Ping Identity PT-AM-CPE Practice Questions

Shared by Pacheco on 21-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which of the following components is used to return data to PingGateway or the agent to be included with the policy decision?

Options:

A- Subjects
B- Resources
C- Response attributes
D- Actions

Answer:

C

Explanation:

When PingAM 8.0.2 evaluates an authorization policy, the primary output is a 'Permit' or 'Deny' decision. However, applications and Policy Enforcement Points (PEPs)---like PingGateway or a Web Agent---often require additional metadata about the user or the session to function correctly (e.g., the user's employee ID, department, or a specific preference).

According to the PingAM documentation on 'Policies' and 'Requesting Decisions':

The mechanism used to provide this extra information is Response Attributes. When defining a policy in the PingAM UI or via REST, an administrator can configure 'Response Attributes' which map internal attributes (from the User Profile or the Session) to keys that are sent back in the policy decision payload.

How it works: If a policy is configured with a response attribute mapping uid to User-ID, when PingGateway asks 'Can user X access resource Y?', PingAM responds with 'Permit' AND a map containing User-ID: X.

Consumption: PingGateway or the Web Agent can then take these attributes and inject them into HTTP headers (e.g., X-User-ID) so the downstream application can consume them without having to query AM again.

Subjects (Option A), Resources (Option B), and Actions (Option D) are all input components used to define the scope of a policy; they are not used to return data to the enforcer. Only Response Attributes serve the purpose of enriching the decision response with additional context.

==========

Question 2

Question Type: MultipleChoice

Examine the following JWT client assertion in JSON format. From the subset of claims listed below, which claim can be optional?

JSON

JSON

{

"iss": "myClient",

"sub": "myClient",

"aud": "https://am.example.com/login/oauth2/access_token",

"jti": "id012345",

"exp": 1633363568,

"iat": 1633356368

}

Options:

A- jti
B- aud
C- iss
D- sub

Answer:

A

Explanation:

When an OAuth2 client uses Private Key JWT or Client Secret JWT for authentication at the PingAM 8.0.2 token endpoint, it must present a JWT (JSON Web Token) containing specific claims that identify and authorize the client. This is governed by the OIDC and OAuth2 JWT Profile specifications (RFC 7523).

According to the PingAM documentation on 'OAuth 2.0 Client Authentication' and the 'JWT Profile for Client Authentication':

iss (Issuer): Mandatory. This must be the client_id of the OAuth2 client.

sub (Subject): Mandatory. This must also be the client_id of the OAuth2 client (as the client is the subject of the authentication).

aud (Audience): Mandatory. This must be the URL of the PingAM OAuth2 service (the token endpoint) or the issuer URL.

exp (Expiration Time): Mandatory. This protects against the long-term use of intercepted assertions.

The jti (JWT ID) (Option A) provides a unique identifier for the token. In the context of standard JWT validation, jti is used to prevent replay attacks by ensuring that a specific token is only processed once. While highly recommended for security hardening, the PingAM 8.0.2 technical reference for OAuth2 client assertions marks jti as optional unless the server is explicitly configured to require it for replay detection. Without a jti, PingAM will still validate the iss, sub, aud, and exp claims to authenticate the client. Therefore, among the choices provided, jti is the claim that can be omitted without inherently violating the base OAuth2 JWT authentication request requirements.

==========

Question 3

Question Type: MultipleChoice

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

Options:

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:

In PingAM 8.0.2 Intelligent Access, the Set Session Properties node is a specialized utility node designed to modify the session object once it is created.

According to the 'Authentication Node Reference':

During an authentication journey, data is typically stored in the sharedState. However, sharedState is transient and is destroyed once the tree finishes. If an administrator wants to take a piece of information (e.g., a 'Risk Score' calculated during the tree, or a 'Branch ID' retrieved from a legacy system) and make it a permanent part of the user's session, they must use the Set Session Properties node.

Functionality: This node allows you to map a value from the sharedState or transientState to a session property name. After the tree reaches a Success node, these properties are persisted in the session (either in the CTS for server-side sessions or the JWT for client-side sessions).

Usage: Once set, these properties can be retrieved later for Response Attributes in policies, or by applications using the /json/sessions endpoint.

Option A (Get Session Data node) is used to retrieve existing properties from an active session, not set them. Option B is incorrect because while webhooks can trigger external logic, the native way to modify the session within a tree is a node. Option C (Provision Dynamic Account node) is for creating user entries in the Identity Store (LDAP), not for managing session-level properties. Therefore, Set Session Properties (Option D) is the correct technical tool for this requirement in version 8.0.2.

Question 4

Question Type: MultipleChoice

What is a SAML2 artifact?

Options:

A- The SAML2 assertion
B- The SAML2 binding name
C- The name of a specific attribute in the assertion
D- A value sent by the service provider to retrieve the assertion

Answer:

D

Explanation:

In SAML 2.0, an Artifact is a reference (a 'pointer' or 'ticket') used in the SAML Artifact Binding.5 This is an alternative to the more common POST or Redirect bindings where the actual XML assertion is sent through the user's browser.

According to the PingAM 'SAML 2.0 Bindings' documentation:

When using the Artifact binding, the Identity Provider (IdP) does not send the full SAML Assertion through the browser.6 Instead, it sends a small, opaque string called the Artifact to the Service Provider (SP).

Issuance: The IdP stores the real assertion in its own local memory/cache and sends the Artifact to the SP via the browser redirect.

Resolution: The Service Provider receives the Artifact and then makes a direct, secure back-channel call (SOAP over HTTPS) to the IdP's Artifact Resolution Endpoint.

Exchange: The SP presents the Artifact, and the IdP returns the actual SAML Assertion.

Therefore, the Artifact is the value sent to retrieve the assertion (Option D). It is not the assertion itself (Option A), nor is it a binding name or an attribute name. The Artifact binding is often used for security reasons, as it prevents the sensitive assertion data from ever passing through the user's browser, thus mitigating certain types of interception attacks.

Question 5

Question Type: MultipleChoice

Which set of Directory Server stores can be enabled for affinity in a PingAM cluster configuration?

Options:

A- Identity Store, Configuration Store, Policy Data Store, Application Data Store
B- Core Token Service Store, Identity Store, Policy Data Store, Application Data Store
C- Core Token Service Store, Identity Stores, Configuration Store, Application Data Store
D- Core Token Service Store, Identity Stores, Configuration Store, Policy Data Store

Answer:

D

Explanation:

In a high-availability PingAM 8.0.2 cluster, Affinity Load Balancing is a mechanism used to ensure that requests related to a specific session or configuration are routed to the same Directory Server (DS) instance to avoid issues with replication lag. This is particularly important for stores where data changes frequently or where consistent reads are required immediately after a write.

According to the PingAM documentation on 'Load Balancing' and 'External Data Stores,' affinity can be configured for the following primary stores:

Core Token Service (CTS) Store: This is the most critical area for affinity. Since the CTS handles stateful data like session tokens and OAuth2 tokens that are updated constantly, ensuring that an AM server consistently communicates with a specific DS node (using the HOST:PORT|SERVERID|SITEID syntax) prevents 'token not found' errors that might occur if a request reached a DS node before the token was replicated.

Configuration Store: This store holds the central configuration for the AM deployment. In multi-server environments, affinity ensures that configuration changes are read consistently across the cluster.

Identity Stores: These hold the user profiles. While often read-heavy, affinity is used here to improve caching efficiency and ensure that profile updates (like password changes or attribute updates) are reflected immediately in subsequent authentication steps within the same cluster.

Policy Data Store: This stores authorization policies. Similar to configuration, affinity ensures consistent policy evaluation.

Option D is the correct answer because it includes the Core Token Service, Identity Stores, Configuration Store, and Policy Data Store. The 'Application Data Store' (mentioned in other options) is often logically grouped with or replaced by the Policy Data Store in many 8.0.2 configurations, but the four stores listed in Option D are the specific ones explicitly called out in the 'External Data Stores' secondary configuration documentation for supporting affinity settings.

P2P exams

Question 6

Question Type: MultipleChoice

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

Options:

A- authIndexType and authIndexValue

B- service, authIndexType, and authIndexValue
C- ForceAuth, authIndexType, and authIndexValue
D- service and ForceAuth

Answer:

A

Explanation:

In PingAM 8.0.2, when a client needs to trigger a specific authentication path---such as a higher-level tree for step-up authentication or a specific module for transactional authorization---it must tell the /authenticate endpoint which 'Index' to use.

According to the PingAM 'Authenticate over REST' and 'Session Upgrade' documentation, these are governed by two mandatory parameters:

authIndexType: This defines the category of the authentication mechanism being requested. Valid values include service (for Authentication Trees/Chains), module (for individual modules), or level (to request any mechanism that meets a specific Auth Level).

authIndexValue: This defines the name of the specific instance. For example, if authIndexType is service, the authIndexValue would be the name of the Authentication Tree (e.g., StepUpMFA).

For a step-up or transactional request to succeed, the client must send these two parameters. While service (Option B and D) is a common value for authIndexType, it is not a parameter name itself. ForceAuth (Option C and D) is an optional boolean used to force a fresh login even if a session exists, but it is not a requirement for the basic routing of the request to the correct tree. Therefore, authIndexType and authIndexValue (Option A) are the fundamental parameters required by the AM engine to identify and initiate the intended authentication journey.7

Question 7

Question Type: MultipleChoice

When the OATH Registration node's OATH Algorithm property is set to TOTP in an authentication tree, which node needs to have the same value set?

Options:

A- OATH Token Verifier node
B- Recovery Code Collector Decision node11
C- MFA Registration Options node12

D- OATH Device Storage node13

Answer:

A

Explanation:

In PingAM 8.0.2, Multi-Factor Authentication (MFA) using the OATH standard supports two primary algorithms: TOTP (Time-based One-Time Password) and HOTP (HMAC-based One-Time Password).14 For an authentication journey to function correctly, the 'Registration' phase (where the user's device and AM agree on a secret and algorithm) and the 'Verification' phase (where AM checks the submitted code) must be perfectly synchronized.

According to the 'Authentication Node Reference' for the OATH Token Verifier node and OATH Registration node:

Both nodes contain a configuration property named OATH Algorithm.15 This property determines how the six- or eight-digit code is generated and validated. If the OATH Registration node is configured to set up a user for TOTP, it will generate a QR code containing the TOTP parameters for the user's authenticator app.

When that user later attempts to log in, the OATH Token Verifier node (Option A) must also be set to TOTP.16 If the verifier is accidentally set to HOTP (which uses a counter rather than a time step), the validation will consistently fail because the server will be looking for a counter-based value while the app is providing a time-based value.

Other nodes like the Recovery Code Collector Decision node (Option B) or OATH Device Storage node (Option D) handle subsequent or separate tasks (like account recovery or writing the final profile to LDAP) and do not directly participate in the real-time OATH mathematical validation logic. Thus, the OATH Token Verifier is the mandatory counterpart that must match the registration's algorithm setting.

Question 8

Question Type: MultipleChoice

Which of the following tab pages in the PingAM admin UI can be used to configure the OAuth2 and OpenID Connect may act scripts used for token exchange requests?

A) The OAuth2 provider service > Advanced tab page

B) The OAuth2 provider service > Core tab page

C) The OAuth2 client profile > Advanced tab page

D) The OAuth2 client profile > OAuth2 Provider Overrides tab page

Options:

A- B and D only
B- A and D only
C- A and C only
D- B and C only

Answer:

B

Explanation:

The May Act script is a critical component of the OAuth 2.0 Token Exchange implementation in PingAM 8.0.2. It allows for the validation of impersonation or delegation requests. Because token exchange can be configured both globally for all clients and specifically for individual applications, the script can be attached at two different levels in the Administrative UI.

OAuth2 Provider Service > Advanced Tab (A): This is the global configuration level. If you want to apply a standard 'May Act' validation script across the entire realm for any client performing a token exchange, you configure it here. This script will be the default unless specifically overridden.

OAuth2 Client Profile > OAuth2 Provider Overrides Tab (D): PingAM allows for granular control per client. If a specific 'Confidential Client' (like a backend microservice) requires unique logic for determining who it can act as, you can specify a different script or override the global setting. This is done in the 'OAuth2 Provider Overrides' tab within that specific client's configuration profile.

Why other options are incorrect: The Core tab (B) is used for basic settings like issuer names and token lifetimes, not for advanced scripting hooks. The Advanced tab of the Client Profile (C) contains settings like TTLs and Logout URLs, but the specific ability to override 'Provider' level logic (like the May Act script) is moved to the specialized Overrides tab to keep the interface organized. Therefore, the correct locations are A and D, as identified in the 'Token Exchange Configuration' guide for version 8.0.2.

Question 9

Question Type: MultipleChoice

Sam wants to start a service provider-initiated single sign-on and redirect to their own

application, myapp.com. Which option best URLs is the correct one to perform this action?

Options:

A- http://sso.domain.com/openam/saml2/jsp/idpSSOInit.jsp&RelayState=http%3A%2F%2Fmyapp.com
B- http://sso.domain.com/openam/saml2/jsp/idpSSOInit.jsp&goto=http%3A%2F%2Fmyapp.com
C- http://sso.domain.com/openam/saml2/jsp/spSSOInit.jsp&goto=http%3A%2F%2Fmyapp.com
D- http://sso.domain.com/openam/saml2/jsp/spSSOInit.jsp&RelayState=http%3A%2F%2Fmyapp.com

Answer:

D

Explanation:

In SAML 2.0 federation with PingAM 8.0.2, there are two ways to initiate SSO: IdP-Initiated (where the user starts at the Identity Provider) and SP-Initiated (where the user starts at the Service Provider).3

According to the 'SAML 2.0 Guide' for PingAM:

SP-Initiated SSO: The correct JSP file for an SP-initiated flow is spSSOInit.jsp. 4This script is used by an SP (in this case, PingAM acting as an SP or a 'Fedlet') to generate a SAML AuthnRequest and send it to the IdP.

Redirecting to the Application: In the SAML 2.0 standard, the mechanism used to preserve state (like the final destination URL) across the redirect-heavy SSO process is the RelayState parameter. When the IdP sends the SAML assertion back to the SP, it also returns the RelayState value. The SP then uses this value to redirect the user to the final application.

While PingAM uses the goto parameter for internal redirects (like standard web login), RelayState is the required parameter name for SAML-related JSPs to ensure interoperability with the SAML specification. Therefore, the correct URL is .../spSSOInit.jsp combined with the RelayState parameter (Option D). Using idpSSOInit.jsp (Options A and B) would trigger an IdP-initiated flow, which is not what the question describes. Option C is incorrect because it uses the non-SAML goto parameter in a SAML initialization context.

Question 10

Question Type: MultipleChoice

A SAML2 identity provider (IdP) is configured in a subrealm. Which option best URLs can be used to export the IdP metadata?

Options:

A- It cannot be exported via a JSP, and the Amster tool has to be used
B- http://myserver.domain.com:8080/openam/saml2/jsp/exportmetadata.jsp
C- http://myserver.domain.com:8080/openam/saml2/jsp/exportmetadata.jsp?idp=http://myserver.domain.com:8080/openam&realm=/idprealm
D- http://myserver.domain.com:8080/openam/saml2/jsp/exportmetadata.jsp?entityid=http://myserver.domain.com:8080/openam&realm=/idprealm

Answer:

D

Explanation:

To facilitate federation between a SAML2 Identity Provider (IdP) and a Service Provider (SP), metadata must be exchanged. PingAM 8.0.2 provides a built-in utility page, exportmetadata.jsp, specifically for this purpose.

When an IdP is configured within a subrealm (rather than the Top Level Realm), the metadata export URL must be qualified with specific query parameters to ensure the correct entity configuration is retrieved. According to the 'SAML 2.0 Reference' and 'Exporting SAML 2.0 Metadata' documentation:

entityid: This parameter is mandatory when there are multiple entities configured. It specifies the unique URI of the IdP (e.g., http://myserver.domain.com:8080/openam). This tells the JSP which specific provider's metadata to generate.

realm: This parameter is crucial for subrealm deployments. By default, the JSP looks in the root realm (/). If the IdP resides in a subrealm named /idprealm, the URL must explicitly include &realm=/idprealm.

Option D is the correct technical string. Option B is incorrect as it lacks parameters and would only attempt to export default root-level metadata. Option C is incorrect because the parameter name is entityid, not idp. While Amster (Option A) can indeed be used to export configuration, the exportmetadata.jsp remains the standard and most common method for generating the XML-formatted metadata required by external partners.

To Get Premium Files for PT-AM-CPE Visit

https://www.p2pexams.com/products/pt-am-cpe

For More Free Questions Visit

https://www.p2pexams.com/ping-identity/pdf/pt-am-cpe