Implementing SHIM6 using the Linux XFRM framework

Sébastien Barré, Olivier Bonaventure

Université catholique de Louvain
http://inl.info.ucl.ac.be

Dec. 14th, 2007

Routing in Next Generation Workshop
1 Introduction
   - Introduction to Shim6
   - Shim6: a new layer
   - The REAP exploration protocol

2 The life of an Internet communication...
   - When the TCP SYN is sent...
   - XFRM comes into play
   - Detecting failures: REAP
   - Garbage collection

3 Recent achievements and conclusion
   - CGA support
   - IPv6 at Université catholique de Louvain
   - Conclusion
1 Introduction
- Introduction to Shim6
- Shim6: a new layer
- The REAP exploration protocol

2 The life of an Internet communication...
- When the TCP SYN is sent...
- XFRM comes into play
- Detecting failures: REAP
- Garbage collection

3 Recent achievements and conclusion
- CGA support
- IPv6 at Université catholique de Louvain
- Conclusion
Host-centric multihoming (the context)
Host-centric multihoming (the context)
Host-centric multihoming (the context)
Locators vs Identifiers (ULIDs)

- Application
- Transport
- Network
- Datalink
- Physical

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Locators vs Identifiers (ULIDs)

Application
Transport
Network
Datalink
Physical

IP : Endpoint functions

SHIM

IP : Routing functions
Locators vs Identifiers (ULIDs)

- Application
- Transport
- Network
- Datalink
- Physical

ULPs

IP : Endpoint functions

- SHIM
- IP : Routing functions

IP address = identifier (ULID)

IP address = locator
Locators vs Identifiers (ULIDs)

- ULID: Used as the identifier throughout a transport connection.
- locator: IPv6 address used for routing (locating the peer).
- Shim6 performs a mapping between ULIDs and locators, by use of context tags.
Shim6 operation

'S', ISPX.B

Internet

ISP1

ISP2

Source

Destination

DATA
Shim6 operation

- A, ISP1.A
- 'B', ISPX.B
- ISP1
- ISP2
- DATA
- Destination
- Shim6 negotiation

Implementing SHIM6 using the Linux XFRM framework
Shim6 operation

Introduction to Shim6
Shim6: a new layer
The REAP exploration protocol

Shim6 operation

Internet

'S', ISPX.B

ISP1

ISP2

DATA

Shim6 context
context tag: B_flow1
locators: ISPX.B*
ISP1.A, ISP2.A
cur. ULIDs: ISPX.B
ISP1.A

Shim6 context
context tag: A_flow1
locators: ISP1.A*, ISP2.A
ISPX.B*
cur. ULIDs: ISP1.A
ISPX.B

'Sebastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
REAP operation

Shim6 context
context tag: B_flow1
locators : ISPX.B
ISP1.A*, ISP2.A
cur. ULIDs : ISPX.B
ISP1.A

REAP
OK

Shim6 context
context tag: A_flow1
locators : ISP1.A*, ISP2.A
ISPX.B*
cur. ULIDs : ISP1.A
ISPX.B

REAP
OK

Internet

ISP1

ISP2

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Introduction

The life of an Internet communication...
Recent achievements and conclusion

Introduction to Shim6
Shim6: a new layer
The REAP exploration protocol

REAP operation

Shim6 context
context tag: B_flow1
locators: ISP.X.B*, ISP1.A*, ISP2.A

cur.ULIDs: ISP.X.B, ISP1.A

Internet

ISP1

ISP2

Shim6 context
context tag: A_flow1
locators: ISP1.A*, ISP2.A

cur.ULIDs: ISP.X.B, ISP1.A

REAP probe

DATA

REAP

Expl.

OK

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Implementing SHIM6 using the Linux XFRM framework
REAP operation

Shim6 context
context tag: B_flow1
locators: ISPX.B*
  ISP1.A, ISP2.A*
cur. ULIDs: ISPX.B
  ISP1.A

Shim6 context
context tag: A_flow1
locators: ISPX.B
  ISP1.A, ISP2.A*
cur. ULIDs: ISPX.B
  ISP1.A

REAP Inbd
OK

REAP
OK

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
REAP operation

Shim6 context
context tag: B_flow1
locators: ISPX.B*, ISP1.A, ISP2.A*
cur. ULIDs: ISPX.B, ISP1.A

REAP OK

Shim6 context
context tag: A_flow1
locators: ISP1.A, ISP2.A*
cur. ULIDs: ISPX.B, ISP1.A

REAP OK
TCP connection survival

Figure: Evolution of throughput for an iperf TCP session
LinShim6 is developed at INL (UCLouvain) for two years.

- The implementation now supports almost all the Shim6 draft.
- Version 0.5: Based on IPsec-XFRM framework.
  - Better integration
  - Kernel code minimized.
- CGA support now!
LinShim6 is developed at INL (UCLouvain) for two years.
The implementation now supports almost all the Shim6 draft.
- Version 0.5: Based on IPsec-XFRM framework.
  - Better integration
  - Kernel code minimized.
- CGA support now!
Introduction: LinShim6

LinShim6 is developed at INL (UCLouvain) for two years.

The implementation now supports almost all the Shim6 draft.

Version 0.5: Based on IPsec-XFRM framework.
  - Better integration
  - Kernel code minimized.

CGA support now!
LinShim6 is developed at INL (UCLouvain) for two years.
The implementation now supports almost all the Shim6 draft.
Version 0.5: Based on IPsec-XFRM framework.
- Better integration
- Kernel code minimized.
CGA support now!
1 Introduction
- Introduction to Shim6
- Shim6: a new layer
- The REAP exploration protocol

2 The life of an Internet communication...
- When the TCP SYN is sent...
- XFRM comes into play
- Detecting failures: REAP
- Garbage collection

3 Recent achievements and conclusion
- CGA support
- IPv6 at Université catholique de Louvain
- Conclusion

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Starting an SSH exchange

ISP1.A
ISP2.A

'A'

Implementing SHIM6 using the Linux XFRM framework
Starting an SSH exchange

User space

Kernel space

Physical medium

src : [::] dst : [B]
Starting an SSH exchange

User space

Kernel space

Default address selection (RFC3484)

src : [ISP1.A]  dst : [B]

Physical medium
Starting an SSH exchange

User space

Kernel space

Default address selection (RFC3484)

shim6_pkt_listener.c

Notify

NF_IP6_LOCAL_OUT

src : [ISP1.A] dst : [B]

Physical medium
Introduction

The life of an Internet communication...

Recent achievements and conclusion

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

Starting an SSH exchange

LinShim6 daemon

User space

Kernel space

shim6_pkt_listener.c

create context

New state: [ISP1.A], [B] : 1 pkt

Default address selection (RFC3484)

NF_IP6_LOCAL_OUT

src : [ISP1.A] dst : [B]

Physical medium

Sébastien Barré, Olivier Bonaventure Implementing SHIM6 using the Linux XFRM framework
Starting an SSH exchange
Starting an SSH exchange

LinShim6 daemon

User space

Kernel space

Control flow (I1/I2/R1/R2)

shim6_pkt_listener.c

NF_IP6_LOCAL_IN/
NF_IP6_LOCAL_OUT

local: [ISP1.A]
remote: [B]
Starting an SSH exchange

[Introduction]

The life of an Internet communication...
Recent achievements and conclusion

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

[Implementation]
Implementing SHIM6 using the Linux XFRM framework
The XFRM framework

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

Local: [ISP1.A*], [ISP2.A]
remote: [B*]
REAP state: Operational

LinShim6 daemon

User space

Kernel space
new shim6 transformer
XFRM key manager

NF_IP6_LOCAL_IN/
NF_IP6_LOCAL_OUT

local: [ISP1.A]
remote: [B]

Physical medium
The XFRM framework

LinShim6 daemon

User space

Kernel space

XFRM key manager

XFRM policy check

Shim6 XFRM transformer

Physical medium

Local: [ISP1.A*], [ISP2.A]
remote: [B*]
REAP state: Operational

XFRM comes into play
Detecting failures: REAP
Garbage collection

Recent achievements and conclusion

The life of an Internet communication...
Outbound policy

- Associate a flow with a bundle of transformations.
- Here, policy says:
  
  ```
  if source is [ISP1.A] and dest is [B] then
  Modify output path to go through Shim6 Security Association
  
  The bundle could be AH → ESP → Shim6 (future work).
  ```

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Inbound policy

- Drop the packet if it does not match any policy.
- Here, policy says:

```
if not (source is [B] and dest is [ISP1.A] ) then
  Drop packet
```
Policies vs transformers

- Transformers maintain state (e.g. REAP timers)
- A policy uses generic XFRM code, a transformer has its own code (shim6 address rewriting).
- Shim6 transformers do not necessarily perform translation.
- ... But they always perform failure detection.
Policies vs transformers

- Transformers maintain state (e.g. REAP timers)
- A policy uses generic XFRM code, a transformer has its own code (shim6 address rewriting).
  - Shim6 transformers do not necessarily perform translation.
  - ... But they always perform failure detection.
Policies vs transformers

- Transformers maintain state (e.g. REAP timers)
- A policy uses generic XFRM code, a transformer has its own code (shim6 address rewriting).
- Shim6 transformers do not necessarily perform translation.
- ... But they always perform failure detection.
Policies vs transformers

- Transformers maintain state (e.g. REAP timers)
- A policy uses generic XFRM code, a transformer has its own code (shim6 address rewriting).
- Shim6 transformers do not necessarily perform translation.
- ...But they always perform failure detection.
Failure detection
Failure detection

Introduction
The life of an Internet communication...
Recent achievements and conclusion

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

Local: [ISP1.A*], [ISP2.A]
remote: [B*]
REAP state: Operational

LinShim6 daemon

User space

Kernel space

XFRM policy check

Shim6 XFRM transformer

Physical medium

XFRM policy check

Shim6 XFRM transformer

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Failure detection

- Local: [ISP1.A*], [ISP2.A]
- Remote: [B*]
- REAP state: Operational

LinShim6 daemon

User space

Kernel space

Start exploration

XFRM policy check

Shim6 XFRM transformer

Physical medium

Timeout!
Failure detection

Local: [ISP1.A*], [ISP2.A]
Remote: [B*]
REAP state: Exploring

LinShim6 daemon

User space

Kernel space

probe: [ISP2.A] -> [B]

XFRM policy check

Shim6 XFRM transformer

Physical medium
Failure detection

When the TCP SYN is sent, XFRM comes into play.
Detecting failures: REAP
Garbage collection

Implementing SHIM6 using the Linux XFRM framework

Sébastien Barré, Olivier Bonaventure
Failure detection

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

Implementing SHIM6 using the Linux XFRM framework
Failure detection

Local: [ISP1.A*], [ISP2.A]
remote: [B*]
REAP state: Exploring

LinShim6 daemon

User space

Kernel space

XFRM policy check

Shim6 XFRM transformer

probe: [B]->[ISP2.A]
The life of an Internet communication...

When the TCP SYN is sent... XFRM comes into play
Detecting failures : REAP
Garbage collection

Failure detection

[Diagram showing the process of failure detection with components like LinShim6 daemon, XFRM key manager, and XFRM policy check.]

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Introduction

The life of an Internet communication...
Recent achievements and conclusion

When the TCP SYN is sent...
XFRM comes into play
Detecting failures: REAP
Garbage collection

Failure detection

Local: [ISP1.A], [ISP2.A*]
Remote: [B*]
REAP state: Operational

LinShim6 daemon

[ISP1.A] -> [B]

[a] -> [ISP1.A]

Shim6 XFRM transformer

[ISP2.A] -> [B]

[b] -> [ISP2.A]

Shim6 XFRM transformer

User space

Kernel space

Physical medium

Sébastien Barré, Olivier Bonaventure
Implementing SHIM6 using the Linux XFRM framework
Failure detection
Garbage collecting a Shim6 context

Local: [ISP1.A], [ISP2.A*]
remote: [B*]
REAP state: Operational

LinShim6 daemon

User space

Kernel space: Transformer no longer used

XFRM key manager

XFRM policy check

XFRM policy check

Shim6 XFRM transformer

Shim6 XFRM transformer

Physical medium
Garbage collecting a Shim6 context

User space

Kernel space

Remove Shim6 states/policies

XFRM key manager

XFRM policy check

Shim6 XFRM transformer
Implementing SHIM6 using the Linux XFRM framework
1 Introduction
   - Introduction to Shim6
   - Shim6: a new layer
   - The REAP exploration protocol

2 The life of an Internet communication...
   - When the TCP SYN is sent...
   - XFRM comes into play
   - Detecting failures: REAP
   - Garbage collection

3 Recent achievements and conclusion
   - CGA support
   - IPv6 at Université catholique de Louvain
   - Conclusion
CGA now supported!

- Implementation is based on DoCoMo SEcure Neighbor Discovery
  - [http://www.docomolabs-usa.com/lab_opensource.htm](http://www.docomolabs-usa.com/lab_opensource.htm)
- CGA daemon auto-generates CGA based on RAs
- LinShim6 now only accepts secured addresses.
CGA now supported!

- Implementation is based on DoCoMo SEcure Neighbor Discovery
  - [http://www.docomolabs-usa.com/lab_opensource.htm](http://www.docomolabs-usa.com/lab_opensource.htm)
- CGA daemon auto-generates CGA based on RAs
- LinShim6 now only accepts secured addresses.
CGA now supported!

- Implementation is based on DoCoMo SEcure Neighbor Discovery
  - http://www.docomolabs-usa.com/lab_opensource.htm
- CGA daemon auto-generates CGA based on RAs
- LinShim6 now only accepts secured addresses.
IPv6 state at UCL

- Native IPv6 through Belnet.
- New sixxs tunnel to Easynet obtained recently
  - All our department will soon be IPv6-multihomed. (one-two weeks)
  - Only 3 – 4 ms RTT to the tunnel broker.
  - http://www.sixxs.net/
- Shim6 experimental server : Future.
IPv6 state at UCL

- Native IPv6 through Belnet.
- New sixxs tunnel to Easynet obtained recently
  - All our department will soon be IPv6-multihomed. (one-two weeks)
  - Only 3 – 4 ms RTT to the tunnel broker.
  - http://www.sixxs.net/
- Shim6 experimental server : Future.
Native IPv6 through Belnet.
New sixxs tunnel to Easynet obtained recently
  - All our department will soon be IPv6-multihomed. (one-two weeks)
  - Only 3 – 4 ms RTT to the tunnel broker.
  - http://www.sixxs.net/

Shim6 experimental server : Future.
The XFRM framework has several advantages:

- Less Shim6 specific code
- Easier interoperation with IPsec
- Kernel parts: Only those critical for efficiency.
- CGA support.

Future/ongoing work:

- Provide a public environment for Shim6 experiments/measurements.
- Stabilize the implementation (need for feedback!), make it user friendlier.
- HBA support.
Acknowledgements

- USAGI team: Shinta Sugimoto, Masahide Nakamura
- DoCoMo: SEND implementation
- LinShim6 users: John Ronan, Lu Junxiu, ENST-Bretagne
- And several others, thanks for fruitful discussions!
User space

Kernel space

Questions?

Physical medium
Shim6 XFRM transformers vs Netfilter

User space

Kernel space

Physical medium

Local: [ISP1.A], [ISP2.A*]
remote: [B*]
REAP state: Operational

LinShim6 daemon

XFRM policy check

NF_IP6_LOCAL_OUT

Shim6 XFRM transformer

XFRM policy check

Shim6 XFRM transformer

NF_IP6_LOCAL_IN