

Introduction to Labeled Networking on Linux

Paul Moore

paul.moore@hp.com



Agenda

- Labeled Networking Basics
- Types of Labeled Networking
 - Secmark
 - NetLabel
 - Labeled IPsec
- Linux Security Modules and Labeled Networking
 - SELinux
 - Smack
- Labeled Networking Development

Labeled Networking Basics



Labeled Networking Introduction

- Labeled networking is a form of network access control based on security labels
 - Security labels assigned to network traffic
 - Labels represent both network and security attributes
 - Network traffic identified by its security label
 - Security policy applied to network traffic
 - Defines access rights for network traffic and applications
- Integrates network access controls into the Linux Security Module (LSM) mechanism
 - Requires a labeled security implementation
 - SELinux
 - Smack

Labeled Networking on Linux

- Linux supports two different network label types
 - Secmark labels represent network attributes
 - Netfilter/iptables used to assign labels to packets based on Netfilter matching rules
 - Peer labels represent sender's security attributes
 - Labeling protocols convey the sender's security label across the network
- Labeled networking support varies by LSM
 - SELinux
 - Supports both secmark and peer network labels
 - Smack
 - Supports Peer network labels

Secmark Network Labels

- Secmark labels are locally assigned based on network attributes
 - Netfilter/iptables rule matching is used to assign secmark labels to network packets
 - Flexible assignment of labels using existing Netfilter packet matching and connection tracking
 - Single secmark packet labeling mechanism can be shared across multiple LSMs
 - Does not require any external infrastructure
 - Secmark labels are assigned locally, no labeling protocol needed
- Integrates traditional Linux firewall functionality with the LSM security model

Peer Network Labels

- Peer labels are assigned based on the security attributes of the sender
 - Labeling protocols convey security attributes
 - Commercial IP Security Option (CIPSO)
 - Labeled IPsec
 - Protocol support is required for full functionality
 - Fallback peer label support for unlabeled networks
 - Two peer labeling mechanisms available
 - NetLabel framework
 - Labeled IPsec
- Extends the LSMs labeled security model across the network

Secmark



Secmark Labeling

- Labeling individual packets

- Command format

```
iptables -t mangle -A <CHAIN> <RULES> -j SECMARK \  
--selctx <SECMARK_LABEL>
```

- CHAIN : Netfilter/iptables chain
 - RULES : Netfilter/iptables traffic matching rules
 - SECMARK_LABEL : Secmark label

- Example

```
iptables -t mangle -A INPUT -p tcp --dport 22 \  
-j SECMARK --selctx system_u:object_r:ssh_packet_t:s0
```

- Label an entire connection

- Traditional Netfilter connection marking using CONNSECMARK

NetLabel



NetLabel Framework

- NetLabel is a labeled networking framework
 - Designed for multiple LSMs and protocols
- Provides interoperable labeled networking
 - CIPSO protocol provides labeled networking interoperability with other Trusted OSs
 - Trusted Solaris, HP-UX CMW, and others
 - Limited to Multi-Level Security (MLS) attributes
- Provides fallback network peer labels
 - Peer labels for systems without protocol support
 - Allows peer labels to be assigned to both single hosts and entire networks
 - Supports the full LSM security label

CIPSO Configuration

- Define a CIPSO Domain of Interpretation (DOI)

- Command format

- ```
netlabelctl cipsov4 add pass doi:<DOI> tags:<TAG_LIST>
```

- DOI : CIPSO DOI number
    - TAG\_LIST : Comma delimited list of CIPSO tag numbers

- Example

- ```
netlabelctl cipsov4 add pass doi:8 tags:1,5,2
```

- Map a CIPSO DOI configuration to a LSM domain

- Command format

- ```
netlabelctl map domain:<DOMAIN> protocol:cipsov4,<DOI>
```

- DOMAIN : LSM domain string
    - DOI : CIPSO DOI number

- Example

- ```
netlabelctl map domain:ping_t protocol:cipsov4,8
```

Fallback Peer Label Configuration

- Define a CIPSO Domain of Interpretation (DOI)

- Command format

```
netlabelctl unlbl add interface:<IFACE> address:<ADDR> \  
label:<PEER_LABEL>
```

- IFACE : Network interface
 - ADDR: Network address with optional mask
 - PEER_LABEL: Fallback peer label

- Example

```
netlabelctl unlbl add interface:eth0 \  
address:192.168.0.0/16 \  
label:system_u:object_r:netlabel_peer_t:s0
```

Labeled IPsec



Labeled IPsec

- IPsec Security Associations (SA) assign peer labels to network traffic
 - Peer labels transferred between systems during IKE exchange
 - Network traffic is implicitly labeled by matching SAs
 - Provides peer labeling with packet level encryption and authentication
- Interoperability limited to SELinux systems
 - Requires SELinux specific IKE extensions
 - Conflicts with Explicit Congestion Notification (ECN)
 - Supports the full SELinux security label

Labeled IPsec Configuration

- Create a Labeled IPsec policy

- Command format

```
echo "spdadd <SOURCE> <DEST> any -ctx 1 1 \  
    system_u:object_r:ipsec_spd_t:s0-s15:c0.c1023  
    -P <POLICY>" | setkey -c
```

- SOURCE : Source IP address
 - DEST : Destination IP address
 - POLICY : IPsec policy

- Example

```
echo "spdadd 192.168.0.10 192.168.0.11 any -ctx 1 1 \  
    system_u:object_r:ipsec_spd_t:s0-s15:c0.c1023  
    -P in esp/transport//require" | setkey -c
```

- Labeled SAs created on demand by the IKE daemon

SELinux Labeled Networking



Labeled Networking in SELinux

- SELinux supports all of the Linux labeled networking mechanisms
 - Secmark
 - NetLabel
 - CIPSO support for IPv4
 - Fallback peer label support for IPv4 and IPv6
 - Labeled IPsec support for IPv4 and IPv6
- Consolidated NetLabel and Labeled IPsec labels
 - Peer labels must be equivalent to be allowed
- Dynamic labeled network access controls
 - Access checks only enabled when needed by configuration

Inbound Locally Destined Traffic

- Network labeling points
 - NetLabel or Labeled IPsec provides peer label
 - Netfilter configuration provides secmark label
- Network traffic access control points
 1. Traffic with *peer label* allowed to enter the system via the inbound interface?
 2. Traffic with *peer label* allowed to enter the system from the traffic's source address?
 3. Socket allowed to receive traffic with *peer label*?
 4. Socket allowed to receive traffic with *secmark label*?

Outbound Locally Generated Traffic

- Network labeling points
 - Peer label taken from the sending socket
 - Netfilter configuration provides secmark label
- Network traffic access control points
 1. Socket allowed to send traffic with *secmark label*?
 2. Traffic with *peer label* allowed to exit the system via the outbound interface?
 3. Traffic with *peer label* allowed to exit the system with the traffic's destination address?

Inbound Forwarded Traffic

- Network labeling points
 - NetLabel or Labeled IPsec provides peer label
 - Netfilter configuration provides secmark label based on inbound packet
- Inbound traffic access control points
 1. Traffic with *peer label* allowed to enter the system via the inbound interface?
 2. Traffic with *peer label* allowed to enter the system from the traffic's source address?
 3. Traffic with *peer label* allowed to be forwarded with (*inbound*) *secmark label*?

Outbound Forwarded Traffic

- Network labeling points
 - NetLabel or Labeled IPsec provides peer label
 - Netfilter configuration provides secmark label based on outbound packet
- Outbound traffic access control points
 1. Traffic with *peer label* allowed to be forwarded with (*outbound*) *secmark label*?
 2. Traffic with *peer label* allowed to exit the system via the outbound interface?
 3. Traffic with *peer label* allowed to exit the system with the traffic's destination address?

Smack Labeled Networking



Labeled Networking in Smack

- Smack currently provides limited labeled networking support
 - NetLabel
 - CIPSO support for IPv4
 - Fallback label support for IPv4 and IPv6
 - Not configurable using native Smack tools
- Smack network access control points limited to local network traffic
- Smack labeled networking functionality is expected to improve as Smack matures
 - Smack first included in kernel 2.6.25

Traffic Control Points

- Local network traffic
 - Inbound network traffic access checks
 - Network labeling points
 - NetLabel provides peer label
 - Inbound traffic access control point
 - Traffic with *peer label* allowed to write to the receiving socket?
 - Outbound network traffic access checks
 - Traffic is labeled based on originating socket
 - No access control is applied to outbound traffic
- Forwarded network traffic
 - No labeling or access control

Labeled Networking Development



Recent Labeled Networking Work

- **NetLabel and Labeled IPsec consolidation**
 - Unified network peer label and access controls
 - Easier SELinux policy development
- **New network ingress and egress controls**
 - Access control for local and forwarded traffic
 - Access control for physical network interfaces, subnetworks, and individual hosts
- **New fallback peer labels for unlabeled traffic**
 - Peer labels when protocol support is missing
 - Assign a single peer label to an entire network or a single host

Planned Labeled Networking Work

- NetLabel traffic labeling based on sender's LSM domain and destination address
 - Currently based only on sender's LSM domain
- Improved loopback peer labeling
 - Extend NetLabel/CIPSO to support native LSM labels for loopback traffic
 - Labeled IPsec is slow and problematic over loopback
- Standards body efforts
 - IETF CALIPSO specification
 - Multi-Level Security (MLS) labeling protocol for IPv6
 - Starting point for a generic peer labeling protocol
 - IETF Labeled IPsec specification

More Information

- **NetLabel Website**

<http://netlabel.sourceforge.net>

- **SELinux Wiki**

<http://selinuxproject.org>

- **Smack Website**

<http://www.schaufler-ca.com>

- **Presenter's Email**

paul.moore@hp.com

