Enabling Practical SDN Security Applications with OFX (The OpenFlow eXtension Framework)

John Sonchack, Adam J. Aviv, Eric Keller, and Jonathan M. Smith
Outline

Introduction
Overview of OFX
Using OFX
Benchmarks
Basic Networking: Forwarding and Routing

Packet Forwarding

Route Computation
SDNs: Networking in Two Planes

Route computation

Packet forwarding

Data Plane

Control Plane

OpenFlow

Packet forwarding

Route computation
OpenFlow: A Protocol to Manage Switches

- Route computation
- Flow rules to implement routes
- Packet forwarding
OpenFlow: A Protocol to Manage Switches

Assumption: Interactions between the control plane and data plane are infrequent.
SDNs for Network Security

Access Control Policy

Flow rules to implement access control policy

SDNs for **Dynamic** Network Security

- Traffic Declassification
- Access Control
- DDoS Defense
- Bot Detection

**Control Plane**
- Advanced Processing
- Route for flow

**Data Plane**
- Packet from new flow
SDNs for **Dynamic** Network Security: Flow Monitoring


Collect flow records without routing through a middlebox.

Packet from new TCP flow

Install byte counting rule
SDNs for **Dynamic** Network Security: Traffic Declassification

SDNs for **Dynamic** Network Security

- Traffic Declassification
- Access Control
- DDoS Defense
- Bot Detection

Data Plane

- Packet from new flow
- Route for flow

Control Plane

- OpenFlow
- Advanced Processing
SDNs for **Dynamic** Network Security

**Control Plane**

- Traffic Declassification
- Access Control
- Bot Detection

**Data Plane**

- Advanced Processing
- Route for flow

Assumption: Interactions between the control plane and data plane are *infrequent*. 

Packet from new flow
Obstacle: Low Throughput

Control Path

130 million packets/second!!!!*

*can only forward 500 pps to controller.


Obstacle: Centralized Control Plane
Our question: How Can We Make SDNs More Practical?

- Traffic Declassification
- Access Control
- DDoS Defense
- Bot Detection

Data Plane

Control Plane
The General Approach: Switch Level Security

- Traffic Declassification
- Access Control
- DDoS Defense
- Bot Detection
Previous Work: Security Functionality in the Forwarding Engine

Build new switch chips that support security applications

Our insight: Leverage Switch CPUs

Run security logic on the switch CPUs
OFX: A Framework for Application-Specific Switch Extensions

Each application can load custom functionality into switches. At runtime!
Outline

Introduction

Overview of OFX

Using OFX

Benchmarks
OFX at a High Level
OFX at a High Level
OFX at a High Level

Controller interface

OFX Extension Module

Switch-level logic
OFX at a High Level

Permissions Database

Declassifier Module

Per-Flow Declassification Logic

OFX stack

OpenFlow stack
OFX at the Switch Level

OFX modules use filters to select packets that they need to process.

OFX installs corresponding rules onto OFX tables.

OFX modules process packets with custom handler.

OFX modules use filters to select packets that they need to process.

OFX installs corresponding rules onto OFX tables.

OFX modules process packets with custom handler.

OFX at the Switch Level

OFX Filtering Tables

Controller-managed forwarding tables

Egress Packets

Ingress Packets
Outline

Introduction

Overview of OFX

Using OFX

Benchmarks
Refactoring OpenFlow Applications to use OFX
Refactoring OpenFlow Applications to use OFX

```python
import OFXLib

class DeclassifierApp(app_manager.RyuApp):
    def __init__(self, *args, **kwargs):
        super(SimpleSwitch13, self).__init__(*args, **kwargs)
        self.permissionsDb = dbServer.connect()
        self.monitoredServers = []
        self.switchIds = []

        self.declassifierModule = OFXLib.load_module("dec_module")
        self.declassifierModule.permissions = self.permissionsDb

    def switch_up_handler(self, switch):
        self.switchIds.append(switch.id)
        OFXLib.install(switch, self.declassifierModule)

    def packet_handler(self, switch, pkt):
        action = self.compute_next_hop(pkt, switch)
        switch.send_packet(pkt, action)
```

OFX Declassifier Module
Outline

Introduction
Overview of OFX
Using OFX
Benchmarks
Benchmarking OFX

How much raw overhead is there for processing packets with OFX?

How do OFX based security applications perform, compared with Middlebox and OpenFlow implementations?
OFX Benchmark: Packets Per Second

Packet handler in controller
Packet handler in OFX module

Packet Size

100,000
10,000
1,000
100
10
1

Log^{10} Scale

Log_{10} Scale

Packets per Second

100 PPS @ MTU
45,000 PPS @ MTU
Benchmarking OFX

How much raw overhead is there for processing packets with OFX?

How do OFX based security applications perform, compared with Middlebox and OpenFlow implementations?
### Benchmark: Declassifier Packet Drop Rate

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Frequent arriving flows</th>
<th>Median</th>
<th>High bandwidth flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middlebox Proxy</td>
<td>0.1%</td>
<td>0.1%</td>
<td>20.4%</td>
</tr>
<tr>
<td>OpenFlow</td>
<td>97.5%</td>
<td>88.2%</td>
<td>0.1%</td>
</tr>
<tr>
<td>OFX</td>
<td>5.1%</td>
<td>3.2%</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

- **OpenFlow implementation limited by flow arrival rate**
- **Proxy implementation limited by bit rate**

**OFX implementation performed well in all workloads**

### Workload Name

<table>
<thead>
<tr>
<th>Workload Name</th>
<th>Frequent arriving flows</th>
<th>Median flows</th>
<th>High bandwidth flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow Inter-arrival Period</td>
<td>0.0015 Seconds</td>
<td>0.015 Seconds</td>
<td>0.15 Seconds</td>
</tr>
<tr>
<td>Average Transmission Bandwidth</td>
<td>19.75 Mbps</td>
<td>43.57 Mbps</td>
<td>970.99 Mbps</td>
</tr>
</tbody>
</table>

In the Paper

**OFX API and Implementation Details**

**Application Specific Modules**
- DDoS Defense
- Bot Detection

**Enhanced Switch API Modules**
- TCP Handshake Validation
- Push Based Alerts

**More benchmarks**

- Running on unmodified OpenFlow hardware!

**OFX API and Implementation Details**

- OpenFlow Controller
- OpenFlow Switch
- OpenFlow Packet Path
- Control Platform
- Linux Kernel
- OFX Library
- OFX Agent
- Forwarding Engine Firmware
- Linux Network Stack
- OFX Packet Path

**Application Specific Modules**

- DDoS Defense
- Bot Detection

**Enhanced Switch API Modules**

- TCP Handshake Validation
- Push Based Alerts

**More benchmarks**

- Running on unmodified OpenFlow hardware!
OFX lets OpenFlow security applications push parts of their control plane logic down to switch CPUs, which can greatly improve performance and scalability on existing hardware and software.