PCDRA Q&As

Palo Alto Networks Certified Detection and Remediation Analyst

Pass Palo Alto Networks PCDRA Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/pcdra.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Palo Alto Networks Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

When creating a BIOC rule, which XQL query can be used?

A. dataset = xdr_data | filter event_sub_type = PROCESS_START and action_process_image_name ~= ".*?\.(?:pdf|docx)\.exe"

B. dataset = xdr_data | filter event_type = PROCESS and event_sub_type = PROCESS_START and action_process_image_name ~= ".*?\.(?:pdf|docx)\.exe"

C. dataset = xdr_data | filter action_process_image_name ~= ".*?\.(?:pdf|docx)\.exe" | fields action_process_image

D. dataset = xdr_data | filter event_behavior = true event_sub_type = PROCESS_START and action_process_image_name ~= ".*?\.(?:pdf|docx)\.exe"

Correct Answer: B

Explanation: A BIOC rule is a custom detection rule that uses the Cortex Query Language (XQL) to define the behavior or actions that indicate a potential threat. A BIOC rule can use the xdr_data and cloud_audit_log datasets and presets for these datasets. A BIOC rule can also use the filter stage, alter stage, and functions without any aggregations in the XQL query. The query must return a single field named action_process_image, which is the process image name of the suspicious process. The query must also include the event_type and event_sub_type fields in the filter stage to specify the type and sub-type of the event that triggers the rule. Option B is the correct answer because it meets all the requirements for a valid BIOC rule query. It uses the xdr_data dataset, the filter stage, the event_type and event_sub_type fields, and the action_process_image_name field with a regular expression to match any process image name that ends with .pdf.exe or .docx.exe, which are common indicators of malicious files. Option A is incorrect because it does not include the event_type field in the filter stage, which is mandatory for a BIOC rule query. Option C is incorrect because it does not include the event_type and event_sub_type fields in the filter stage, and it uses the fields stage, which is not supported for a BIOC rule query. It also returns the action_process_image field instead of the action_process_image_name field, which is the expected output for a BIOC rule query. Option D is incorrect because it uses the event_behavior field, which is not supported for a BIOC rule query. It also does not include the event_type field in the filter stage, and it uses the event_sub_type field incorrectly. The event_sub_type field should be equal to PROCESS_START, not true. References: Working with BIOCs Cortex Query Language (XQL) Reference

QUESTION 2

What should you do to automatically convert leads into alerts after investigating a lead?

A. Lead threats can\\'t be prevented in the future because they already exist in the environment.

B. Create IOC rules based on the set of the collected attribute-value pairs over the affected entities concluded during the lead hunting.

C. Create BIOC rules based on the set of the collected attribute-value pairs over the affected entities concluded during the lead hunting.

D. Build a search query using Query Builder or XQL using a list of IOCs.

Correct Answer: B

Explanation: To automatically convert leads into alerts after investigating a lead, you should create IOC rules based on the set of the collected attribute-value pairs over the affected entities concluded during the lead hunting. IOC rules are used to detect known threats based on indicators of compromise (IOCs) such as file hashes, IP addresses, domain

names, etc. By creating IOC rules from the leads, you can prevent future occurrences of the same threats and generate alerts for them. References: PCDRA Study Guide, page 25 Cortex XDR 3: Handling Cortex XDR Alerts, section 3.2 Cortex XDR Documentation, section "Create IOC Rules"

QUESTION 3

Under which conditions is Local Analysis evoked to evaluate a file before the file is allowed to run?

A. The endpoint is disconnected or the verdict from WildFire is of a type benign.

B. The endpoint is disconnected or the verdict from WildFire is of a type unknown.

C. The endpoint is disconnected or the verdict from WildFire is of a type malware.

D. The endpoint is disconnected or the verdict from WildFire is of a type grayware.

Correct Answer: B

Explanation: Local Analysis is a feature of Cortex XDR that allows the agent to evaluate files locally on the endpoint, without sending them to WildFire for analysis. Local Analysis is evoked when the following conditions are met: The endpoint is disconnected from the internet or the Cortex XDR management console, and therefore cannot communicate with WildFire. The verdict from WildFire is of a type unknown, meaning that WildFire has not yet analyzed the file or has not reached a conclusive verdict. Local Analysis uses machine learning models to assess the behavior and characteristics of the file and assign it a verdict of either benign, malware, or grayware. If the verdict is malware or grayware, the agent will block the file from running and report it to the Cortex XDR management console. If the verdict is benign, the agent will allow the file to run and report it to the Cortex XDR management console. References: Local Analysis WildFire File Verdicts

QUESTION 4

Which of the following represents the correct relation of alerts to incidents?

A. Only alerts with the same host are grouped together into one Incident in a given time frame.

B. Alerts that occur within athree-hourtime frame are grouped together into one Incident.

C. Alerts with same causality chains that occur within a given time frame are grouped together into an Incident.

D. Every alert creates a new Incident.

Correct Answer: C

Explanation: The correct relation of alerts to incidents is that alerts with same causality chains that occur within a given time frame are grouped together into an incident. A causality chain is a sequence of events that are related to the same malicious activity, such as a malware infection, a lateral movement, or a data exfiltration. Cortex XDR uses a set of rules that take into account different attributes of the alerts, such as the alert source, type, and time period, to determine if they belong to the same causality chain. By grouping related alerts into incidents, Cortex XDR reduces the number of individual events to review and provides a complete picture of the attack with rich investigative details1. Option A is incorrect, because alerts with the same host are not necessarily grouped together into one incident in a given time frame. Alerts with the same host may belong to different causality chains, or may be unrelated to any malicious activity. For example, if a host has a malware infection and a network anomaly, these alerts may not be grouped into the same incident, unless they are part of the same attack. Option B is incorrect, because alerts that occur within a three hour time frame are not always grouped together into one incident. The time frame is not the only criterion for grouping alerts

into incidents. Alerts that occur within a three hour time frame may belong to different causality chains, or may be unrelated to any malicious activity. For example, if a host has a file download and a registry modification within a three hour time frame, these alerts may not be grouped into the same incident, unless they are part of the same attack. Option D is incorrect, because every alert does not create a new incident. Creating a new incident for every alert would result in alert fatigue and inefficient investigations. Cortex XDR aims to reduce the number of incidents by grouping related alerts into one incident, based on their causality chains and other attributes. References: Palo Alto Networks Certified Detection and Remediation Analyst (PCDRA) Study Guide, page 9 Palo Alto Networks Cortex XDR Documentation, Incident Management Overview2 Cortex XDR: Stop Breaches with AI-Powered Cybersecurity1

QUESTION 5

Which statement best describes how Behavioral Threat Protection (BTP) works?

A. BTP injects into known vulnerable processes to detect malicious activity.

B. BTP runs on the Cortex XDR and distributes behavioral signatures to all agents.

C. BTP matches EDR data with rules provided by Cortex XDR.

D. BTP uses machine Learning to recognize malicious activity even if it is not known.

Correct Answer: D

Explanation: The statement that best describes how Behavioral Threat Protection (BTP) works is D, BTP uses machine learning to recognize malicious activity even if it is not known. BTP is a feature of Cortex XDR that allows you to define custom rules to detect and block malicious behaviors on endpoints. BTP uses machine learning to profile behavior and detect anomalies indicative of attack. BTP can recognize malicious activity based on file attributes, registry keys, processes, network connections, and other criteria, even if the activity is not associated with any known malware or threat. BTP rules are updated through content updates and can be managed from the Cortex XDR console. The other statements are incorrect for the following reasons: A is incorrect because BTP does not inject into known vulnerable processes to detect malicious activity. BTP does not rely on process injection, which is a technique used by some malware to hide or execute code within another process. BTP monitors the behavior of all processes on the endpoint, regardless of their vulnerability status, and compares them with the BTP rules. B is incorrect because BTP does not run on the Cortex XDR and distribute behavioral signatures to all agents. BTP runs on the Cortex XDR agent, which is installed on the endpoint, and analyzes the endpoint data locally. BTP does not use behavioral signatures, which are predefined patterns of malicious behavior, but rather uses machine learning to identify anomalies and deviations from normal behavior. C is incorrect because BTP does not match EDR data with rules provided by Cortex XDR. BTP is part of the EDR (Endpoint Detection and Response) capabilities of Cortex XDR, and uses the EDR data collected by the Cortex XDR agent to perform behavioral analysis. BTP does not match the EDR data with rules provided by Cortex XDR, but rather applies the BTP rules defined by the Cortex XDR administrator or the Palo Alto Networks threat research team. References: Cortex XDR Agent Administrator Guide: Behavioral Threat Protection Cortex XDR: Stop Breaches with AI-Powered Cybersecurity