Investigate > View entity risk scores

Question: 43

You use Google Security Operations (SecOps) curated detections and YARA-L rules to detect suspicious activity on Windows endpoints. Your source telemetry uses EDR and Windows Events logs. Your rules match on the principal.user.userid UDM field. You need to ingest an additional log source for this field to match all possible log entries from your EDR and Windows Event logs. What should you do?

A. Ingest logs from Microsoft Entra ID.

B. Ingest logs from Windows Procmon.

C. Ingest logs from Windows PowerShell.

D. Ingest logs from Windows Sysmon.

Answer: A

Explanation:

Comprehensive and Detailed Explanation

The correct answer is Option A. This question is about entity context enrichment and aliasing.

Endpoint telemetry from EDR and Windows Event Logs (like 4624) identifies users by their Windows Security Identifier (SID) (e.g., S-1-5-21-12345...). However, detection rules are more effective when they match on a human-readable and consistent identifier, like an email address or username, which is stored in principal.user.userid.

To "connect the dots" between the SID found in endpoint events and the userid, Google SecOps must ingest an authoritative user context data source. In a modern Windows environment, this source is Microsoft Entra ID (formerly Azure AD) or on-premises Active Directory.

Ingesting Entra ID logs as a USER_CONTEXT feed populates the SecOps entity graph. This allows the platform to automatically alias the SID from an endpoint log to the corresponding userid (e.g., jsmith@company.com) at ingestion time. This ensures the principal.user.userid field is correctly populated, allowing the detection

rules to match.

Options B, C, and D are all additional event sources (like EDR) and would provide more SIDs, but they do not provide the central directory data needed to perform the aliasing.

Exact Extract from Google Security Operations Documents:

UDM enrichment and aliasing overview: Google Security Operations (SecOps) supports aliasing and enrichment for assets and users. Aliasing enables enrichment. For example, using aliasing, you can find the job title and employment status associated with a user ID.

How aliasing works: User aliasing uses the USER_CONTEXT event type for aliasing. This contextual data is stored as entities in the Entity Graph. When new Unified Data Model (UDM) events are ingested, enrichment uses this aliasing data to add context to the UDM event. For example, an EDR log might contain a principal.windows_sid. The enrichment process queries the entity graph (populated by your Active Directory or Entra ID feed) and populates the principal.user.userid and other fields in the principal.user noun.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Event processing > UDM enrichment and aliasing overview

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Collect Microsoft Entra ID logs

Question: 44

Your organization uses the curated detection rule set in Google Security Operations (SecOps) for high priority network indicators. You are finding a vast number of false positives coming from your onpremises proxy servers. You need to reduce the number of alerts. What should you do?

A. Configure a rule exclusion for the target.ip field.

B. Configure a rule exclusion for the principal.ip field.

C. Configure a rule exclusion for the network.asset.ip field.

D. Configure a rule exclusion for the target.domain field.

Answer: B

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option B. This is a common false positive tuning scenario.

The "high priority network indicators" rule set triggers when it sees a connection to or from a known- malicious IP or domain. The problem states the false positives are coming from the on-premises proxy servers.

This implies that the proxy server itself is initiating traffic that matches these indicators. This is often benign, legitimate behavior, such as:

Resolving a user-requested malicious domain via DNS to check its category.

Performing an HTTP HEAD request to a malicious URL to scan it.

Fetching its own threat intelligence or filter updates.

In all these cases, the source of the network connection is the proxy server. In the Unified Data Model (UDM), the source IP of an event is stored in the principal.ip field.

To eliminate these false positives, you must create a rule exclusion (or add a not condition to the rule) that tells the detection engine to ignore any events where the principal.ip is the IP address of your trusted proxy servers. This will not affect the rule's ability to catch a workstation behind the proxy (whose IP would be the principal.ip) connecting through the proxy to a malicious target.ip.

Exact Extract from Google Security Operations Documents:

Curated detection exclusions: Curated detections can be tuned by creating exclusions to reduce false positives from known-benign activity. You can create exclusions based on any UDM field.

Tuning Network Detections: A common source of false positives for network indicator rules is trusted network infrastructure, such as proxies or DNS servers. This equipment may generate traffic to malicious domains or IPs as part of its normal operation (e.g., DNS resolution, content filtering lookups). In this scenario, the traffic originates from the infrastructure device itself. To filter this noise, create an exclusion where the principal.ip field matches the IP address (or IP range) of the trusted proxy server. This prevents the rule from firing on the proxy's administrative traffic while preserving its ability to detect threats from end-user systems.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Curated detections > Tune curated detections with exclusions

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Overview of the YARA-L 2.0 language

Question: 45

You are responsible for evaluating the level of effort required to integrate a new third-party endpoint detection tool with Google Security Operations (SecOps). Your organization's leadership wants to minimize customization for the new tool for faster deployment. You need to verify that the Google SecOps SOAR and SIEM support the expected workflows for the new third-party tool. You must recommend a tool to your leadership team as quickly as possible. What should you do?

Choose 2 answers

A. Review the architecture of the tool to identify the cloud provider that hosts the tool.

B. Review the documentation to identify if default parsers exist for the tool, and determine whether the logs are supported and able to be ingested.

C. Identify the tool in the Google SecOps Marketplace, and verify support for the necessary actions in the workflow.

D. Develop a custom integration that uses Python scripts and Cloud Run functions to forward logs and orchestrate actions between the third-party tool and Google SecOps.

E. Configure a Pub/Sub topic to ingest raw logs from the third-party tool, and build custom YARA-L rules in Google SecOps to extract relevant security events.

Answer: B, C

Explanation:

Comprehensive and Detailed Explanation

The core task is to evaluate a new tool for fast, low-customization deployment across the entire Google SecOps platform (SIEM and SOAR). This requires checking the two main integration points: data ingestion (SIEM) and automated response (SOAR).

SIEM Ingestion (Option B): To minimize customization for the SIEM, you must verify that Google SecOps can ingest

and understand the tool's logs out-of-the-box. This is achieved by checking the Google SecOps documentation for a default parser for that specific tool. If a default parser exists, the logs will be automatically normalized into the Unified Data Model (UDM) upon ingestion, requiring zero custom development.

SOAR Orchestration (Option C): To minimize customization for SOAR, you must verify that pre-built automated actions exist. The Google SecOps Marketplace contains all pre-built SOAR integrations (connectors). By finding the tool in the Marketplace, you can verify which actions (e.g., "Quarantine Host," "Get Process List") are supported, confirming that response playbooks can be built quickly without custom scripting.

Options D and E describe high-effort, custom integration paths, which are the exact opposite of the "minimize customization for faster deployment" requirement.

Exact Extract from Google Security Operations Documents:

Default parsers: Google Security Operations (SecOps) provides a set of default parsers that support many common security products. When logs are ingested from a supported product, SecOps automatically applies the correct parser to normalize the raw log data into the structured Unified Data Model (UDM) format. This is the fastest method to begin ingesting and analyzing new data sources.

Google SecOps Marketplace: The SOAR component of Google SecOps includes a Marketplace that contains a large library of pre-built integrations for common third-party security tools, including EDR, firewalls, and identity providers. Before purchasing a new tool, an engineer should verify its presence in the Marketplace and review the list of supported actions to ensure it meets the organization's automation and orchestration workflow requirements.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Default parsers > Supported default parsers

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations

Question: 46

You are writing a Google Security Operations (SecOps) SOAR playbook that uses the VirusTotal v3 integration to look up a URL that was reported by a threat hunter in an email. You need to use the results to make a preliminary recommendation on the maliciousness of the URL and set the severity of the alert based on the output. What should you do?

Choose 2 answers

A. Use a conditional statement to determine whether to treat the URL as suspicious or benign.

B. Pass the response back to the SIEM.

C. Verify that the response is accurate by manually checking the URL in VirusTotal.

D. Create a widget that translates the JSON output to a severity score.

E. Use the number of detections from the response JSON in a conditional statement to set the severity.

Answer: A, E

Explanation:

Comprehensive and Detailed Explanation

The goal is to automate a decision-making process within a SOAR playbook based on data from an integration. This requires two steps: getting the specific data point (Option E) and then using it in a logical operator (Option A).

Get the Data Point (Option E): The VirusTotal integration returns a detailed JSON object. The most critical data point for determining maliciousness is the number of detections (i.e., how many scanning engines flagged the URL). The playbook must parse this specific value from the JSON output.

Use the Data in Logic (Option A): Once the playbook has the number of detections, it must use a conditional statement (an "If/Then" block) to act on it. This logic is how the playbook makes a recommendation and sets the severity. For example: IF number_of_detections > 3, THEN set severity to CRITICAL and add a comment URL is suspicious. ELSE, set severity to LOW and add a comment URL appears benign.

Option C is incorrect as it describes a manual process, which defeats the purpose of automation. Option D is incorrect as widgets are for displaying data in the case UI, not for executing logic within a playbook.

Exact Extract from Google Security Operations Documents:

Playbook logic and conditional actions: SOAR playbooks execute a series of actions to automate incident response. A core component of this automation is the conditional statement. After an enrichment action (like querying VirusTotal) runs, the playbook can use a conditional block to evaluate the results.

The playbook can parse the JSON output from the integration to extract key values, such as the

number of positive detections. This value can then be used in the conditional (e.g., IF detections > 0) to determine the next step, such as setting the alert's severity, escalating to an analyst, or automatically determining if an indicator should be treated as suspicious or benign.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Playbooks > Playbook logic and conditional actions

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations > VirusTotal v3

Question: 47

Your organization has recently onboarded to Google Cloud with Security Command Center Enterprise (SCCE) and is now integrating it with your organization's SOC. You want to automate the response process within SCCE and integrate with the existing SOC ticketing system. You want to use the most efficient solution. How should you implement this functionality?

A. Use the SCC notifications feed to send alerts to Pub/Sub. Ingest these feeds using the relevant SIEM connector.

B. Evaluate each event within the SCC console. Create a ticket for each finding in the ticketing system, and include the remediation steps.

C. Disable the generic posture finding playbook in Google Security Operations (SecOps) SOAR and enable the playbook for the ticketing system. Add a step in your Google SecOps SOAR playbook to generate a ticket based on the event type.

D. Configure the SCC notifications feed to send alerts to a Cloud Storage bucket. Create a Dataflow job to read the new files, extract the relevant information, and send the information to the SOC ticketing system.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct answer is Option C. The prompt asks for the most efficient and automated solution for handling SCCE findings and integrating with a ticketing system. This is the primary use case for Google Security Operations SOAR.

The native workflow is as follows:

SCCE detects a finding.

The finding is automatically ingested into Google SecOps SIEM, which creates an alert.

The alert is automatically sent to SecOps SOAR, which creates a case.

The SOAR case automatically triggers a playbook.

Option C describes this process perfectly. An administrator would disable the default playbook and enable a specific playbook that uses a pre-built integration (from the Marketplace) for the organization's ticketing system (e.g., ServiceNow, Jira). This playbook would contain an automated step to generate a ticket, thus fulfilling the requirement efficiently.

Option B is a manual process. Options A and D describe complex, custom-built data engineering pipelines, which are far less efficient than using the built-in SOAR capabilities.

Exact Extract from Google Security Operations Documents:

SOAR Playbooks and Integrations: Google SecOps SOAR is designed to automate and orchestrate responses to alerts. When an alert from a source like Security Command Center (SCC) is ingested and creates a case, it can be configured to automatically trigger a playbook.

Ticketing Integration: A common playbook use case is integration with an external ticketing system. Using a pre-built integration from the SOAR Marketplace, an administrator can add a step to the playbook (e.g., Create Ticket). This action will automatically generate a ticket in the external system and populate it with details from the alert, such as the finding, the affected resources, and the recommended remediation steps. This provides a seamless, automated workflow from detection to ticketing.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Use cases > Case Management

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations

Question: 48

You are an incident responder at your organization using Google Security Operations (SecOps) for monitoring and investigation. You discover that a critical production server, which handles financial transactions, shows signs of unauthorized file changes and network scanning from a suspicious IP

address. You suspect that persistence mechanisms may have been installed. You need to use Google SecOps to immediately contain the threat while ensuring that forensic data remains available for investigation. What should you do first?

A. Use the firewall integration to submit the IP address to a network block list to inhibit internet access from that machine.

B. Deploy emergency patches, and reboot the server to remove malicious persistence.

C. Use the EDR integration to quarantine the compromised asset.

D. Use VirusTotal to enrich the IP address and retrieve the domain. Add the domain to the proxy block list.

Answer: C

Explanation:

Comprehensive and Detailed Explanation

The correct answer is Option C. The prompt specifies two critical, simultaneous requirements: immediate containment and preservation of forensic data.

Immediate Containment: The server is actively scanning the network, so it must be taken ofline to prevent lateral movement and further compromise.

Forensic Preservation: The suspicion of persistence mechanisms means a full investigation is required. This investigation relies on volatile data (running processes, memory, active network connections) that must not be destroyed.

Option C is the only action that satisfies both requirements. Using a Google SecOps SOAR playbook to trigger the EDR integration's "quarantine" action instructs the EDR agent on the server to block all its network connections. This immediately contains the threat. However, the server itself remains running, which preserves all volatile forensic data for the investigation.

Option B (reboot) is incorrect because it is an eradication step that would destroy all volatile forensic evidence. Options A and D are incomplete containment or investigation steps that do not fully isolate the compromised host.

Exact Extract from Google Security Operations Documents:

Incident Response and Containment: When a critical asset is compromised, the first priority is containment. Google SecOps SOAR playbooks integrate with Endpoint Detection and Response (EDR) tools to automate this step.

EDR Integration Actions: The most common containment action is "Quarantine Host" or "Isolate

Asset." This action instructs the EDR agent on the endpoint to block all network communications, effectively isolating it from the rest of the network. This step immediately stops the threat from spreading or communicating with a C2 server. A key benefit of this approach, as opposed to a shutdown or reboot, is that the host remains powered on, which preserves volatile memory and process data for forensic investigation.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Playbooks > Playbook Actions

Google Cloud Documentation: Google Security Operations > Documentation > SOAR > Marketplace integrations > (e.g., CrowdStrike, SentinelOne, Microsoft Defender)

Question: 49

You are implementing Google Security Operations (SecOps) with multiple log sources. You want to closely monitor the health of the ingestion pipeline's forwarders and collection agents, and detect silent sources within five minutes. What should you do?

A. Create an ingestion notification for health metrics in Cloud Monitoring based on the total ingested log count for each collector_id.

B. Create a notification in Cloud Monitoring using a metric-absence condition based on sample policy for each collector_id.

C. Create a Looker dashboard that queries the BigQuery ingestion metrics schema for each log_type and collector_id.

D. Create a Google SecOps dashboard that shows the ingestion metrics for each iog_cype and collector_id.

Answer: B

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option B. This question requires a low-latency (5 minutes) notification for a silent source.

The other options are incorrect for two main reasons:

Dashboards vs. Notifications: Options C and D are incorrect because dashboards (both in Looker and Google SecOps) are for visualization, not active, real-time alerting. They show you the status when you look at them but do not proactively notify you of a failure.

Metric-Absence vs. Metric-Value: Google SecOps streams all its ingestion health metrics to Google Cloud Monitoring, which is the correct tool for real-time alerting. However, Option A is monitoring the "total ingested log count." This metric would require a threshold (e.g., count < 1), which can be problematic. The specific and most reliable method to detect a "silent source" (one that has stopped sending data entirely) is to use a metric-absence condition. This type of policy in Cloud Monitoring triggers only when the platform stops receiving data for a specific metric (grouped by collector_id) for a defined duration (e.g., five minutes).

Exact Extract from Google Security Operations Documents:

Use Cloud Monitoring for ingestion insights: Google SecOps uses Cloud Monitoring to send the ingestion notifications. Use this feature for ingestion notifications and ingestion volume viewing... You can integrate email notifications into existing workflows.

Set up a sample policy to detect silent Google SecOps collection agents:

In the Google Cloud console, select Monitoring.

Click Create Policy.

Select a metric, such as chronicle.googleapis.com/ingestion/log_count.

In the Transform data section, set the Time series group by to collector_id.

Click Next.

Select Metric absence and do the following:

Set Alert trigger to Any time series violates.

Set Trigger absence time to a time (e.g., 5 minutes).

In the Notifications and name section, select a notification channel.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Ingestion > Use Cloud Monitoring for ingestion insights

Question: 50

You have a close relationship with a vendor who reveals to you privately that they have discovered a vulnerability in their web application that can be exploited in an XSS attack. This application is running on servers in the cloud and on-premises. Before the CVE is released, you want to look for signs of the vulnerability being exploited in your environment. What should you do?

A. Create a YARA-L 2.0 rule to detect a time-ordered series of events where an external inbound connection to a server was followed by a process on the server that spawned subprocesses previously not seen in the environment.

B. Activate a new Web Security Scanner scan in Security Command Center (SCC), and look for findings related to XSS.

C. Ask the Gemini Agent in Google Security Operations (SecOps) to search for the latest vulnerabilities in the environment.

D. Create a YARA-L 2.0 rule to detect high-prevalence binaries on your web server architecture communicating with known command and control (C2) nodes. Review inbound traffic from those C2 domains that have only started appearing recently.

Answer: A

Explanation:

Comprehensive and Detailed Explanation

The correct solution is Option A. The key to this question is that the vulnerability is a zero-day (the CVE is not yet

released). Therefore, you cannot hunt for known signatures, and tools that rely on public intelligence are useless. The only way to find it is to hunt for the behavior or TTPs (Tactics, Techniques, and Procedures) of its exploitation.

A critical XSS attack can often be used to achieve Remote Code Execution (RCE). The logical TTP for this would be:

An external inbound connection to the web server (the exploit delivery).

This connection causes the web server process to spawn a new subprocess (the payload, e.g., a reverse shell, whoami, or powershell.exe).

Option A perfectly describes a behavioral YARA-L rule to detect this exact time-ordered series of events. By correlating an inbound NETWORK_CONNECTION with a subsequent PROCESS_LAUNCH from the same server and checking if that process is anomalous ("previously not seen"), you are effectively hunting for the post-exploitation behavior.

Option B is incorrect: WSS is a vulnerability scanner that looks for known classes of vulnerabilities. It will not find a specific, unknown zero-day.

Option C is incorrect: Gemini relies on public threat intelligence. If the CVE is not released, Gemini will not know about the vulnerability.

Option D is incorrect: This is a generic C2 detection and is less specific than Option A. An exploit would also likely use low-prevalence or unusual binaries, not "high-prevalence" ones.

Exact Extract from Google Security Operations Documents:

YARA-L 2.0 language overview: YARA-L 2.0 is a computer language used to create rules for searching through your enterprise log data... A typical multiple event rule will have the following: A match section which specifies the time range over which events need to be grouped. A condition section specifying what condition should trigger the detection and checking for the existence of multiple events.

This allows an analyst to hunt for specific TTPs by correlating a time-ordered series of events. For example, a rule can be written to join a NETWORK_CONNECTION event (e.g., an external inbound connection) with a subsequent PROCESS_LAUNCH event on the same host... By enriching this with entity context, the detection can be scoped to trigger only when the spawned process is anomalous or previously not seen in the environment, indicating a likely post-exploitation activity, such as a web shell or remote code execution resulting from an exploit.

Reference:

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Overview of the YARA-L 2.0 language

Google Cloud Documentation: Google Security Operations > Documentation > Detections > Context- aware analytics