Okta Administrator Certification Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Questions

1. **Is a registered but unmanaged macOS device a prerequisite for Okta Verify integration?**

   A. Yes, it is a valid device type

   B. No, it is not valid

   C. Only if it is connected to the internet

   D. It must be managed to integrate

2. **What is the main focus of the "Minimum password age" policy?**

   A. Ensure password strength

   B. Limit frequent password changes

   C. Control password expiration dates

   D. Regulate password history

3. **Which feature is supported by the Okta org authorization server but not by the Okta custom Authorization server?**

   A. Use Okta Developer SDKs and widgets for SSO

   B. Mint Access Tokens with Okta API Scopes

   C. Configure Custom Login Pages

   D. Integrate with LDAP directories

4. **What process should an Okta administrator use for importing users?**

   A. Bulk import with CSV files

   B. Single user API integration

   C. Manual user creation only

   D. File transfer protocol (FTP)

5. **What role does an Okta Administrator need to create other Okta administrators?**

   A. Super Administrator

   B. Help Desk Administrator

   C. Application Administrator

   D. Group Administrator

6. What is the expected behavior when an application is enabled as a source?

A. Okta provisions users to the application

B. The application provisions users to Okta

C. Okta syncs with external directories

D. Users manage their own credentials

7. Is OIDC an appropriate sign-on method for a web application that doesn't support federation in Okta?

A. Yes

B. No

C. Only in specific cases

D. Only if they’re both managed

8. What is a key benefit of using SAML for authentication and authorization?

A. It allows users to access legacy web applications

B. It reduces the attack surface for organizations

C. It provides real-time monitoring of user activities

D. It eliminates the need for single sign-on

9. Does Okta service handle the Kerberos validation in Agentless Desktop Single Sign-on (DSSO)?

A. Yes

B. No

C. Only with certain configurations

D. Only in cloud environments

10. How can an Okta Administrator enforce user step-up authentication for a remote workforce?

A. By creating or modifying password policies

B. By creating or modifying global session policies

C. By restricting application access based on location

D. By implementing biometric verification methods

Answers

1. B
2. B
3. B
4. A
5. A
6. B
7. B
8. B
9. A
10. B

Explanations

1. Is a registered but unmanaged macOS device a prerequisite for Okta Verify integration?

A. Yes, it is a valid device type

B. No, it is not valid

C. Only if it is connected to the internet

D. It must be managed to integrate

A registered but unmanaged macOS device is not a valid prerequisite for Okta Verify integration because Okta Verify requires devices to be managed to ensure enhanced security and compliance. Only managed devices are enrolled in a mobile device management (MDM) solution, which allows for the application of security policies, device tracking, and remote control capabilities. Unmanaged devices lack these controls and may pose a higher risk, making them unsuitable for integrating with services like Okta Verify that are built with security protocols in mind. Managed devices provide a level of assurance for security, ensuring that the device complies with corporate policies and can be trusted to access secure applications. This is crucial in an enterprise environment where access controls and data security are paramount. Therefore, to implement Okta Verify successfully, the devices must be managed, reinforcing the importance of having a structured and secure device management policy within the organization.

2. What is the main focus of the "Minimum password age" policy?

A. Ensure password strength

B. Limit frequent password changes

C. Control password expiration dates

D. Regulate password history

The "Minimum password age" policy is primarily concerned with limiting how frequently users can change their passwords. By setting a minimum password age, organizations can prevent users from changing their passwords multiple times in a short period to bypass password history requirements. This is important for maintaining the integrity of password policies and encouraging users to adopt and retain strong passwords for a longer duration. The idea behind this policy is to strike a balance between user convenience and security. Allowing users to change passwords too frequently can enable them to quickly revert to a previously used password that may not be secure, especially if a password is weak. Therefore, enforcing a minimum duration before a password can be changed helps to enhance overall security by compelling users to remain with a given password longer and, ideally, ensure that they are creating more complex and secure passwords when they do change them. While other policies focus on aspects like password strength, expiration dates, or maintaining a history of used passwords, the minimum password age specifically deals with the frequency of changes, underlining its primary focus in managing password practices effectively.

3. Which feature is supported by the Okta org authorization server but not by the Okta custom Authorization server?

A. Use Okta Developer SDKs and widgets for SSO

B. Mint Access Tokens with Okta API Scopes

C. Configure Custom Login Pages

D. Integrate with LDAP directories

The choice regarding minting Access Tokens with Okta API Scopes is supported by the Okta org authorization server because it is designed to work seamlessly with OAuth 2.0 and OpenID Connect standards, which include the use of API scopes for defining the permissions associated with Access Tokens. This feature allows organizations to have fine-grained access control over the resources they protect with the tokens. The Okta org authorization server is the default server that comes pre-configured with the necessary features for developers and enterprises looking to implement an identity and access management solution quickly and efficiently. It is fully integrated with Okta's capabilities to handle API tokens and negotiate scopes that correspond to various actions or data sets across APIs. In contrast, while custom authorization servers offer flexibility and the ability to tailor authorization flows and implementations, they focus more on customized and specific use cases in a way that might not include the standardized minting of Access Tokens with API Scopes as the primary implementation. It is important to recognize how each feature aligns with the intended usage of the service being discussed. The org authorization server prioritizes ease of use and compliance with standard practices, making it the appropriate choice for minting Access Tokens in a way that adheres to established protocols.

4. What process should an Okta administrator use for importing users?

A. Bulk import with CSV files

B. Single user API integration

C. Manual user creation only

D. File transfer protocol (FTP)

The process of bulk importing users with CSV files is an efficient method used by Okta administrators to add multiple users at once. This approach allows administrators to prepare a single file containing user details such as usernames, emails, and profile attributes in a structured format. Once the CSV file is created, it can be uploaded to Okta, significantly streamlining the process compared to creating users individually. Utilizing CSV files for bulk imports is particularly beneficial when onboarding a large number of users or during migrations from another system, as it reduces administrative overhead and minimizes potential errors associated with entering data manually. Additionally, this process can be automated or integrated into existing workflows, further enhancing efficiency. Other methods, such as single user API integration and manual user creation, generally cater to situations that require immediate or specific user setups but do not scale effectively for large volumes. File transfer protocol (FTP) is not typically used for user imports in Okta, as the platform relies primarily on structured data formats like CSV for such tasks.

5. What role does an Okta Administrator need to create other Okta administrators?

A. Super Administrator

B. Help Desk Administrator

C. Application Administrator

D. Group Administrator

The Super Administrator role in Okta has the highest level of privileges and permissions within the organization. This role is specifically designed to manage the Okta environment comprehensively, including the authority to create, modify, and delete other administrator accounts. Given the responsibilities of a Super Administrator, they can assign various administrative roles to other users, thereby enabling or restricting access to Okta functionalities based on organizational needs. This capability is crucial for maintaining the security and integrity of the identity management system within an organization, as it ensures that only qualified individuals can hold administrative privileges and perform critical tasks. In contrast, the other roles—Help Desk Administrator, Application Administrator, and Group Administrator—have limited scopes and do not possess the same level of authority to create or manage additional administrators. Each of these roles is tailored for specific tasks within the Okta platform, such as managing user support, overseeing applications, or controlling group memberships, but they do not have the overarching permissions that a Super Administrator has.

6. What is the expected behavior when an application is enabled as a source?

A. Okta provisions users to the application

B. The application provisions users to Okta

C. Okta syncs with external directories

D. Users manage their own credentials

When an application is enabled as a source in Okta, it indicates that the application will be responsible for provisioning users into Okta. In this setup, Okta acts as a consumer of identity information provided by the application. This means that the application has the authority and mechanism to create, update, or deactivate user accounts directly in Okta based on its own user management processes. Enabling an application as a source allows organizations to streamline user management by having the application push identity details into Okta, reducing the need for manual entry and ensuring alignment between user data in the application and in Okta. This is especially useful in environments where user permissions and attributes may be dynamically adjusted within the application itself, further enhancing security and efficiency in user lifecycle management. The implications of this setup can be significant, especially for organizations using multiple applications as their source of truth. Instead of relying on Okta to provision users, the application takes the lead, facilitating greater integration and automation of identity processes.

7. Is OIDC an appropriate sign-on method for a web application that doesn't support federation in Okta?

A. Yes

B. No

C. Only in specific cases

D. Only if they’re both managed

OpenID Connect (OIDC) is indeed a federation-based sign-on method designed to enable application authentication using identity providers like Okta. It operates as an identity layer on top of the OAuth 2.0 protocol, allowing users to authenticate and providing additional user identity information. If a web application does not support federation, it means that the application does not have the capability to participate in federated identity management protocols and cannot delegate the user authentication process to an identity provider such as Okta. In this context, OIDC cannot be utilized effectively because it fundamentally relies on having an identity provider that can authenticate and manage user sessions. Therefore, using OIDC would not be feasible since the web application would not be able to process the federated authentication. The appropriate sign-on methods for applications unable to support federation are typically simpler, traditional authentication mechanisms such as username and password, rather than relying on federation-based standards like OIDC.

8. What is a key benefit of using SAML for authentication and authorization?

A. It allows users to access legacy web applications

B. It reduces the attack surface for organizations

C. It provides real-time monitoring of user activities

D. It eliminates the need for single sign-on

Using SAML (Security Assertion Markup Language) for authentication and authorization significantly reduces the attack surface for organizations. This occurs because SAML facilitates a single sign-on (SSO) experience, allowing users to authenticate once and gain access to multiple applications without needing to enter their credentials repeatedly. By minimizing the number of times users input their credentials, SAML inherently reduces the risk of credentials being intercepted during transmission or captured through phishing attacks. Additionally, SAML permits the use of robust authentication methods and user identity verification processes, which enhances security. By handling the authentication process through a centralized identity provider, organizations can implement strong security measures at that point, rather than relying on each application to enact its own, which might lead to inconsistencies in security practices across the board. This is different from allowing access to legacy apps, monitoring user activities, or eliminating the need for SSO, which either do not align with the primary benefits SAML offers or directly contradict its functionality.

9. Does Okta service handle the Kerberos validation in Agentless Desktop Single Sign-on (DSSO)?

A. Yes

B. No

C. Only with certain configurations

D. Only in cloud environments

In the context of Agentless Desktop Single Sign-On (DSSO), Okta indeed handles Kerberos validation. This feature is essential for enabling seamless SSO experiences for users accessing applications that rely on Kerberos authentication. By managing Kerberos ticketing and validation, Okta allows users to authenticate to applications without needing to manually enter credentials after their initial login. This capability is a core part of how Okta integrates with on-premises environments, ensuring that organizations can provide a smooth and cohesive authentication experience across various platforms and applications—effectively bridging the gap between cloud and on-premises systems. With the support for Kerberos validation, organizations can leverage existing identity providers and infrastructure while still benefiting from Okta's powerful identity management tools. The other options suggest limitations or conditions where Kerberos validation would not be handled by Okta, which does not reflect the service's actual capabilities in this area.

10. How can an Okta Administrator enforce user step-up authentication for a remote workforce?

A. By creating or modifying password policies

B. By creating or modifying global session policies

C. By restricting application access based on location

D. By implementing biometric verification methods

Enforcing user step-up authentication for a remote workforce involves ensuring that additional layers of security are applied when users access sensitive resources, particularly from locations or devices that are considered high-risk. This approach helps mitigate threats associated with remote access. Creating or modifying global session policies is an effective way to enforce step-up authentication. These policies can specify conditions under which additional verification measures are required, such as when a user is accessing the organization's resources from an unrecognized location or device. By configuring these policies, the administrator can mandate that users provide additional authentication factors — for instance, through multifactor authentication (MFA) — whenever they access critical applications remotely. This ensures that even if a user's password is compromised, an additional layer of security is in place to protect sensitive data. In contrast, while modifying password policies may enhance overall security, it does not directly pertain to step-up authentication, as it does not necessarily introduce additional verification during a session. Restricting application access based on location is more about access control than step-up authentication itself, as it might simply block access rather than require additional verification. Implementing biometric verification methods would also provide a form of authentication, but it's not a method specifically tailored to handle the step-up process within the existing