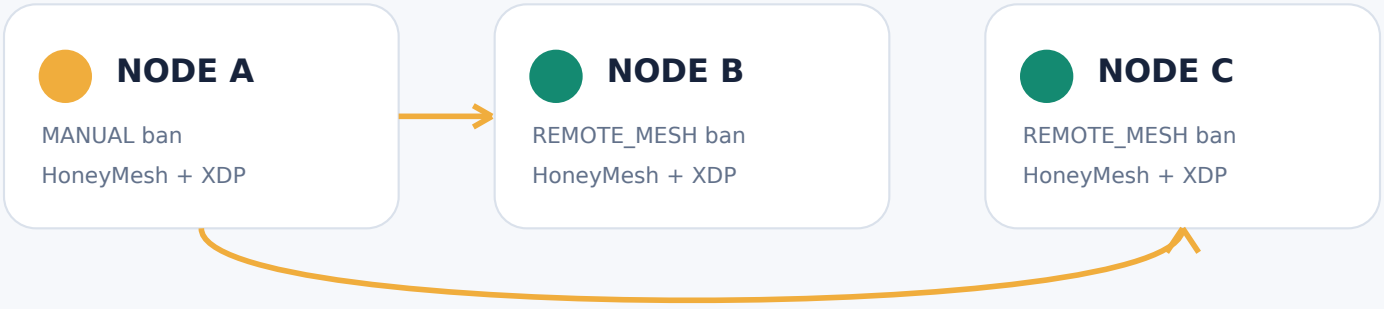
Federation ban: three-node proof

A release-binary test of signed ban sharing and packet enforcement across three AMD64 HoneyMesh nodes.

PASS Node A issued one ban. Nodes B and C accepted it as REMOTE_MESH; all three blocked the banned source while an unbanned source remained reachable.

3	5	15 / 15
AMD64 nodes in the mesh	protocol / port cases	blocked-case checks passed

Ban propagation



A BROADCASTS TO BOTH PEERS OVER MUTUAL TLS / TCP 9000

TRAFFIC CHECK

Before: banned-source candidate 10.77.0.4 reached all five endpoints on every node. After: it received no reply on any endpoint. The separate control source 10.77.0.5 still reached all five.

The installed binary matched the published rc2 AMD64 tarball. The updated license generator's signed Pro license worked without a test-key overlay.

A ban that reaches the packet path

Each row was checked on nodes A, B, and C. All commands returned SSM status Success in the final run.

TRAFFIC	BEFORE BAN / .4	BANNED / .4	CONTROL / .5
TCP port 80	3 / 3 replied	3 / 3 no reply	3 / 3 replied
TCP port 8448	3 / 3 replied	3 / 3 no reply	3 / 3 replied
UDP port 53	3 / 3 replied	3 / 3 no reply	3 / 3 replied
UDP port 9999	3 / 3 replied	3 / 3 no reply	3 / 3 replied
ICMP echo request	3 / 3 replied	3 / 3 no reply	3 / 3 replied

Live node state after federation

NODE A
MANUAL
Pro license • 3 mesh entries

NODE B
REMOTE_MESH
Pro license • 3 mesh entries

NODE C
REMOTE_MESH
Pro license • 3 mesh entries

Method: each node received probes through a private virtual Ethernet pair and an isolated network namespace. The blocked result is a 1.5-second socket timeout. The control source shows that the listeners were still running.

Proof you can inspect

The linked JSON files are unedited AWS Systems Manager command-invocation records. They include instance ID, command ID, status, timestamps, stdout, and stderr.

SSM RECORDS

Before ban	A baseline-a.json	B baseline-b.json	C baseline-c.json
After ban	A post-a.json	B post-b.json	C post-c.json
Control	A control-a.json	B control-b.json	C control-c.json

Chain of evidence

- Three x86_64 EC2 instances installed the same rc2 AMD64 package and reported the same binary SHA-256.
- Each instance reported Pro enabled, a trusted mesh, an attached XDP program, and a restricted service user.
- A's ban API returned HTTP 200; B and C recorded REMOTE_MESH and received mesh intelligence.
- Local race tests, signed-license compatibility and tamper rejection, and Go vet all exited successfully.

PACKAGE HASH

`bd2937beabd212bfd59422bffcf985337da681f632d51dece1f68bf6efb51e72`

INSTALLED BINARY HASH

`04e28867f2be785c3a64ead1b6d6c072a25855fc7c0bfee21b1884855844a3ca`

LIMITS & CLEANUP

This run covers one IPv4 ban, five traffic cases, and three AMD64 nodes. It does not establish IPv6, all ports, long-term reconnects, concurrent conflicting bans, or Arm64 behavior. The preliminary probe script had a timeout-labeling error; it was corrected before the passing run. The test-mode ban lasts one minute, so a fresh ban was used for the final run. All three instances were terminated; the test stack and bucket were deleted.