

BSP Test Cases

Loopback Test Cases

Test Directory Names: 100.bsp-bab-phn-udp, 100.bsp-pcb-phn-udp, 100.bsp-pib-phn-udp, 100.bsp-all-phn-udp

Description: The four directories/test categories listed above conduct the following test cases for their corresponding security block. In the test category 100.bsp-all-phn-udp, we use all security blocks at once. In all the test cases, we are sending a bundle from one service point to another on the same node. We use the ltp transport protocol.

Test 1 tries to send a bundle when there is no applicable rule present. However, the node does contain a security policy for that block..so the bundle will not be delivered. The exception is when we send with a PCB block alone, as ion will still deliver the unencrypted block to the application.

Test 2 attempts to send a bundle with a security block(s). This should be delivered.

Test 3 will send an anonymous bundle, which should deliver successfully.

Visual:

1 - - - 2

Configuration:

For all tests:

a contact +1 +3600 1 1 100000

a range +1 +3600 1 1 1

a endpoint ipn:1.1 q

a endpoint ipn:1.2 q

Multi-node Test Cases

Test Directory Names: bsp-bab-multinode-test, bsp-pib-multinode-test, bsp-pcb-multinode-test, bsp-all-multinode-test

Description: The four directories/test categories listed above conduct the following test cases for their corresponding security block. In the test category bsp-all-multinode-test, we use all security blocks at once. Tests 1-5 involve sending a bundle from node 2 to node 3. Tests 6-7 involve sending a bundle from node 2 to node 4, going through node 3 in the process. We use the ltp transport protocol.

Test 1 sends bundle when the nodes don't have ANY security rules. In this case, no security blocks are added and the bundle will be asserted as authentic for delivery since no security policy exists. It should be delivered in all cases.

Test 2 attempts to send a bundle with a security block(s). This should be delivered.

Test 3 attempts to send a bundle with a security block(s). However, node 2 has a different key in

its security rule than node 3. Therefore, the bundle will be judged as inauthentic and fail to deliver.

Test 4 clears the rules added in test 2 and 3, and repeats test 1. This explicitly tests ionsecadmin's "clear rules" function. It's also thorough in ensuring test 2 and 3 doesn't quietly mess up anything.

Test 5 attempts to send a bundle via a file with security blocks. It should deliver.

Test 6 attempts to send a bundle from node 2 to 3 to 4. For BAB block, the bundle should be authenticated at each hop as all nodes will have the appropriate rules and keys. For PCB and PIB block, node 3 doesn't not have the appropriate rule and keys- yet it should recognize that the security destination is node 4 and simply forward the bundle with their blocks. The bundle should be delivered to node 4.

Test 7 is exactly like test 6, except with an anonymous bundle. The bundle should still come from node 2's outduct, through node 3, and finally to node 4. It should be delivered.

Visual:

2 - - - - 3 - - - - 4

Configuration:

For all tests:

```
a contact +0 +3600 2 2 10000
a contact +0 +3600 2 3 10000
a contact +0 +3600 3 2 10000
a contact +0 +3600 3 4 10000
a contact +0 +3600 4 3 10000
a range 2008/01/01-00:00:00 2009/01/01-00:00:00 2 3 0
a range 2008/01/01-00:00:00 2009/01/01-00:00:00 3 4 0
```