Engineering of Host Identity Protocol on Linux (HIPL) Software

Miika Komu <miika@iki.fi>
Helsinki Institute for Information Technology
16.11.2009
Table of Contents

- Motivation
- Naming and Layering
- Control Plane
- Data Plane
- Software Engineering
Motivation

- Identity-locator split separates the “who” from “where”
  - This indirection is beneficial to support mobility
  - Realized in HIP, SHIM6, LISP

- HIP fixes several short-comings in the Internet within a single architecture
  - Protects and authenticates mobile application data
  - Mobility and multihoming for transport layer
  - End-to-end NAT traversal
  - Backwards compatible (TCP, UDP, IPv4 and IPv6)
Naming and Layering
Layering

- User Interface
  - FQDN
- Application Layer
  - HIT (or LSI), port, and proto
- Transport Layer
  - HIT, port
- HIP Layer
  - HI
- Network Layer
  - IP address
Socket Bindings

Process <-> Socket

Endpoint <-> Location

Socket IP address

Process <-> Socket

Endpoint <-> Host Identity

Dynamic Binding

Location <-> IP address
## APIs

<table>
<thead>
<tr>
<th>Layer</th>
<th>Application</th>
<th>IPv4 API</th>
<th>IPv6 API</th>
<th>HIP API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socket Layer</td>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport Layer</td>
<td></td>
<td>UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIP Layer</td>
<td>HIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Layer</td>
<td>IPv4</td>
<td></td>
<td>IPv6</td>
<td></td>
</tr>
<tr>
<td>Link Layer</td>
<td>Ethernet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Client-Side Name Look Up Example

1. getaddrinfo(hostname)
2. hostname
3. <HIT, IP>
4. <HIT, IP>
5. 
6. HIT
7. connect(HIT)

Resolver or DNS Proxy

DNS

Application

Socket Layer
Transport
HIP
IPsec
Network

Peer Host

base exchange

9. ESP protected application data
HIP Name Translation

- #1 LD_PRELOAD getaddrinfo()
- #2 Local DNS Proxy
  - #2a Snoop DNS requests with iptables
  - #2b Replace the nameserver to /etc/resolv.conf
- Note: LSIs for applications requesting A records
- #3 No changes to DNS interaction
  - Implement lower in the stack (opp. mode)
  - Implemented in router (HIP proxy)
Control Plane
The Base Exchange

**I1**: trigger base exchange

**R1**: puzzle, D-H, key, signature

**I2**: puzzle solution, D-H, key, signature

**R2**: signature
Handover (UPDATE)

1) UPD: ESP_INFO, LOCATOR, SEQ [, D-H]

2) UPD: ESP_INFO, ACK, SEQ, E_RQ [, D-H]

3) [create SA]

4) UPD: ESP_INFO, E_RS

5) [create SA]
Packet Flooding Hacks

- “Shotgun” for base exchange and UPDATE
  - Send the I1 or first UPDATE packet using all source/destination address combinations
  - Make sure that we find a route that works
  - Can find the route with smallest latency
- I1 broadcasting to 255.255.255.255
  - Last resort when peer IP address is unknown
- Opportunistic I1 to 255.255.255.255 (BOS)
  - Scans all HIP capable hosts in the network
NAT Traversal using HIP and ICE

1. base exchange with locators
2. pair up locators
3. connectivity tests
4. ESP

Initiator

Responder

HIP Relay Server

ESP Relay Server
NAT Traversal using Teredo

- Teredo vs. ICE
  - Plenty of free Teredo servers available
  - Teredo requires an IPv6 application (+socket opt), but the “magic” happens outside the application
  - ICE is more intrusive for the application because requires changing the protocol semantics

- NAT traversal with HIP
  - HIP-ICE: ICE changes hidden within HIP software
  - HIP-Teredo: no changes to the HIP software
IPv4-IPv6 Interoperability

- **Network layer**
  - Identity-locator split hides underlying access technology from applications
  - Crossfamily handovers from IPv4 to IPv6 and vice versa become trivial

- **Application layer**
  - HITs for applications requesting IPv6
  - LSIs for applications requesting IPv4
Data Plane
HIP and IPsec

- Currently BEET mode ESP is the default
  - Protocol allows to negotiate other modes
  - Implemented in the Linux and BSD kernel
  - Implemented also in user space in Linux and Windows

- Public-key protected data plane (hiccups)
  - Avoids the base exchange and use of IPsec
  - Encapsulates transport layer packets into HIP control packets protected with public key signatures
  - Switching to IPsec by sending an R1
Opportunistic Mode 1/3

- How to support HIP without modifying DNS look ups?
  - Transition mechanism (HITs not in DNS)
  - Opportunistic mode establishes a connection to an unknown HIT

- Identifier problem: what id to use in connect(id) and sendto(id) calls in opportunistic mode?
  - Alternative 1: “pseudo-HIT”
  - Alternative 2: IP address (implemented)

- Where to implement HIP translation?
  - Hooking with LD_PRELOAD to a library
  - Hooking with iptables lower in the stack
Opportunistic Mode 2/3

The diagram illustrates the components involved in the Opportunistic Mode 2/3. It includes:

- **Hostname**: The starting point for the process.
- **Application**: The layer where applications are hosted.
- **Opp. library**: The library responsible for the opportunistic mode.
- **Libc6**: The library for the operating system.
- **Sockets**: The layer for socket communication.
- **Transport**: The layer for transport protocols.
- **Ipsec**: The layer for IP security.
- **Network**: The layer for network communication.
- ** HIP daemon**: The daemon for handling HIP (Host Identity Protocol).

The diagram shows the flow from hostname to the HIP daemon, indicating the process flow in the Opportunistic Mode 2/3.
Opportunistic Mode 3/3

- Opportunistic mode hack: I1 is a TCP option
- Benefit: faster fallback to TCP/IP when peer does not support HIP
- Drawback: works only for TCP, not UDP
- Supported by the implementation already as an implementation extension
HIP Proxy

- Proxy support on an intermediary host
  - Benefit: no changes at client or server side
- Can be implemented on different layers
  - ARP level proxy: Tofino security product based HIP
  - IP level proxy: client and/or server-side proxy
  - HTTP proxy: HIP between the client and proxy
- Can use different naming or routing methods
  - Normal or opportunistic mode
  - Normal IP routing or overlays (e.g. Tofino)
Network Software Engineering
Why not TLS instead of IPsec?

- **Benefits**
  - TLS has wider deployment (HTTPS)
  - TLS-over-TCP passes through NAT/FWs
  - TLS has automatic MTU discovery

- **Drawbacks**
  - (Client-side) mobility support not yet available
  - TLS tunnels cumbersome to set up
  - Same service requires two ports with and without TLS
  - TLS-over-TCP is more prone to e.g. RST attacks
  - Separate protocol for UDP (DTLS)
  - TLS requires to modify the application - both a burden and also the key to TLS success?
Implementation Caveats

- Workarounds for poor multihoming support
  - HIP with a single HIT is a fix to legacy multihoming problems
  - Raw sockets were preferred over UDP sockets for NAT traversal

- IPv4-IPv6 interoperability issues
  - Queuing issues in firewall and hipd (interfamily handovers)
  - LSIs still required because many apps don't support IPv6

- Virtual interfaces
  - Dummy interface (simple) vs. tun/tap (portable)

- HIP loopback
On the Cost of Development

- Kernel vs. userspace development
  - Kernel: fundamental functionality
  - Userspace: rapid prototyping, portability

- Development language
  - C-language (suitable for embedded systems)
  - Python (faster development time, quite portable)

- Version control

- Binary Packaging
Questions?

Miika Komu <miika@iki.fi>

Documentation and software for HIPL:

http://infrahip.hiit.fi/

Other two HIP implementations:

http://www.openhip.org/
http://www.hip4inter.net/
Literature 1/3

- RFC5201, Host Identity Protocol, Jokela et al, Apr 2008
- RFC5206: End-host Mobility and Multihoming with Host Identity Protocol, Henderson et al, Apr 2008
- Integrating Mobility, Multi-homing and Security in a HIP way, Pekka Nikander et al, Feb 2003
Literature 2/3

- Basic Socket Interface Extensions to Host Identity Protocol, Komu et al, Sep 2009
- Overview and Comparison Criteria for Host Identity Protocol and Related Technologies, Koponen et al, Feb 2005
- Leap-of-faith security is enough for IP mobility, Komu et al, Jan 2009
- Enterprise Network Packet Filtering for Mobile Cryptographic Identities, Janne Lindqvist et al, June 2007
- Basic HIP Extensions for Traversal of Network Address Translators, Keränen et al, Oct 2009
Literature 3/3

- Host Identity Protocol (HIP), Towards the Secure Mobile Internet, Gurtov, June 2008
- SSL and TLS: Designing and Building Secure Systems, Rescorla, 2001