

DUMPSBOSS.

**ServiceNow Certified Implementation Specialist
- Service Provide**

ServiceNow CIS-SP

Version Demo

Total Demo Questions: 5

Total Premium Questions: 56

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

**support@dumpsboss.co
dumpsboss.co**

Topic Break Down

Topic	No. of Questions
Topic 1, Service Provider Architecture	52
Topic 2, Customer Management	4
Total	56

QUESTION NO: 1

An administrator in a child domain modifies an inherited application module to meet a customer-specific navigation requirement. Which field does the system set to identify the child-domain record as an override?

- A. sys_domain_path
- B. sys_overrides
- C. sys_primary_domain
- D. sys_domain_owner

ANSWER: B

Explanation:

The system sets `sys_overrides`. This identifies the child-domain record as an override of the inherited definition, allowing the child domain to use its local configuration without changing the parent-domain module.

The override does not alter the parent definition or make the child configuration available to unrelated domains.

QUESTION NO: 2

A System Administrator wants to setup their domain hierarchy in a new instance, which practice should they follow when creating the structure?

- A. A domain hierarchy 3-5 layers deep that allows for use of contains if needed and contains a default domain
- B. Using Service Offerings in the domain hierarchy
- C. A domain hierarchy 3-5 layers deep that allows for use of contains if needed and does not contain a default domain
- D. Having a totally flat domain hierarchy with no TOP domain
- E. Adding several domain layers below TOP before getting to the customer domain

ANSWER: A

Explanation:

A domain hierarchy 3-5 layers deep that allows for use of contains if needed and contains a default domain is the recommended approach. A domain hierarchy should reflect meaningful business, customer, or organizational boundaries while remaining manageable for administrators. Keeping the design to roughly three to five levels helps avoid an overly complex domain path while providing sufficient separation for provider, customer, and subsidiary or departmental structures when required. This depth also supports appropriate use of domain relationships, including the *contains* relationship, to make data and processes available across the intended portions of the hierarchy.

Including a default domain is important because it provides a defined domain context for records that are not intended for a more specific customer or child domain. Establishing this structure before configuring domain-separated applications, users, groups, and data prevents avoidable redesign work later. ServiceNow domain separation uses the hierarchy to control data visibility and process behavior, so a deliberate, scalable hierarchy is a foundation for a successful service-provider implementation. Review ServiceNow's [Domain separation documentation](#) and its [domain hierarchy guidance](#) for implementation details.

QUESTION NO: 3

An administrator uses the Domain Separation Center after a hierarchy redesign. Which capabilities are provided by a domain configuration audit?

Choose 3 answers

- A. Identify records or configurations affected by a finding
- B. Provide recommended actions for errors and warnings
- C. Re-run the audit after remediation
- D. Automatically create incidents for every finding
- E. Correct incorrect user login credentials

ANSWER: A B C

Explanation:

A domain configuration audit identifies affected records or configurations and provides recommended actions for reported errors and warnings. After remediation, the administrator can re-run the audit to validate the current configuration. The audit does not automatically create incidents or correct user login errors.

QUESTION NO: 4

What type of detailed results or actions are included in the domain audit?

Choose 3 answers

- A. incorrect user domain logins
- B. escalate the errors as incidents
- C. records or configurations affected
- D. recommended actions to remedy errors and warnings
- E. ability to re-run the audit

ANSWER: C D E

Explanation:

Domain Audit is a Domain Separation diagnostic capability that evaluates domain-related configuration and reports findings in a form administrators can investigate and remediate. The audit details the **records or configurations affected**, allowing the administrator to identify the specific domain-separation setup, data, or configuration items associated with a finding. This detail is important because remediation needs to be applied to the actual affected records while preserving the intended domain hierarchy and data visibility model.

The audit also provides **recommended actions to remedy errors and warnings**. Findings are actionable rather than merely informational: the results guide an administrator toward correcting detected configuration problems or addressing warning conditions in the domain setup. After corrective work has been performed, the **ability to re-run the audit** supports validation. Re-running the assessment produces current results and confirms whether the remediation resolved the reported conditions or whether additional work remains.

These capabilities make Domain Audit useful as a repeatable configuration-health review process for a domain-separated instance. See ServiceNow's [Domain Separation administration documentation](#) and the [ServiceNow product documentation portal](#) for Domain Separation administration guidance.

QUESTION NO: 5

What is the best practice regarding User Criteria and Shared Knowledge Bases?

- A. Knowledge bases and User criteria should be defined in Global
- B. Knowledge bases and User criteria should be defined in the company domain

- C. Knowledge bases and User criteria should be defined in the parent domain so that they can be visible in a child domain
- D. Knowledge bases and User criteria should be defined in the service provider domain

ANSWER: C

Explanation:

Knowledge bases and User criteria should be defined in the parent domain so that they can be visible in a child domain. In a domain-separated implementation, placing shared Knowledge Management configuration in a parent domain makes it available to that domain's descendants through the domain hierarchy. This supports the intended shared-knowledge model: a provider or parent organization can centrally manage a knowledge base and the access rules that govern it, while child domains can use the same approved content without requiring duplicate knowledge-base or user-criteria records in every child domain.

Defining both the knowledge base and its associated user criteria at the same appropriate parent level also keeps access design consistent. The parent domain becomes the administrative scope for shared content, while the hierarchy determines which child domains can inherit visibility. This approach is especially important for service-provider environments, where some knowledge must be reused across multiple customers or business units but still must follow domain boundaries and the criteria governing who can read or contribute content.

ServiceNow describes domain separation as a hierarchy used to separate data and administration while enabling appropriate sharing across domains. See the [ServiceNow Domain Separation documentation](#) and the [ServiceNow Knowledge Management documentation](#).