DUMPSARENA

CrowdStrike Falcon Certification Program

CrowdStrike CCFA-200b

Version Demo

Total Demo Questions: 12

Total Premium Questions: 121

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Falcon Platform	23
Topic 2, Falcon Sensor Deployment	9
Topic 3, Falcon Sensor Management	38
Topic 4, Falcon Prevention	24
Topic 5, Falcon Detection and Response	27
Total	121

QUESTION NO: 1

How are custom roles assigned to users to perform a specific action on a module?

A. Users get all permissions by default

B. Permissions are enabled in roles, and these roles are assigned to users

C. By adding each module to a role

D. Permissions are assigned to users directly in user management

ANSWER: B

Explanation:

Permissions are enabled in roles, and these roles are assigned to users. In the Falcon console's role-based access control model, an administrator creates or edits a custom role by selecting the permission groups and specific actions the role should be allowed to perform. The administrator then assigns that role to the appropriate user account. This model ties a user's authorized actions to the roles assigned to them, allowing access to be scoped to operational responsibilities such as responding to detections, administering sensors, managing policies, or performing Real Time Response activities.

Using custom roles supports least-privilege administration because a role can contain only the permissions needed for a particular job function. It also makes access administration more consistent and auditable: when several users require the same level of access, the same defined role can be assigned to each of them rather than configuring permissions separately for every user. CrowdStrike documents the management of user roles and permissions through Falcon console user-management and role-based access-control workflows. See the CrowdStrike Falcon User Management documentation and the CrowdStrike Falcon platform overview.

QUESTION NO: 2

How are prevention policies assigned to hosts in the Falcon platform?

A. Through host group membership

B. Through direct host assignment

C. Through IP address ranges

D. Through manual configuration

ANSWER: A

Explanation:

Through host group membership is correct because Falcon applies prevention policy configuration at scale by targeting host groups. An administrator creates or identifies a static or dynamic host group, associates that group with the desired prevention policy, and endpoints that are members of the group receive the policy. This enables consistent security controls across systems with shared operating-system, business, location, tagging, or other grouping characteristics, rather than requiring policy administration endpoint by endpoint.

Host groups are therefore the assignment boundary used to determine which hosts receive a prevention policy. Falcon can evaluate dynamic group criteria continuously, so a host that meets the group's criteria becomes subject to the policy as its membership changes. Where more than one applicable policy relationship exists, Falcon uses its policy-assignment and precedence behavior to determine the effective configuration. This group-based approach supports centralized, repeatable administration and is fundamental to managing endpoint prevention controls in larger environments. CrowdStrike describes Falcon Prevent as its next-generation antivirus and endpoint protection capability at CrowdStrike Falcon Prevent. Additional official product context for endpoint protection is available in CrowdStrike's Endpoint Security overview.

QUESTION NO: 3

When would the No Action option be assigned to a hash in IOC Management?

A. When you want to save the indicator for later action, but do not want to block or allow it at this time

B. There is no such option as No Action available in the Falcon console

C. When you want to add the indicator to your allowlist, but not detect it

D. When you want to add the indicator to your blocklist and show it as a detection

ANSWER: A

Explanation:

The correct use of **No Action** is when an administrator wants to retain a hash indicator in IOC Management for future use without currently applying an enforcement or detection outcome. This lets security teams create, preserve, review, and organize known indicators while validation, investigation, change control, or a later response decision is still pending. The hash remains available in the Falcon console and can subsequently be updated with the appropriate action when the organization decides it should be blocked, detected, or explicitly allowed.

Using No Action is particularly helpful for threat-intelligence staging and operational workflows. For example, a team may receive a potentially relevant hash from an intelligence feed but need to confirm its relevance to its environment before making it active. Saving the indicator without action preserves the contextual record and avoids prematurely changing endpoint behavior. CrowdStrike's documentation describes custom indicators of compromise and their configurable actions in the Falcon console; administrators should select an action that matches the desired endpoint response and use no-action staging when no response is presently intended. See CrowdStrike's IOC Management overview and the Indicators of Compromise technical hub.

QUESTION NO: 4

Your development team is working on a new enterprise application, but Falcon starts creating alerts during testing. The alert points to "C:\Users\Bob\DevCode\felix.dll". In the detection, you see that it is triggering only on a specific Falcon IOA. What would be the best course of action for this situation?

A. Create an IOA exclusion for "C:\Users\Bob\DevCode\felix.dll"

B. Create a Custom IOC and set it to "Allow" for "C:\Users\Bob\DevCode\felix.dll"

C. Manually turn off the built-in IOA through prevention policies

D. Create a sensor visibility exclusion for "C:\Users\Bob\DevCode\felix.dll"

ANSWER: A

Explanation:

Create an IOA exclusion for "C:\Users\Bob\DevCode\felix.dll" because the alert is specifically associated with a Falcon Indicator of Attack rule. IOAs are behavioral detections: they identify activity patterns that may indicate malicious behavior rather than relying only on a static file identity. When testing a legitimate development artifact produces a repeatable false positive from one identified IOA, an exclusion can be scoped to that rule's relevant criteria and the known development path. This preserves the intended behavior of the application while minimizing the effect on Falcon's broader prevention and detection coverage. The exclusion should be narrowly defined and reviewed periodically, since development paths and application behavior may change. CrowdStrike describes IOAs as behavioral indicators used to identify attacker techniques and suspicious activity, which is why an IOA-specific exception is the appropriate remediation for a detection attributed to a particular IOA. See CrowdStrike's overview of Indicators of Attack and its guidance on endpoint detection and response.

QUESTION NO: 5

What is the recommended approach for managing host groups over time?

A. Create separate groups for each department

B. Create groups based on IP ranges

C. Maintain multiple overlapping host groups

D. Minimize the number of groups

ANSWER: D

Explanation:

Minimize the number of groups is the recommended long-term approach because Falcon host groups are a fundamental mechanism for targeting and assigning policies, including prevention policies, sensor update policies, firewall policies, and response-related workflows. A deliberately small group structure makes policy targeting clearer, simplifies administration, and reduces the likelihood that a host's membership in several groups leads to an unintended policy outcome. Effective group design should be driven by a stable operational purpose, such as platform, server versus workstation role, environment, or a specifically required security configuration. Where group membership can be determined consistently from host properties, dynamic grouping provides a maintainable way to keep membership current without recurring manual changes. As environments evolve, administrators should periodically review group purpose, membership criteria, and policy associations, then consolidate groups that no longer have a distinct operational need. This keeps the Falcon deployment understandable and auditable while preserving the flexibility required for exceptions and targeted controls. CrowdStrike documents host groups as the method used to organize hosts and apply policy settings, so a concise, governed group model supports reliable policy management over time. See the CrowdStrike Falcon Host and Host Group Management documentation and the CrowdStrike Falcon Prevent overview.

QUESTION NO: 6

What best describes the relationship between Sensor Update policies and Operating Systems?

A. A Sensor Update policy must be configured for each Operating System (Windows, Mac, Linux)

B. Sensor Update polices are not Operating System specific; one policy can be applied to all Operating Systems

C. Windows has its own Sensor Update polices; Mac and Linux share Sensor Update policies

D. Windows and Mac share Sensor Update policies; Linux requires its own set of polices based on the different kernel versions

ANSWER: A

Explanation:

Sensor Update policies are platform-specific, so a separate update-policy configuration is required for Windows, macOS, and Linux. This reflects the fact that Falcon sensor releases, version availability, and update packages are maintained independently for each supported operating-system platform. An administrator uses the relevant platform's Sensor Update policy to control such settings as the sensor version to deploy and the timing or behavior of sensor updates for hosts assigned to that policy. Host groups then provide the targeting mechanism for applying the appropriate policy to hosts within that platform. In practical terms, this means an organization can stage a Windows sensor release for a Windows host group while independently selecting and deploying the applicable macOS or Linux sensor release to their respective host groups. The policy model is therefore organized first by platform and then by the hosts or host groups that need a particular update cadence. This separation supports safe, controlled rollout management because each platform has its own sensor build and compatibility requirements. CrowdStrike's Falcon Sensor resources describe the sensor as a platform-specific endpoint component, and CrowdStrike's deployment guidance covers managing sensors across supported operating systems. See the CrowdStrike Falcon Sensor overview and the CrowdStrike endpoint security technical hub.

QUESTION NO: 7

Using Host setup and management inside the Falcon Console, how can you display sensors in Reduced Functionality Mode?

A. From Host management, filter for RFM

B. From Host status, filter for RFM

C. From Sensor health, sort using the column heading Sensor status

D. From Sensor status, click on the widget RFM

ANSWER: A

Explanation:

“From Host management, filter for RFM” is correct because the Falcon Console’s Host Management view supports filtering the host inventory by sensor-related attributes and states, including Reduced Functionality Mode (RFM). Applying the RFM filter narrows the host list to endpoints whose Falcon sensor is operating in that state, allowing an administrator to identify the affected systems and investigate them directly from the inventory workflow. This is the appropriate operational path because RFM is a sensor condition that must be located across managed hosts; Host Management is the console area designed for searching, filtering, and acting on those endpoint records. After the filtered list is displayed, the administrator can open individual host details to review sensor information and determine the remediation required for that device. RFM commonly indicates that the sensor cannot provide its full protection capability because an operating-system or kernel compatibility requirement is not met, so identifying these endpoints promptly is important for maintaining intended coverage. CrowdStrike’s product and documentation resources are available through the CrowdStrike Documentation portal and the CrowdStrike Support portal, where entitled users can consult the current console guidance and sensor support information.

QUESTION NO: 8

Which ML exclusion pattern would be the most accurate for all .exe binaries in “C:\Program Files\Software\”, including any subfolders of Software?

A. Program Files\Software* *.exe*

B. *Program Files\Software*.exe*

C. *Program Files\Software** *.exe

D. ***.exe

E. Program Files\Software***.exe

ANSWER: E

Explanation:

`Program Files\Software\**\*.exe` is the appropriate Machine Learning exclusion because it constrains the exclusion to executable files beneath the intended Software directory while also covering nested directories. Falcon exclusion patterns use glob-style matching. In this syntax, `Program Files\Software\` establishes the directory scope, `**\` recursively traverses all subfolders under that scope, and `*.exe` limits matching to files with the executable extension. Together, those elements express the required scope precisely: every executable in Software itself and every executable in any child directory, without extending the match to unrelated locations. For Windows ML exclusions, the path is expressed relative to the drive root in the policy pattern, so the drive designator and initial backslash from `C:\Program Files\Software\` are not included. This level of specificity is important when configuring prevention-policy exclusions because exclusions should be narrowly scoped to the required files and directories. Review the applicable exclusion and policy guidance in the CrowdStrike Documentation portal and the CrowdStrike Tech Hub.

QUESTION NO: 9

What log would you use to investigate unusual activity invoked with a script interfacing with the Falcon platform?

A. Falcon UI audit

B. RTR session audit

C. Prevention policy debug

D. API audit

ANSWER: D

Explanation:

API audit is the appropriate log for investigating unusual activity initiated by a script that interfaces with the Falcon platform. Scripts and other automated integrations generally authenticate to Falcon through API clients, using credentials such as a client ID and client secret or another supported API authentication mechanism. API audit records are designed to provide visibility into activity performed through those programmatic interfaces, allowing an administrator to investigate the identity associated with the API client, the timing of activity, and the operations requested through the Falcon API. This audit trail is especially important when activity appears unexpected, because it helps connect an action to the automation, integration, or API credential responsible for it. Reviewing API audit events also supports incident response and governance workflows: investigators can validate whether the behavior aligns with an approved integration and can identify the API client that requires remediation if credentials may have been misused. CrowdStrike documents its API access and authentication model in the Falcon API access guide and provides the available API operations and related platform resources through the CrowdStrike Developer Center OpenAPI documentation.

QUESTION NO: 10

What is the highest level of protection for a prevention policy?

A. Phase 1

B. Phase 2

C. Phase 3

ANSWER: C

Explanation:

Phase 3 is the highest protection level in Falcon's phased prevention-policy deployment approach. The phase model is designed to let administrators progressively strengthen endpoint prevention controls: organizations first validate behavior and operational impact, then increase enforcement as they gain confidence in policy tuning and exception handling. By the time a prevention policy reaches Phase 3, its settings represent the most protective posture intended for broad production enforcement.

This staged approach supports a practical balance between strong prevention and operational stability. Before moving endpoints to the highest-protection phase, administrators should review detections, investigate potential false positives, validate legitimate business applications, and apply narrowly scoped exclusions only where justified. This helps ensure that the Phase 3 policy can deliver maximum prevention coverage without unnecessarily disrupting approved workloads. Falcon Prevent is CrowdStrike's next-generation antivirus capability and provides the prevention controls administered through Falcon prevention policies. See CrowdStrike's Falcon Prevent product overview and the CrowdStrike Support Portal for product and policy-administration documentation.

QUESTION NO: 11

What default user role can manage API credentials?

A. Falcon Security Lead

B. Falcon Administrator

C. Falcon API Manager

D. Endpoint Manager

ANSWER: B

Explanation:

Falcon Administrator is the default Falcon role authorized to manage API credentials. API credentials, commonly created as API clients with a client ID and client secret, enable programmatic access to Falcon services for integrations, automation, and external tools. Because these credentials can be assigned API scopes that grant access to security data and administrative functions, their creation and lifecycle management require broad administrative authority.

The Falcon Administrator role is designed for full console administration and is therefore the appropriate default role for this sensitive capability. This includes managing the access configuration that supports API-based integrations, while ensuring credentials are governed through role-based access controls and least-privilege scope assignment. Administrators should create separate API clients for distinct integrations, grant only the required scopes, securely store client secrets, and rotate or revoke credentials when they are no longer needed. These practices reduce the impact of credential exposure and preserve accountability for API activity.

CrowdStrike's API guidance describes using API clients and scoped permissions for Falcon integrations. For additional context on Falcon APIs and credential setup, see CrowdStrike Developer Documentation and CrowdStrike's RBAC overview.

QUESTION NO: 12

When troubleshooting a Windows sensor that appears to be installed but is not running, what should be verified to ensure they are installed and running?

A. LMHosts and Windows Base Filtering Engine

B. Windows firewall and internet connectivity to the CrowdStrike cloud

C. Network Store Interface and Network List Service

ANSWER: A

Explanation:

LMHosts and Windows Base Filtering Engine is correct because the Falcon sensor for Windows relies on required underlying Windows services to support its networking and filtering functions. During troubleshooting, an administrator should confirm that these prerequisite services exist, are not disabled by policy or security hardening, and have an operational running state. The LMHosts service supports Windows name-resolution and network-related functionality that can be required by the sensor environment. Windows Base Filtering Engine is particularly important because it is the core Windows service that manages filtering-platform policy and supports components that interact with Windows networking and security controls. If a dependency is stopped, disabled, or fails to initialize, the sensor may appear installed while its service cannot start or operate normally. Verifying the service state in the Windows Services console, and checking related Windows Event Viewer entries for service-start or dependency errors, is an appropriate first diagnostic step. CrowdStrike's documentation portal provides product documentation and support material for sensor deployment and troubleshooting, while Microsoft documents the Base Filtering Engine service and its role in the Windows Filtering Platform. See CrowdStrike Tech Hub and Microsoft's Base Filtering Engine documentation.