PassQueen
a.k.a dumpsqueen.com

Certified ProfessionalPingAM Exam

Ping Identity PT-AM-CPE

Version Demo

Total Demo Questions: 9

Total Premium Questions: 92

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

Which of the following parameters must be provided by the edge client when requesting step-up authentication or transactional authorization?

A. authIndexType and authIndexValue

B. service, authIndexType, and authIndexValue

C. ForceAuth, authIndexType, and authIndexValue

D. service and ForceAuth

ANSWER: A

Explanation:

authIndexType and authIndexValue are the required pair used to direct PingAM to the requested authentication target. The edge client supplies `authIndexType` to state how the target is being selected, such as by authentication service, module, or authentication level. It then supplies `authIndexValue` with the corresponding value, for example the name of an authentication tree or chain when the type is `service`, or the required level when the type is `level`.

This pairing is what lets PingAM identify the authentication flow needed for a step-up request or the flow that protects a transaction. It is also flexible: a client can select an appropriate type for the use case rather than always selecting a service. In REST authentication requests, these values are passed as request parameters to select the authentication mechanism instead of relying solely on the realm's default authentication configuration. PingAM documents authentication indexing and the supported index types in its REST authentication guidance: PingAM REST authentication documentation. The parameters are also part of the authentication request model described in the PingAM Authentication REST API reference.

QUESTION NO: 2

Which authentication node can you use in PingAM to add a key:value property to the user's session after successful authentication?

A. The Get Session Data node

B. You have to use a webhook, not a node

C. The Provision Dynamic Account node

D. The Set Session Properties node

ANSWER: D

Explanation:

The Set Session Properties node is designed to write custom key:value properties into the current user session during an authentication journey. You configure the node with one or more property names and values; when the journey succeeds and the session is created or updated, those values are available as session properties. This is useful when later journey steps, applications, policy conditions, or session-related logic need context established during authentication, such as an authentication method indicator, a user classification, a selected service level, or an application-specific attribute. Property values can also be populated dynamically by referencing values held in the shared state, allowing information collected or calculated earlier in the journey to be persisted in the session. The node therefore directly meets the requirement to add a key:value property after successful authentication, without requiring external integration. PingAM documents this node as setting session properties on the user's session, and its session-property documentation describes how these values are stored and accessed in session context. See the PingAM Set Session Properties node reference and PingAM session properties documentation.

QUESTION NO: 3

What authentication tree nodes are provided for device registration in PingAM?

A. MFA Registration Options node, OATH Registration node, Push Registration node

B. MFA Registration Options node, OATH Registration node, WebAuthn Registration node

C. MFA Registration Options node, Push Registration node, WebAuthn Registration node

D. OATH Registration node, Push Registration node, WebAuthn Registration node

ANSWER: D

Explanation:

The correct set is **OATH Registration node, Push Registration node, WebAuthn Registration node**. PingAM provides these dedicated authentication-tree nodes to enroll a user's device or authenticator for the corresponding multifactor authentication method. The OATH Registration node registers an OATH-compatible one-time-password authenticator, enabling subsequent OTP verification. The Push Registration node enrolls a device for push-based authentication, associating the PingID/PingOne MFA push capability with the user's account as configured in the tree. The WebAuthn Registration node performs WebAuthn credential enrollment, allowing a platform authenticator or security key to be registered for later passwordless or phishing-resistant authentication. These nodes are used during an enrollment journey before the related authentication nodes can verify the registered factor. PingAM's authentication-node documentation categorizes these as registration nodes and describes their configuration and outcomes in the context of authentication trees. See the PingAM authentication nodes documentation and the PingAM multifactor authentication documentation.

QUESTION NO: 4

When making a token exchange request for an ID token using the /oauth2/access_token endpoint, what is the value for the grant_type parameter?

A. urn:ietf:params:oidc:grant-type:token-exchange

B. urn:ietf:params:oauth2:grant-type:token-exchange

C. urn:ietf:params:oauth:grant-type:token-exchange

D. urn:ietf:params:oauth:grant-type:idtoken-exchange

ANSWER: C

Explanation:

The correct value is `urn:ietf:params:oauth:grant-type:token-exchange`. PingAM's token exchange capability implements the OAuth 2.0 Token Exchange grant defined by RFC 8693. That specification registers this exact URN as the grant type identifier, including `oauth` rather than `oauth2`. The same grant type is used regardless of whether the exchanged token requested by the client is an access token, refresh token, or ID token; the desired token format is specified separately through the `requested_token_type` parameter. For an ID token, that parameter is typically `urn:ietf:params:oauth:token-type:id_token`, while the grant type remains the standard token-exchange URN. A token exchange request also supplies a `subject_token` and its corresponding `subject_token_type`, enabling PingAM to evaluate the presented token and issue an appropriate exchanged token under the configured policies. RFC 8693 explicitly defines the grant-type value in its token exchange request parameters, and Ping Identity documents token exchange as an OAuth 2.0 token endpoint operation. See RFC 8693: OAuth 2.0 Token Exchange and PingAM Token Exchange documentation.

QUESTION NO: 5

An OpenID Connect client includes a cryptographically random `nonce` value in an authorization request. Which validation must the client perform when it receives the ID token?

A. Verify that the nonce equals the authorization code

B. Verify that the nonce equals the client secret

C. Verify that the ID token nonce matches the value sent in the authorization request

D. Verify that the nonce equals the session cookie value

ANSWER: C

Explanation:

The client must verify that the `nonce` claim in the ID token matches the nonce value that it created and retained for the authorization request. This associates the ID token with the client-initiated authentication transaction.

The `state` parameter is primarily used to correlate the authorization response and mitigate CSRF, whereas the nonce is used to mitigate replay of an ID token. The client identifier is validated against the token's `aud` claim, not its nonce claim.

QUESTION NO: 6

A single-page application uses the authorization code grant with PKCE. It successfully receives an authorization code, but the token request fails because the value used to verify PKCE does not match the authorization request. Which value must the application send to the token endpoint?

A. The original `code_challenge`

B. The client secret

C. The original `code_verifier`

D. The authorization code as the `state` value

ANSWER: C

Explanation:

The application must send the original `code_verifier`. Before redirecting the user to PingAM, the client creates a high-entropy verifier and derives a `code_challenge` from it. The challenge is sent to the authorization endpoint, while the original verifier is retained by the client.

When redeeming the authorization code, the client sends the `code_verifier`. PingAM derives the expected challenge and compares it with the challenge associated with the authorization code. Sending the challenge again, a client secret, or the authorization code itself does not satisfy PKCE verification.

QUESTION NO: 7

An administrator has a requirement to reconfigure the attribute used to search for users in a LDAP Data Store. What Data Store configuration attribute would they need to change?

A. LDAP Users Search Attribute

B. LDAP Users Index Attribute

C. LDAP Users Bind Attribute

D. LDAP Users Find Attribute

ANSWER: A

Explanation:

LDAP Users Search Attribute is the LDAP Data Store setting that controls which LDAP attribute PingAM uses when it searches for a user entry. This value is typically an attribute that uniquely identifies users in the directory, such as `uid` or `cn`, depending on the directory schema and the organization's chosen login identifier. When the required lookup attribute changes—for example, when users should be found using an employee ID rather than a user ID—the administrator updates this setting so PingAM constructs its LDAP user searches with the intended attribute. The configured attribute must exist on the relevant user entries and should be indexed by the LDAP server where possible, particularly in larger directories, to support efficient user lookup operations. This setting is part of the LDAP user configuration for the data store and directly determines the attribute used for these searches; it is therefore the appropriate configuration point for the stated requirement. PingAM's documentation describes LDAP user data-store configuration and the directory-based user lookup model in its administration and setup documentation. See the PingAM documentation and the LDAP data store setup documentation.

QUESTION NO: 8

Which of the following multi-factor authentication protocols are supported by PingAM?

A) Open authentication

B) Security questions

C) Web authentication

D) Universal 2nd factor authentication

E) Push authentication

A. B, C, and D

B. A, B, and E

C. A, C, and E

D. A, B, and C

ANSWER: C

Explanation:

A, C, and E is correct. PingAM supports OATH-based authentication, WebAuthn, and push authentication as MFA capabilities that can be composed in authentication journeys (trees). "Open authentication" in this question refers to OATH, the open standard used for one-time-password factors such as time-based OTP and HMAC-based OTP. PingAM provides OATH-related nodes for registering and verifying these OTP credentials.

Web authentication refers to WebAuthn, the modern FIDO authentication standard that enables phishing-resistant authentication using platform authenticators, biometrics, or external security keys. PingAM provides WebAuthn registration and authentication functionality for use as a strong additional factor or passwordless factor.

Push authentication is also supported through PingAM's push-based authentication capabilities and the Ping Identity/ForgeRock Authenticator application. A journey can send an approval request to an enrolled device and use the user's approval, typically together with device binding and cryptographic verification, to complete the authentication step. These supported methods let administrators build MFA flows appropriate to the organization's assurance and user-experience requirements. See the PingAM MFA documentation and the PingAM WebAuthn documentation.

QUESTION NO: 9

A PingAM service provider receives a SAML2 assertion whose AudienceRestriction does not contain the service provider's entity ID. What should the service provider do?

A. Accept the assertion if its signature is valid

B. Accept the assertion if the user exists in the local user store

C. Reject the assertion because this service provider is not an intended audience

D. Replace the assertion audience with the local service provider entity ID

ANSWER: C

Explanation:

The service provider should reject the assertion. The AudienceRestriction identifies the relying party or parties for which the assertion was issued. PingAM must verify that its own service provider entity ID is an intended audience before accepting the assertion.

A valid XML signature alone does not make an assertion acceptable to every service provider that trusts the identity provider. The assertion can be correctly signed and unexpired while still being intended for a different relying party.