

[Google Discussions](#)

Exam Professional Cloud Security Engineer All Questions

View all questions & answers for the Professional Cloud Security Engineer exam

[Go to Exam](#)

EXAM PROFESSIONAL CLOUD SECURITY ENGINEER TOPIC 1 QUESTION 223 DISCUSSION

Actual exam question from Google's Professional Cloud Security Engineer

Question #: 223

Topic #: 1

[\[All Professional Cloud Security Engineer Questions\]](#)

After completing a security vulnerability assessment, you learned that cloud administrators leave Google Cloud CLI sessions open for days. You need to reduce the risk of attackers who might exploit these open sessions by setting these sessions to the minimum duration.

What should you do?

- A. Set the session duration for the Google session control to one hour.
- B. Set the reauthentication frequency for the Google Cloud Session Control to one hour.
- C. Set the organization policy constraint constraints/iam.allowServiceAccountCredentialLifetimeExtension to one hour.
- D. Set the organization policy constraint constraints/iam.serviceAccountKeyExpiryHours to one hour and inheritFromParent to false.


[Show Suggested Answer](#)

by  Mithung30 at Aug. 4, 2023, 1:13 p.m.

Comments

Type your comment...

[Submit](#)

  **Pime13** 7 months, 3 weeks ago

Selected Answer: B

<https://support.google.com/a/answer/9368756?hl=en>

Reauthentication Frequency: Setting the reauthentication frequency ensures that users must re-authenticate after a specified period, in this case, one hour. This reduces the window of opportunity for an attacker to exploit an open session

A. Session Duration: While setting the session duration can help, reauthentication frequency is more directly related to ensuring users re-authenticate regularly.

C. Service Account Credential Lifetime: This constraint is specific to service account credentials and does not directly address user session durations.

D. Service Account Key Expiry: Similar to option C, this focuses on service account keys rather than user session management.

   upvoted 1 times

  **MoAk** 8 months, 1 week ago

Selected Answer: B

As of late, it appears that answer B is the only correct answer.

https://support.google.com/a/answer/7576830?hl=en&ref_topic=7556597&sjid=10540575594857625427-EU

   upvoted 1 times

  **Betotoxicity** 1 year, 3 months ago

Selected Answer: D

D:

Granular Control: This policy constraint specifically targets serviceAccountKeyExpiryHours, directly controlling how long service account credentials (used by the Cloud CLI) remain valid.

Minimum Duration: Setting the expiry to one hour enforces session termination after that timeframe, mitigating the risk of open sessions being exploited.

Inheritance Override: Using inheritFromParent: false ensures this policy applies to the specific organization, preventing accidental overrides from higher levels in the hierarchy.

Why not B?: Reauthentication Frequency: This might prompt users to re-authenticate within the console but doesn't directly terminate open Cloud CLI sessions.

   upvoted 1 times

  **MMNB2023** 1 year, 8 months ago

Selected Answer: B

1 hour as min duration and max 24hours.

   upvoted 3 times

  **MisterHairy** 1 year, 8 months ago

Selected Answer: B

The best option would be B. Set the reauthentication frequency for the Google Cloud Session Control to one hour.

This is because Google Cloud Session Control allows you to set a reauthentication frequency, which determines how often users are prompted to reauthenticate during their session. By setting this to one hour, you ensure that CLI sessions are only open for a maximum of one hour without reauthentication, reducing the risk of attackers exploiting these open sessions.

Option A is incorrect because there is no such thing as a "Google session control". Option C and D are related to service account keys and credential lifetime extension, not user sessions in the Google Cloud CLI.

   upvoted 3 times

  **alvinlxw** 1 year, 8 months ago

Selected Answer: B

<https://cloud.google.com/blog/products/identity-security/improve-security-posture-with-time-bound-session-length>



   upvoted 1 times

  **ArizonaClassics** 1 year, 10 months ago

B. Set the reauthentication frequency for the Google Cloud Session Control to one hour.

Option B is the correct approach because by setting the reauthentication frequency to one hour, you're ensuring that any active sessions automatically require reauthentication after that time period, mitigating the risk associated with long-lived sessions.

   upvoted 1 times

  **desertlotus1211** 1 year, 10 months ago

Answer B:



<https://support.google.com/a/answer/9368756?hl=en>

Get session length for Google Cloud sessions

Set session length for Google Cloud services

Answers A & B are a play on words... In order to do session during (A), you must adjust the reauthenticate policy duration (B).



   upvoted 1 times

  **GCBC** 1 year, 11 months ago

Selected Answer: A

session length to 1 hour is good other options are disturbing and expiring or reauthenticate every hour is not good for user experience

   upvoted 2 times

  **BR1123** 1 year, 11 months ago

D. By setting the organization policy constraint constraints/iam.serviceAccountKeyExpiryHours to one hour and inheritFromParent to false, you are specifically controlling the duration for which the service account keys (credentials) are valid. This directly addresses the issue of open sessions and the risk of exploitation by ensuring that the credentials used for these sessions expire after a shorter time, reducing the window of opportunity for attackers.

In summary, option D provides a more targeted approach to mitigating the risk posed by open Google Cloud CLI sessions by setting the service account key expiry duration to one hour and ensuring it doesn't inherit from parent policies.



   upvoted 1 times

  **anshad666** 1 year, 11 months ago

Selected Answer: B

https://support.google.com/a/answer/9368756?hl=en&ref_topic=7556597&sjid=4209356388025132107-AP



   upvoted 3 times

  **cyberpunk21** 1 year, 11 months ago

Selected Answer: A

A and B both satisfies the question but the effective and easy to do will be A, BTW B does the same job

   upvoted 1 times

  **desertlotus1211** 1 year, 10 months ago

B is what you need to do....

   upvoted 1 times

  **RuchiMishra** 1 year, 11 months ago

Selected Answer: B

<https://support.google.com/a/answer/9368756?hl=en>

   upvoted 3 times

  **gcp4test** 1 year, 11 months ago

Selected Answer: A

C,D serviceAccountKeyExpiryHours is for Service Account not human (users) - as in the question.
B - reauthenticate it is not user friendly, to reauthenticate user once every hour

So correct is A.

   upvoted 3 times

  **pfilourenco** 1 year, 11 months ago

The session-length control settings affect sessions with all Google web properties that a user accesses while signed in. I think B is the most appropriated:

" for Google Cloud tools, and how these controls interact with the parent session control on this page, see Set session length for Google Cloud services."

<https://support.google.com/a/answer/7576830?hl=en>

<https://support.google.com/a/answer/9368756?hl=en>

   upvoted 3 times

  **Mithung30** 1 year, 11 months ago

D. Set the organization policy constraint constraints/iam.serviceAccountKeyExpiryHours to one hour and inheritFromParent to false. This will set the default expiry time for service account keys to one hour and prevent the keys from being inherited from parent organizations.

In this case, the best option is to set the organization policy constraint constraints/iam.serviceAccountKeyExpiryHours to one hour and inheritFromParent to false. This will ensure that all service account keys expire after one hour and cannot be inherited from parent organizations. This will help to reduce the risk of attackers who might exploit open sessions.

   upvoted 2 times



Platform

> [Home](#)

> [Examtopics PRO](#)

> [All Exams](#)

> [Training Courses](#)

