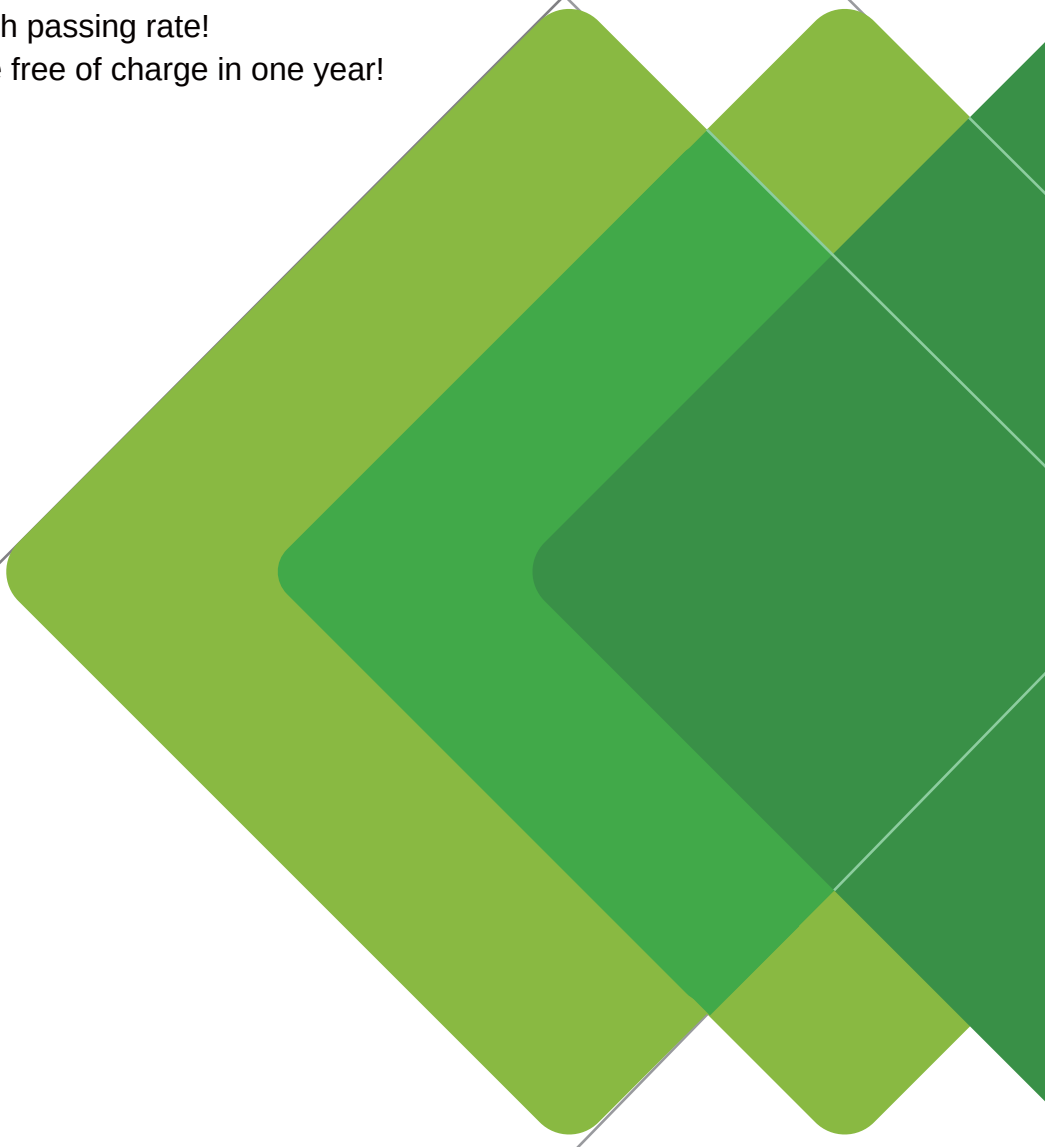
IT-DUMPS Q&A

Accurate study guides, High passing rate!
IT-dumps provides update free of charge in one year!



Exam : **JN0-637**

Title : Security, Professional (JNCIP-SEC)

Version : DEMO

1.You are enabling advanced policy-based routing. You have configured a static route that has a next hop from the inet.0 routing table. Unfortunately, this static route is not active in your routing instance.
In this scenario, which solution is needed to use this next hop?
A. Use RIB groups.
B. Use filter-based forwarding.
C. Use transparent mode.
D. Use policies.
Answer: A
Explanation:
To enable advanced policy-based routing in Junos OS and activate a static route with a next-hop address in the inet.0 table within your routing instance, you should utilize RIB groups. RIB groups allow you to import routes from one routing table to another. In this scenario, the static route within the routing instance needs access to the inet.0 routes, which is facilitated by configuring a RIB group. Juniper's documentation outlines RIB groups as a necessary component for handling instances where routes need to be shared across routing tables, thereby ensuring seamless traffic flow through specified routes. For more details, refer to the Juniper Networks Documentation on RIB Groups.
In Junos OS for SRX Series devices, when enabling advanced policy-based routing and configuring a static route with a next-hop from the inet.0 routing table, the issue arises because the static route is not being used in the routing instance. This is a common scenario when the next-hop belongs to a different routing table or instance, and the routing instance is not aware of that next-hop.
To resolve this, RIB (Routing Information Base) groups are used. RIB groups allow routes from one routing table (RIB) to be shared or imported into another routing table. This means that the routing instance can import the necessary routes from inet.0 and make them available for the routing instance where the policy-based routing is applied.
Detailed Steps:
Configure the Static Route: First, configure the static route pointing to the next-hop in inet.0.
Here's an example:
bash set routing-options static route 10.1.1.0/24 next-hop 192.168.1.1 This static route will be placed in the inet.0 routing table by default.
Create and Apply a RIB Group: To import routes from inet.0 into the routing instance, create a RIB group configuration. This will allow the static route from inet.0 to be visible within the routing instance.
Example configuration for the RIB group:
bash
set routing-options rib-groups RIB-GROUP import-rib inet.0
set routing-options rib-groups RIB-GROUP import-rib <routing-instance-name>.inet.0
This configuration ensures that routes from inet.0 are imported into the specified routing instance.
Apply the RIB Group to the Routing Instance:
Once the RIB group is configured, apply it to the appropriate routing instance:
bash
set routing-instances <routing-instance-name> routing-options rib-group RIB-GROUP
Verify Configuration: Use the following command to verify that the static route has been imported into the routing instance:
bash
show route table <routing-instance-name>.inet.0

The output should now display the static route imported from inet.0.
Juniper Security
Reference: RIB Groups Overview: Juniper's documentation provides detailed information on how RIB groups function and how to use them to share routes between different routing tables. This is essential for scenarios involving policy-based routing where routes from one instance (like inet.0) need to be available in another instance.
Reference: Juniper Networks Documentation on RIB Groups.
By using RIB groups, you ensure that the static route from inet.0 is available in the appropriate routing instance for policy-based routing to function correctly. This avoids the need for other methods like filter-based forwarding or transparent mode, which do not address the specific issue of static route visibility across routing instances.

2.Exhibit:

```
Aug  3 02:10:28 02:10:28.045090:CID-0:THREAD_ID-01:RT:  <10.10.101.10/60858->10.10.102.10/22;6,0x0> matched filter filter-1:
...
Aug  3 02:10:28 02:10:28.045100:CID-0:THREAD_ID-01:RT:  no session found, start first path. in_tunnel - 0x0, from_cp_flag -
0
Aug  3 02:10:28 02:10:28.045104:CID-0:THREAD_ID-01:RT:  flow_first_create_session
...
Aug  3 02:10:28 02:10:28.045143:CID-0:THREAD_ID-01:RT:  routed (x_dst_ip 10.10.102.10) from trust (ge-0/0/4.0 in 0) to ge-
0/0/5.0, Next-hop: 10.10.102.10
Aug  3 02:10:28 02:10:28.045158:CID-0:THREAD_ID-01:RT:  flow_first_policy_search: policy search from zone trust-> zone dmz
(0x0,0xedba0016,0x16)
...
Aug  3 02:10:28 02:10:28.045191:CID-0:THREAD_ID-01:RT:  packet dropped, denied by policy
Aug  3 02:10:28 02:10:28.045192:CID-0:THREAD_ID-01:RT:  denied by policy default-policy-logical-system-00(2), dropping pkt
Aug  3 02:10:28 02:10:28.045192:CID-0:THREAD_ID-01:RT:  packet dropped,  policy deny.
Aug  3 02:10:28 02:10:28.045195:CID-0:THREAD_ID-01:RT:  flow_initiate_first_path: first pak no session
```

Referring to the flow logs exhibit, which two statements are correct? (Choose two.)
A. The packet is dropped by the default security policy.
B. The packet is dropped by a configured security policy.
C. The data shown requires a traceoptions flag of host-traffic.
D. The data shown requires a traceoptions flag of basic-datapath.
Answer: AD
Explanation:
Understanding the Flow Log Output:
From the flow logs in the exhibit, we can observe the following key events:
The session creation was initiated (flow_first_create_session), but the policy search failed (flow_first_policy_search), which implies that no matching policy was found between the zones involved (zone trust-> zone dmz).
The packet was dropped with the reason "denied by policy." This shows that the packet was dropped either due to no matching security policy or because the default policy denies the traffic (packet dropped, denied by policy).
The line denied by policy default-policy-logical-system-00(2) indicates that the default security policy is responsible for denying the traffic, confirming that no explicit security policy was configured to allow this traffic.
Explanation of Answer A (Dropped by the default security policy):
The log message clearly states that the packet was dropped by the default security policy (default-policy-logical-system-00). In Junos, when a session is attempted between two zones and no explicit policy exists to allow the traffic, the default policy is to deny the traffic. This is a common behavior in Junos OS

when a security policy does not explicitly allow traffic between zones. Explanation of Answer D (Requires traceoptions flag of basic-datapath):
The information displayed in the log involves session creation, flow policy search, and packet dropping due to policy violations, which are all part of basic packet processing in the data path. This type of information is logged when the traceoptions flag is set to basic-datapath. The basic-datapath traceoption provides detailed information about the forwarding process, including policy lookups and packet drops, which is precisely what we see in the exhibit.
The traceoptions flag host-traffic (Answer C) is incorrect because host-traffic is typically used for traffic destined to or generated from the Junos device itself (e.g., SSH or SNMP traffic to the SRX device), not for traffic passing through the device.
To capture flow processing details like those shown, you need the basic-datapath traceoptions flag, which provides details about packet forwarding and policy evaluation.
Step-by-Step Configuration for Tracing (Basic-Datapath):
Enable flow traceoptions:
To capture detailed information about how traffic is being processed, including policy lookups and flow session creation, enable traceoptions for the flow. bash
set security flow traceoptions file flow-log
set security flow traceoptions flag basic-datapath
Apply the configuration and commit:
bash
commit
View the logs:
Once enabled, you can check the trace logs for packet flows, policy lookups, and session creation details:
bash
show log flow-log
This log will contain information similar to the exhibit, including session creation attempts and packet drops due to security policy.
Juniper Security
Reference: Default Security Policies: Juniper SRX devices have a default security policy to deny all traffic that is not explicitly allowed by user-defined policies. This is essential for security best practices.
Reference: Juniper Networks Documentation on Security Policies.
Traceoptions for Debugging Flows: Using traceoptions is crucial for debugging and understanding how traffic is handled by the SRX, particularly when issues arise from policy misconfigurations or routing.
Reference: Juniper Traceoptions.
By using the basic-datapath traceoptions, you can gain insights into how the device processes traffic, including policy lookups, route lookups, and packet drops, as demonstrated in the exhibit.

3.Exhibit:

```
[edit]
user@srx# show security nat
source {
    pool ipv4-source-pool {
        address {
            10.10.101.10/32;
        }
    }
    rule-set ipv6-source {
        from zone trust;
        to zone untrust;
        rule ipv6-host-source {
            match {
                source-address 2001:db8::1/128;
                destination-address 10.10.201.10/32;
            }
            then {
                source-nat {
                    pool {
                        ipv4-source-pool;
                    }
                }
            }
        }
    }
}
```

You are configuring NAT64 on your SRX Series device. You have committed the configuration shown in the exhibit. Unfortunately, the communication with the 10.10.201.10 server is not working. You have verified that the interfaces, security zones, and security policies are all correctly configured.
In this scenario, which action will solve this issue?
A. Configure source NAT to translate return traffic from IPv4 address to the IPv6 address of your source device.
B. Configure proxy-ARP on the external IPv4 interface for the 10.10.201.10/32 address.
C. Configure proxy-NDP on the IPv6 interface for the 2001:db8::1/128 address.
D. Configure destination NAT to translate return traffic from the IPv4 address to the IPv6 address of your source device.
Answer: D

4.What are three core components for enabling advanced policy-based routing? (Choose three.)
A. Filter-based forwarding
B. Routing options
C. Routing instance
D. APBR profile

E. Policies

Answer: ACD

Explanation:

To enable Advanced Policy-Based Routing (APBR) on SRX Series devices, three key components are necessary: filter-based forwarding, routing instances, and APBR profiles. Filter-based forwarding is utilized to direct specific traffic flows to a routing instance based on criteria set by a policy. Routing instances allow the traffic to be managed independently of the main routing table, and APBR profiles define how and when traffic should be forwarded. These elements ensure that APBR is flexible and tailored to the network’s requirements. Refer to Juniper's APBR Documentation for more details.

Advanced policy-based routing (APBR) in Juniper's SRX devices allows the selection of different paths for traffic based on policies, rather than relying purely on routing tables.

To enable APBR, the following core components are required:

Filter-based Forwarding (Answer A): Filter-based forwarding (FBF) is a technique used to forward traffic based on policies rather than the default routing table. It is essential for enabling APBR, as it helps match traffic based on filters and directs it to specific routes.

Configuration Example:

bash

set firewall family inet filter FBF match-term source-address 192.168.1.0/24 set firewall family inet filter FBF then routing-instance custom-routing-instance

Routing Instance (Answer C): A routing instance is required to define the separate routing table used by APBR. You can create multiple routing instances and assign traffic to these instances based on policies. The traffic will then use the routes defined within the specific routing instance.

Configuration Example:

bash

set routing-instances custom-routing-instance instance-type forwarding

set routing-instances custom-routing-instance routing-options static route 0.0.0.0/0 next-hop 10.10.10.1

APBR Profile (Answer D): The APBR profile defines the rules and policies for advanced policy-based routing. It allows you to set up conditions such as traffic type, source/destination address, and port, and then assign actions such as redirecting traffic to specific routing instances.

Configuration Example:

bash

set security forwarding-options advanced-policy-based-routing profile apbr-profile match application http set security forwarding-options advanced-policy-based-routing profile apbr-profile then routing-instance custom-routing-instance

Other Components:

Routing Options (Answer B) are not a core component of APBR, as routing options define the general behavior of the routing table and protocols. However, APBR works by overriding these default routing behaviors using policies.

Policies (Answer E) are crucial in many network configurations but are not a core component of enabling APBR. APBR specifically relies on profiles rather than standard security policies. Juniper Security

Reference: Advanced Policy-Based Routing (APBR): Juniper’s APBR is a powerful tool that allows routing based on specific traffic characteristics rather than relying on static routing tables. APBR ensures that specific types of traffic can take alternate paths based on business or network needs.

Reference: Juniper Networks APBR Documentation.

5.You want to bypass IDP for traffic destined to social media sites using APBR, but it is not working and IDP is dropping the session.
What are two reasons for this problem? (Choose two.)
A. The session did not properly reclassify midstream to the correct APBR rule.
B. IDP disable is not configured on the APBR rule.
C. The application services bypass is not configured on the APBR rule.
D. The APBR rule does a match on the first packet.
Answer: AC
Explanation:
Explanation of Answer A (Session Reclassification):
APBR (Advanced Policy-Based Routing) requires the session to be classified based on the specified rule, which can change midstream as additional packets are processed. If the session was already established before the APBR rule took effect, the traffic may not be correctly reclassified to match the new APBR rule, leading to IDP (Intrusion Detection and Prevention) processing instead of being bypassed. This can occur especially when the session was already established before the rule change.
Explanation of Answer C (Application Services Bypass):
For APBR to work and bypass the IDP service, the application services bypass must be explicitly configured. Without this configuration, the APBR rule may redirect the traffic, but the IDP service will still inspect and potentially drop the traffic. This is especially important for traffic destined for specific sites like social media platforms where bypassing IDP is desired. Example configuration for bypassing IDP services:
bash
set security forwarding-options advanced-policy-based-routing profile <profile-name> application-services-bypass
Step-by-Step Resolution:
Reclassify the Session Midstream:
If the traffic was already being processed before the APBR rule was applied, ensure that the session is reclassified by terminating the current session or ensuring the APBR rule is applied from the start.
Command to clear the session:
bash
clear security flow session destination-prefix <ip-address>
Configure Application Services Bypass:
Ensure that the APBR rule includes the application services bypass configuration to properly bypass IDP or any other security services for traffic that should not be inspected.
Example configuration:
bash
set security forwarding-options advanced-policy-based-routing profile <profile-name> application-services-bypass
Juniper Security
Reference: Session Reclassification in APBR: APBR requires reclassification of sessions in real-time to ensure midstream packets are processed by the correct rule. This is crucial when policies change dynamically or new rules are added.
Application Services Bypass in APBR: This feature ensures that security services such as IDP are

bypassed for traffic that matches specific APBR rules. This is essential for applications where performance is a priority and security inspection is not necessary.