Encrypting OVN tunnels with IPsec

Qiuyu Xiao (qiuyu.xiao.qyx@gmail.com)
Ben Pfaff (blp@ovn.org)
Open Virtual Network (OVN)

OVN provides a logical network abstraction on top of a physical network.
Open Virtual Network (OVN)

VMs are oblivious to the physical network states
Open Virtual Network (OVN)

Network appliances can be implemented and placed in the logical network.
OVN Tunnel Traffic
OVN Tunnel Traffic

<table>
<thead>
<tr>
<th>Outer Ethernet Header</th>
<th>Outer IP Header</th>
<th>Outer UDP Header</th>
<th>Geneve Header</th>
<th>Inner Ethernet Header</th>
<th>Inner IP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

![Diagram showing Hypervisor 1 connected to VM1, VM6, VM2, VM7, and Hypervisor 2 connected to VM8, VM3, VM4, VM9, VM5.](attachment:ovn_tunnel_traffic_diagram.png)
# OVN Tunnel Traffic

<table>
<thead>
<tr>
<th>Outer Ethernet Header</th>
<th>Outer IP Header</th>
<th>Outer UDP Header</th>
<th>Geneve Header</th>
<th>Inner Ethernet Header</th>
<th>Inner IP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

![Diagram showing OVN Tunnel Traffic]

- VM1
- VM2
- VM3
- VM4
- VM5
- VM6
- VM7
- VM8
- VM9

Hypervisor 1 is connected to Hypervisor 2, and each hypervisor is connected to multiple VMs.
OVN Tunnel Traffic

<table>
<thead>
<tr>
<th>Outer Ethernet Header</th>
<th>Outer IP Header</th>
<th>Outer UDP Header</th>
<th>Geneve Header</th>
<th>Inner Ethernet Header</th>
<th>Inner IP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

Diagram:
- Hypervisor 1
- Hypervisor 2
- VM1
- VM2
- VM3
- VM4
- VM5
- VM6
- VM7
- VM8
- VM9

Outer Ethernet Header
Outer IP Header
Outer UDP Header
Geneve Header
Inner Ethernet Header
Inner IP Header
Payload
OVN Tunnel Traffic

<table>
<thead>
<tr>
<th>Inner Ethernet Header</th>
<th>Inner IP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

Diagram:
- VM1, VM6 connected to Hypervisor 1
- VM2, VM7 connected to Hypervisor 2
- VM8, VM3, VM4 connected to Hypervisor 1
- VM9, VM5 connected to Hypervisor 2
The Needs for Tunnel Encryption

- VMs compute and communicate sensitive data, e.g., financial and health data
- Physical network devices (e.g., router, switch) cannot be trusted or might be compromised
  - Traffic across datacenters
  - Router misconfiguration
  - Attackers breaking into internal network
  - Phishing or social engineering attacks on administrators
### Encrypting Tunnel Traffic with IPsec

<table>
<thead>
<tr>
<th>Outer Ethernet Header</th>
<th>Outer IP Header</th>
<th>Outer UDP Header</th>
<th>Geneve Header</th>
<th>Inner Ethernet Header</th>
<th>Inner IP Header</th>
<th>Payload</th>
</tr>
</thead>
</table>

**IPsec Encryption**

- Confidentiality
- Integrity
- Authenticity
IPsec in Linux

- IKE daemon
- IKE protocol
- User space
- Kernel
- security policy
- security association
- IPsec kernel stack
- ESP/AH protocol
IPsec in Linux

IKE daemon
- Authentication
- Negotiates cryptographic algorithms
- Generates keying material

IKE daemon

IKE protocol

IPsec kernel stack

ESP/AH protocol

User space
Kernel

security policy
security association
IPsec in Linux

IKE daemon
- Authentication
- Negotiates cryptographic algorithms
- Generates keying material
- Installs security policy and security association

IKE daemon
- IKE protocol
  - User space
    - Kernel
      - security policy
      - security association

IPsec kernel stack
- ESP/AH protocol
IPsec in Linux

IKE daemon
• Authentication
• Negotiates cryptographic algorithms
• Generates keying material
• Installs security policy and security association

Which traffic to protect
IPsec in Linux

IKE daemon
- Authentication
- Negotiates cryptographic algorithms
- Generates keying material
- Installs security policy and security association

How to protect the selected traffic

IKE daemon

IKE protocol

User space
Kernel

security policy
security association

IPsec kernel stack

ESP/AH protocol
IPsec in Linux

IPsec kernel stack
• Encryption and decryption
• Checks integrity and authenticity
OVS IPsec Tunnel

ovsdb → ovs-monitor-ipsec → IKE daemon

User space
Kernel

ovs datapath ↔ IPsec kernel stack
OVS IPsec Tunnel

Configuring IPsec tunnel via ovsdb
• Using pre-shared key

For example:

```
root@ubuntu:~# ovs-vsctl add-port br-int tun \
  -- set interface tun type=geneve \ 
  options:local_ip=10.33.78.172 \ 
  options:remote_ip=10.33.79.149 \ 
  options:psk=swordfish
```
OVS IPsec Tunnel

Configuring IPsec tunnel via ovsdb
- Using pre-shared key
- Using self-signed certificate

For example:

```
root@vm1:~# ovs-vsctl set Open_vSwitch .
   > other_config:certificate=/etc/ipsec.d/certs/vm1-cert.pem
   > other_config:private_key=/etc/ipsec.d/private/vm1-privkey.pem

root@vm1:~# ovs-vsctl add-port br-int int tun
   > -- set interface tun type=_geneve
   > options:local_ip=10.33.78.172
   > options:remote_ip=10.33.79.149
   > options:remote_cert=/etc/ipsec.d/certs/vm2-cert.pem
```
OVS IPsec Tunnel

Configuring IPsec tunnel via ovsdb
- Using pre-shared key
- Using self-signed certificate
- Using CA-signed certificate

For example:

```
root@vml:~# ovs-vsctl set Open_vSwitch .
> other_config:certificate=/etc/ipsec.d/certs/vml-cert.pem
> other_config:private_key=/etc/ipsec.d/private/vml-privkey.pem
> other_config:ca_cert=/etc/ipsec.d/cacerts/ca-cert.pem

root@vml:~# ovs-vsctl add-port br-int tun
> -- set interface tun type=geneve
> options:local_ip=10.33.78.172
> options:remote_ip=10.33.79.149
> options:remote_name=vm2
```
OVS IPsec Tunnel

Establishing IPsec tunnel
• ovs-monitor-ipsec configures IKE daemon

![Diagram of OVS IPsec Tunnel]

- ovsdb
- ovs-monitor-ipsec
- IKE daemon
- ovs datapath
- IPsec kernel stack

User space
Kernel

security policy
security association
OVS IPsec Tunnel

Establishing IPsec tunnel
- ovs-monitor-ipsec configures IKE daemon
- IKE daemon sets up security policy and security association
OVS IPsec Tunnel

Establishing IPsec tunnel
- ovs-monitor-ipsec configures IKE daemon
- IKE daemon sets up security policy and security association

For example (geneve tunnel):
OVS IPsec Tunnel

IPsec kernel stack
• Encryption and decryption
• Checks integrity and authenticity
OVN IPsec

- northbound db
  - ovn-northd
  - southbound db
    - ovn-controller
      - ovsdb
      - vswitchd
    - ovn-controller
      - ovsdb
      - vswitchd

Hypervisor 1
Hypervisor n
OVN IPsec

• In each hypervisor, configure ovsdb to use CA-signed certificate for authentication
• Enable IPsec by configuring northbound database

For example:

```
root@ubuntu:~# ovn-nbctl set nb_global.ipsec=true
```
IPsec Evaluation

- Environment: StrongSwan 5.3.5, Linux 4.4.0, Intel Xeon 2 GHz, 10 Gbps NIC
- iperf generates TCP stream (window size: 85KB), which is encrypted in a single core
IPsec Evaluation

- Environment: StrongSwan 5.3.5, Linux 4.4.0, Intel Xeon 2 GHz, 10 Gbps NIC
- iperf generates TCP stream (window size: 85KB), which is encrypted in a single core

Throughput (Mbps)

- aes256-sha256
- aes-gcm
- no encryption

CPU Usage

- iperf-client
- iperf-server
IPsec Evaluation

- Environment: StrongSwan 5.3.5, Linux 4.4.0, Intel Xeon 2 GHz, 10 Gbps NIC
- iperf generates TCP stream (window size: 85KB), which is encrypted in a single core

![Throughput (Mbps)](image1)

![CPU Usage](image2)
IPsec Evaluation

- Environment: StrongSwan 5.3.5, Linux 4.4.0, Intel Xeon 2 GHz, 10 Gbps NIC
- iperf generates TCP stream (window size: 85KB), which is encrypted in a single core

![Throughput (Mbps)](chart-throughput)

![CPU Usage](chart-cpu-usage)
Current Status

- Compatible with StrongSwan and LibreSwan IKE daemon
- Packages for Ubuntu and Fedora
- Tutorials on using OVN IPsec
- Need to use OVS upstream kernel module
Future Directions

More flexible tunnel encryption policies:
• Only encrypting tunnel traffic between certain hypervisors
• Only encrypting tunnel traffic from certain logical network