Implementing SHIM6 Protocol

Kunwoo Park
Hosik Cho
Indong Jang
Taewan You
Seungyun Lee

IETF 67 San Diego
Contents

• Implementation overview
• Schedule
• System architecture
• Implementation features
• Testbed & Demonstration
• Further API works
• Collaboration plan
Implementation Overview

• Administrative Data
  – Period: April. 2006 – November. 2006
  – Participants: ETRI & SNU, Korea

• Contact persons
  – Taewan You, ETRI (twyou@etri.re.kr)
  – Hosik Cho, SNU (hscho@mmlab.snu.ac.kr)
  – Kunwoo Park, SNU (kwpark@mmlab.snu.ac.kr)

• Additional info
  – Webpage will be available soon
Schedule

• Phase 1: May, 2006 ~ November, 2006
  – SHIM6 core daemon
  – REAP
  – Simple testbed

• Phase 2: January, 2007 ~ TBD
  – Add security aspects
  – SHIM6 Stack via direct kernel patch
  – Library for SHIM6
Environments

• Reference
  – Level 3 multihoming shim protocol
    • draft-ietf-shim6-proto-05.txt
  – Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming
    • draft-ietf-shim6-failure-detection-05

• Platform
  – Target OS: Linux
  – Requirement: Linux 2.6.x kernel or higher, Netfilter with iptables 1.3.5
System Architecture

Application layer

Transport layer

IP layer
  Netfilter/ip6tables

Link layer

libipq

SHIM DAEMON

IETF 67 San Diego
Implementation Features

• **SHIM6 CORE**
  – Observing incoming and outgoing packets in shim6d module
  – Initial handshake (4-way) and locator set exchange
  – ULID / Locator mapping and substitution
  – CGA/HBA parameters
  – Context forking

• **SHIM6 REAP**
  – Reachability detection – Keepalive
  – Path exploration – Probe
  – Change locators to use alternate path
  – New message format 05 > 06
  – Security issues
Testbed environment
Introduction to further works
API works for SHIM6

- **Socket API for SHIM6**
  - Draft-ietf-shim6-multihoming-shim-api-01.txt

- **Make a Policy functions**
  - Directly control SHIM core and REAP
    - Occur address pair changing trigger without failure

- **Receive Link Information**
  - Changing address pair without REAP

- **Inter-process Communication**
  - Optimize operation
Collaboration Works

• Members
  – ETRI&SNU (Taewan You, twyou@etri.re.kr)
  – Ericsson (Shinta Sugimoto, shinta.sugimoto@ericsson.com)
  – OpenHIP (Thomas R Henderson, thomas.r.henderson@boeing.com)

• Inter-operability
  – Basic functionality test
  – Advanced functionality test
  – Stress test

• API works