Network Programming in a Cisco Open Network Environment
Start using onePK and EEM
T-SDN4/L3

Bruno Klauser
Consulting Engineer
BN EMEAR CTO Team
bklauser@cisco.com
An Analogy

Highly motivated individuals
Full control over every single detail

Highly skilled and trained crew
Human brain in every control loop

Specialized distributed crew
Reasonable control within boundaries

From: Detailed control by a single central authority
Towards: Collaborative operations of a partially autonomic system
“Computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity,
and especially because it produces objects of beauty”

Donald Knuth, 1974
(Author of “The Art of Computer Programming”)
Cisco Open Network Environment
Describing Software Architectures

4+1 View Model

<table>
<thead>
<tr>
<th>Scenarios:</th>
<th>Use Case, Who, What and Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples:</td>
<td>Routing-for-(Dollars), Application-Flow-Manipulation, Network Slicing, SDDC Provisioning, CIN, …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Development View:</th>
<th>SDKs, Packages, Libraries, Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples:</td>
<td>onePK SDK, binaries, .so, Eclipse, Debugger, IOL, …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logical View:</th>
<th>Features, Functions, Classes, Abstractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples:</td>
<td>onePK APIs in C, Java, Application APIs, …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process View:</th>
<th>Architectures, Processes, Interactions, Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples:</td>
<td>Agents, Controllers, Thrift IDL, Cloud Connectors, Sentinels, …</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Physical View:</th>
<th>Deployment, Hosting, Topology, Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples:</td>
<td>Virtual Containers, Blades, Endpoints, L1-2-3, Overlay</td>
</tr>
</tbody>
</table>

See: [https://en.wikipedia.org/wiki/4+1_Architectural_View_Model](https://en.wikipedia.org/wiki/4+1_Architectural_View_Model)
Cisco Open Network Environment – ONE

Preserve What is Working
- Resilience, Scale, Security
- Functionality and Rich Features
- Instrumentation

Evolve for New Requirements
- Operational Simplicity and Automations
- Programmability and Network-Awareness
- Upcoming Innovations

Open and Integrated Framework
- Software Defined Network concepts are a component of the Open Network Environment
- Existing APIs, Agents, Controllers and Infrastructure contribute

Open Network Environment
- Development View
  - Logical View
- Process View
  - (Software) Architectures and Patterns
- Physical View
  - Deployment and Virtualization

Scenarios and Motivations
Cisco Open Network Environment – ONE

Preserve What is Working
- Resilience, Scale, Security
- Functionality and Rich Features
- Instrumentation

Evolve for New Requirements
- Operational Simplicity and Automations
- Programmability and Network-Awareness
- Upcoming Innovations

Open and Integrated Framework
- Software Defined Network concepts are a component of the Open Network Environment
- Existing APIs, Agents, Controllers and Infrastructure contribute

Open Network Environment

<table>
<thead>
<tr>
<th>Network Programming</th>
<th>(Software) Architectures and Patterns</th>
<th>Deployment and Virtualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>onePK</td>
<td>Controllers (ONE/Openflow PoC)</td>
<td>Nexus 1000v</td>
</tr>
<tr>
<td>developer.cisco.com, CDN, Training, Certification, Partners, EEM, EASy</td>
<td>(SBC, WLC, +++), CIN, CloudConnect, Sentinels, Agents</td>
<td>CSR 1000v, VSG and vFW/ASA, vWAAS, vNAM, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cisco Openstack Ed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blade Hosting (UCS-E, …), Virtual Containers (AirVision, Cat, ISR, ASR, …)</td>
</tr>
</tbody>
</table>

Scenarios and Motivations
Cisco Open Network Environment – ONE

Preserve What is Working
- Resilience, Scale, Security
- Functionality and Rich Features
- Instrumentation

Evolve for New Requirements
- Operational Simplicity and Automations
- Programmability and Network-Awareness
- Upcoming Innovations

Open and Integrated Framework
- Software Defined Network concepts are a component of the Open Network Environment
- Existing APIs, Agents, Controllers and Infrastructure contribute

<table>
<thead>
<tr>
<th>Open Network Environment</th>
<th>Network Programming</th>
<th>(Software) Architectures and Patterns</th>
<th>Deployment and Virtualization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>onePK</td>
<td>Controllers (ONE/Openflow PoC)</td>
<td>Nexus 1000v</td>
</tr>
<tr>
<td></td>
<td>developer.cisco.com, CDN, Training, Certification, Partners, EEM, EASy</td>
<td>(SBC, WLC, +++ )</td>
<td>CSR 1000v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIN, CloudConnect, Sentinels, Agents</td>
<td>VSG and vFW/ASA, vWAAS, vNAM, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cisco Openstack Ed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Blade Hosting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(UCS-E, … ), Virtual Containers (AirVision, Cat, ISR, ASR, …)</td>
</tr>
</tbody>
</table>

Scenarios and Motivations
## Network Automation and Manageability

**Cisco IOS® Device Manageability Instrumentation (DMI)**

### Fault
- **IP OAM**—Ping, Trace, BFD, ISG per session
- **802.3ah**—Link monitoring and remote fault indication
- **802.1 ag**—Continuity check, L2 ping, trace, AIS
- **MPLS OAM**—LSP ping, LSP trace, VCCV
- **EEM**—Embedded Event Manager
- **EVENT-MIB**—OID-based triggers, events, or SNMP Set, IETF DISMON
- **EXPRESSION-MIB**—OID expression-based triggers, IETF DISMON
- ...  

### Configuration
- **Config CLI**—diff, logging, lock, replace, rollback
- **E-LMI**—parameter and status signaling
- **E-DI**—Enhanced Device Interface, CLI, Perl, IETF Netconf
- **EMM**—Embedded Menu Manager
- **NETCONF**—IETF NETCONF XML PI
- **CNS and WSMA**
- **TR-069**
- **KRON**—command scheduler
- **AutoInstall**—bootstrapping
- **IOS.sh**—IOS Shell
- **SmartInstall**
- **Auto SmartPorts**

### Performance
- **Auto IP SLA**—delay, jitter, loss probability
- **CBQoS MIB**—class-based QoS
- **NBAR**
- **RMON**
- **EPC**—Embedded Packet Capture
- **ERM**—Embedded Resource Manager
- **GOLD**—Generic Online Diagnosis
- **Smart Call Home**—preventive maintenance
- **VidMon**—Video Monitoring
- ...

### Accounting
- **Flexible NetFlow**—IETF IPFIX
- **BGP policy accounting**—includes AS information
- **Periodic MIB bulk data collection and transfer**
- ...

### Security
- **Auto Secure**—one-touch device hardening
- **LDP Auth**—message authentication
- **Routing Auth**—MD5 authentication, BGP, OSPF
- ...

---

**Device Manageability Instrumentation Has Evolved Significantly**
Network Automation Life-Cycle Span

Is it built to Specification?

Does it meet Requirements?

Is there room for yet another service?

How to configure?

Is it working as specified?

How to take out of service?

Network Automation spans across full Life Cycle
**Problem:** Sometimes we want to receive remote information on a Router / Switch and be able to react to it locally – for example a notification from a UPS System.

**Solution:** Use Network Automation based on Cisco IOS Embedded Event Manager leveraging the EEM SNMP Notification Event Detector

- Router / Switch can received SNMP Notifications
- Execute (trigger) EEM Policy to take local action
- Policy can query varbind info
- Supports Incoming or Outgoing Notifications
- Outgoing only for locally generated Notifications

```bash
Router(config)# event manager applet catch-a-trap
router(config-applet)# description test snmp notification unmanaged service
router(config-applet)# event snmp-notification oid 1.3.6.1.6.3.1.1.4.1.0
  oid-val "1.3.6.1.6.3.1.1.5.3" op eq src-ip-address 10.51.89.176
direction incoming
router(config-applet)# action 010 ...
router(config-applet)# action 020 ...
```
Trigger a Workflow from the Network

**Problem:** A new rogue WLAN device in sensitive areas should be detected by Cisco CleanAir and automatically focus/pan/zoom a security camera.

**Solution:** Use Network Automation based on Cisco IOS Embedded Event Manager to receive an SNMP Notification from WLC and trigger the Video Operations Manager via HTTP.

1. Rogue WLAN Device added
2. Rogue Device detected by CleanAir AP
3. WLC sends SNMP Notification
4. EEM triggers upon SNMP Notification
5. EEM notifies VSOM via HTTP
6. Security Camera Focus/Pan/Zoom
**Problem:** How to actively gather and share information from a router and from a few devices behind the router – across organizational and technical borders?

**Solution 1:** Initiate a project to make use of SNMP, Syslog, Event Management Software, Reporting, Provisioning and CRM Systems ...

**Solution 2:** Use Cisco IOS Network Automation to collect and post the information

Using Cisco IOS Embedded Event Manager and Tcl:

1. Import the http package into EEM policy
   ```bash
   namespace import ::http::*
   ```

2. Collect the information required

3. Build a query for the http POST operation
   ```bash
   set my_query [::http::formatQuery "status" $my_info]
   ```

4. POST the information to a website
   ```bash
   set my_reply [::http::geturl $my_server_url -query $my_query]
   ```
Collect and Share Remote Information – 2/2

Good Morning: Fan Running successfully. Thursday 20091210 at 07:45:01 UTC
about 2 hours ago from API
Pseudorandom factoid: SNMP sysobjectID is 1.3.1.19.1.642
about 18 hours ago from API
Pseudorandom factoid: Image file is flash.c181x-adviservicesk9-
mz.150-1.M.dpi
12:55 AM Dec 9th from API
Print Server is down. Saturday 20091205 at 21:24:51 UTC
1:20 PM Dec 5th from API
Print Server is now up and alive (Toner levels: B60% C90% M70% Y70%). Saturday 20091205 at 21:18:00 UTC
1:10 PM Dec 5th from API

See: http://twitter.com/EASyDMI
Note: it is NOT recommended to use a public site or feed other than for demo purpose
Embedded Event Manager (EEM)

Embedded Event Manager

Actions

EEM Applets
multi-event-correlation

Event Detectors

Syslog ED
SNMP EDs
Timer EDs
none ED
HW EDs
Watchdog ED
Interface Counter ED
XML RPC ED
CLI ED
OIR ED
ERM ED
EOT ED
RF ED
GOLD ED
NetFlow ED
IPSLA ED
Route ED
CDP LLDP ED
802.1x ED
MAC ED

Remote:
• Notification
Local:
• Notification
• Get/Set
• Cron
• Count
down
• Fan
• Temp
• Env
...

Syslog Event

Interface Descriptor Blocks

Process Scheduler Database

CDP
LLDP
802.1x
MAC
## Embedded Event Manager – Applet Evolutions

<table>
<thead>
<tr>
<th>EEM Version</th>
<th>Release</th>
<th>Applet Modifications</th>
<th>Peanut Gallery Comments</th>
</tr>
</thead>
</table>
| 1.0         | 12.0(26)S, 12.3(4)T | 2 Events: Syslog, SNMP  
Actions: Log, CNS event, Reload, Switchover | Initial Version  
Limited benefits                                           |
| 2.0         | 12.3(14)T1    | No structure changes  
Various New event detectors  
New actions: cli, info, mail, policy, SNMP trap, Modify counters, Publish  
application events,  
Read/set tracked objects | Many new actions  
Popular CLI / mail  
Actions run linear                                                   |
| 2.1         | 12.2(18)SXF5  |                                                                         |                                                                 |
| 2.2         | 12.4M         |                                                                         |                                                                 |
| 2.3         | 12.4(11)T     | maxrun support  
pattern parameter for CLI actions | Maxrun support == security  
Can handle CLI prompts                                               |
| 2.4         | 12.4(20)T     | multi-event support                                                               | Boolean correlation  
of events within applet.                                            |
| 3.0         | 12.4(22)T, 12.2(33)SE | Redesign of action mechanism  
Program counter added  
Loops, conditionals, regexps,  
context save, error handling | Applets now rock!  
Programming language feel.                                      |
| 3.1/3.2     | -             | No changes                                                                           | -                                                              |
| 4.0         | 15.2(2)T      | Applet file actions  
TLS/SSL support for SMTP actions  
Custom port for SMTP actions | File manipulation handy!                                           |
Example: EEM Applets – Loops, Variables

**Problem:** None in Particular

**Solution:** Have fun exploring EEM Applet capabilities

```plaintext
event manager applet 99-bob
description written by bklauser inspired by http://www.99-bottles-of-beer.net
event none
action 100 set b 99
action 110 while $b gt 1
  action 120 puts "$b bottles of beer on the wall, $b bottles of beer."
  action 130 decrement b
  action 140 puts "Take one down, pass it around, "
  action 150 puts "$b bottles of beer on the wall.\n"
action 160 end
  action 170 puts "$b bottle of beer on the wall, $b bottle of beer.
  action 180 puts "Take one down, pass it around, "
  action 190 puts "no more bottles of beer on the wall.\n"
  action 200 puts "No more bottles of beer on the wall, "
  action 210 puts "no more bottles of beer."
  action 220 puts "Go to the store and buy some more, "
  action 230 puts "99 bottles of beer on the wall.\n"
!
alias exec sing event manager run 99-bob
```

**See also:** http://www.99-bottles-of-beer.net/language-cisco-ios-embedded-event-manager-applet-2909.html
Packaging Network Automations

**Problem:** Cisco IOS Embedded Automation Systems often include multiple configuration items, files, checks and procedures – how to ensure they are deployed consistently?

**Solution:** Cisco EASy provides a simple packaging mechanism and open-source EASy Installer. A developer guide is available online to assist with the creation of EASy packages.

- Package Description
- Pre-Requisite Verification
- Pre-Installation Config
- Pre-Installation Exec
- Environment Variables
- Configuration
- Files
- Post-Requisite Verification
- Post-Installation Config
- Post-Installation Exec
- Uninstall

EASy Installer + MyPackage.tar = Menu Guided Installation

Router# easy-installer tftp://10.1.1.1/mypackage.tar flash:/easy

Configure and Install EASy Package 'mypackage-1.03'

1. Display Package Description
2. Configure Package Parameters
3. Deploy Package Policies
4. Exit

Enter option: 2

See: [http://www.cisco.com/go/easy](http://www.cisco.com/go/easy)
Embedded Automation Systems (EASy)

1. Browse and Download EASy Packages
   www.cisco.com/go/easy

2. Make Sure to also download EASy Installer

3. Browse Other Embedded Automations
   www.cisco.com/go/ciscobeyond

4. Learn About The Technology Under The Hood
   www.cisco.com/go/instrumentation
   www.cisco.com/go/eem
   www.cisco.com/go/pec

5. Discuss, Ask Questions, Suggest Answers
   supportforums.cisco.com
   supportforums.cisco.mobi

6. Upload your own Examples to CiscoBeyond
   www.cisco.com/go/ciscobeyond

7. Engage via ask-easy@cisco.com
Operational Network Automation

Business Operations

APIs and Agents

Domain Controllers

Virtual / Overlay Networks

Network

Survivability → Manageability → Automation

ICT Operations

1 3 5

6 7

c b a
Inflection: Business-Driven Network Automations ...
Inflection: Business-Driven Network Automations …

Business Operations

Value to Business

Type I
Automate Existing Task

Benefits:
- OPEX ↓
- Quality ↑

Type II
Automate New Task

Benefits:
- Revenue enabler
- OPEX ↓
- CAPEX ↓
- Quality ↑
- Reactive → Proactive

Type III – Automation as Integral Part of Solution Design

Benefits:
- Revenue enabler
- OPEX ↓
- CAPEX ↓
- Quality ↑
- Reactive → Proactive
- Corporate Learning enabler

Maturity and Innovation

Network
Survivability → Manageability → Automation
Self-Service for Human Users

1. Business Operations
2. Virtual / Overlay Networks
3. Wireless Controller(s) (WLC, etc, …)

ICT Operations

Network
Survivability → Manageability → Automation
Self-Service for Human Users

Network
Survivability → Manageability → Automation

Virtual / Overlay Networks

Wireless Controller(s) (WLC, etc, …)

Business Operations

Operations (FCAPS)

ICT Operations

© 2012 Cisco and/or its affiliates. All rights reserved.
Self-Service for Human Users

Business Operations

Wireless Controller(s) (WLC, etc, …)

Virtual / Overlay Networks

Resource Allocation

Operations (FCAPS)

ICT Operations

Network

Survivability ➔ Manageability ➔ Automation
Self-Service for Human Users

Network
   Survivability → Manageability → Automation

Business Operations

Virtual / Overlay Networks

Operations (FCAPS)

ICT Operations

Services (Location, Guestnet, Onboarding, …)

Resource Allocation

© 2012 Cisco and/or its affiliates. All rights reserved.
Self-Service for Human Users

Business Operations

Wireless Controller(s) (WLC, etc, …)

Operations (FCAPS)

ICT Operations

Overlay Networks

Resource Allocation

Survivability → Manageability → Automation

Virtual / Overlay Networks

Services (Location, Guestnet, Onboarding, …)

ICT Operations
Inflection: Network Programming

1. Survivability
2. Manageability
3. Automation
4. Autonomy

Business Operations

APIs and Agents

Domain Controllers

Virtual / Overlay Networks

ICT Operations

Network

Survivability → Manageability → Automation → Autonomy
Inflection: Network Programming

Virtual / Overlay Networks

What if the ‘User’ is a Software App?

Network
Survivability → Manageability → Automation → Autonomy

ICT Operations

1. APIs and Agents
2. Domain Controllers
3. APIs and Agents
4. Domain Controllers
5. Business Operations
6. Business Operations
7. Business Operations

© 2012 Cisco and/or its affiliates. All rights reserved.
Cisco ONE Platform Kit (onePK)

Your applications

Network Programming Environment to:
- Innovate
- Extend
- Automate
- Customize
- Enhance
- Modify

C, JAVA Program

API Presentation

API Infrastructure

Catalyst
Nexus
ASR ISR
Cisco ONE Platform Kit (onePK)

onePK Provides

Abstractions (Service Sets)

Programmatic Interfaces (C, Java, (REST) …)

Software Development Kit (SDK)

Anatomy of a onePK Application

Software Application (currently C and Java)

Interfaces and Abstractions (Service Sets)

Communication Bus (Thrift IDL)

Connected-Apps Agent in Network OS

Network OS Features and Embedded Automations

Service Set Description Version v.6.0.5 EFT 2012/2013

Data Path Provides packet delivery service to application: Copy, Punt, Inject

Policy Provides filtering (NBAR, ACL), classification (Class-maps, Policy-maps), actions (Marking, Policing, Queueing, Copy, Punt) and applying policies to interfaces on network elements

Routing Read RIB routes, add/remove routes, receive RIB notifications

Element Get element properties, CPU/memory statistics, network interfaces, element and interface events

Discovery L3 topology and local service discovery

Utility Syslog events notification, Path tracing capabilities (ingress/egress and interface stats, next-hop info, etc.)

Developer Debug capability, CLI extension which allows application to extend/integrate application’s CLIs with network element

Write once run anywhere
Network Programming
Cisco ONE Platform Kit (onePK)
onePK Service Sets – Element Properties – 1/2

- **System**
  - CPU, Memory, Platform, Serial #, Versions, Uptime, Location, OIR, CLI Changes

- **Interfaces**
  - Port, Slot, BW, MTU, TX/RX, BPS, PPS, Errors, Other Stats, Config, Link Changes

- **Discovery**
  - CDP, Topology Graph, Edges, Nodes, Topology Changes

YOUR Applications
onePK Service Sets – Element Properties – 2/2
Example: Connect and Get Properties via onePK (C)

```c
char *str = NULL;
onep_element_connect(elemA, user, pwd, NULL, &sh);
onep_element_get_property(elemA, &property);
if (property) {
    onep_element_to_string(elemA, &str);
    if (str) {
        fprintf(stderr, "\nElement Info: %s\n", str);
        free(str);
    }
}
```
Monitor Memory Usage

**Problem:** What if we need to dynamically investigate further upon a resource symptom?

**Solution:** Use the integration of EEM + ERM to trigger an EEM event when processor memory is greater than 80%.

```
resource policy
policy critmem global
  system
    memory processor
      critical rising 80
  interval 5
  user global critmem

event manager applet totmemcheck
  event resource policy critmem
  action 100 mail server "<server>" to "<to>" from "<from>" subject
  "Warning: proc memory spike"
```
A Network “Top”

Use onePK to build a live process monitor similar to UNIX *top*

The same app can connect to multiple devices to display the top processes across the entire network
onePK Service Sets – Policy and Routing – 1/2

RIB, Next-Hop, metric, AD, scope (VRF), Changes
Configured Classes
Configured ACLs
YOUR Applications
onePK Service Sets – Policy and Routing – 2/2

Routing

QoS

Security

Static routes

Service-Policies (Police, Mark, Shape, Queue)

ACLs

YOUR Applications
Example: Get and Set Routes via onePK (Java)

• Getting Routes

```java
L3UnicastScope scope = new L3UnicastScope("", AFIType.IPV4, SAFIType.UNICAST, ";");
NetworkPrefix prefix = new NetworkPrefix(InetAddress.getByName("0.0.0.0"), 0);
L3UnicastRIBFilter ribFilter = new L3UnicastRIBFilter(OwnerType.NONE, "NONE", prefix);
L3UnicastRouteRange range = new L3UnicastRouteRange(prefix, RouteRange.RangeType.EQUAL_OR_LARGER, 100);
List<TopoNode> mynodes = TopoNode.getAllNodes();
for(TopoNode thisnode : mynodes) {
    Routing routing = Routing.getInstance(thisnode.ne);
    RIB rib = routing.getRib();
    List<Route> routeList = rib.getRouteList(scope, ribFilter, range);
    for (Route route : routeList) {
    }
```

• Setting Routes

```java
L3UnicastRoute aRoute = new L3UnicastRoute(prefix, nextHopList);
aRoute.setAdminDistance(1);
RouteOperation op = new L3UnicastRouteOperation(RouteOperationType.ADD, aRoute);
List<RouteOperation> opList = new ArrayList<RouteOperation>();
opList.add(op);
AppRouteTable art = routing.getAppRouteTable();
art.updateRoutes(scope, opList);
```
Example: Routing for Dollars / CO$_2$ / Tulips /…

Setup

EIGRP
Routing Topology
No External Metrics
No External Algorithm
Example: Routing for Dollars / CO$_2$ / Tulips /…

Application Routes

- EIGRP
- onePK
- External Metrics
- External Algorithm
Example: Routing for Dollars / CO₂ / Tulips / ...

```
routing ospf 1
redistribute application <app name> ...
```
Example: Routing for Dollars / CO$_2$ / Tulips /…

Statistics and Metrics

- **Code Metrics**
  - Total lines of code: 4700 (JAVA)
  - 40% SWING GUI
  - 20% Dijkstra’s algorithm, lowest cost path determination
  - 25% Housekeeping: Node and link database
  - 15% Calls to onePK infrastructure + error checking

- **Code increase to add “Latency based routing” on top of “Routing for Dollars”**
  - 100 lines of code

- Modular code base written in Java has allowed us to port this to mobility client.

  Framework makes it easy to modify code and change business logic.

  Modular java code makes it easy to deploy on multiple clients.
Example: Routing for Dollars / CO₂ / Tulips /…

Recent Extensions

Path determination based on lowest latency

Latency information fed into app through IPSLA

Port to mobility client
onePK Service Sets – Data Path – 1/2
onePK Service Sets – Data Path – 2/2

Data Plane

Inject New or Modified Packets

YOUR Applications
Example: Punt and Inject Packets via onePK (C)

```
TRY(rc, onep_dpss_register_for_packets(
    nel,
    dpss,
    targ_left,
    interesting_class,
    ONEP_DPSS_ACTION_PUNT,
    encrypt_callback,
    (void *)intf_left,
    &reg_handle), "Register for packets");
```
Custom Application Traffic Flow Handling – 1/2

**Problem:** We need to custom encrypt packets of a specific application traffic flow

**Solution:** Use onePK to punt, encrypt and reinject the relevant packets

1. Policy APIs on ingress router are set to punt telnet and syslog to app
2. App encrypts punted traffic and re-injects into data path.
3. Policy APIs on egress router punt telnet and syslog to app
4. App decrypts punted traffic and re-injects into data path.
5. Traffic that does not match policy passes through unencrypted.
Custom Application Traffic Flow Handling – 2/2

What Client Sees

```
client#telnet 10.13.1.1
Trying 10.13.1.1 ... Open

User Access Verification
Username: user1
Password: 
server>en
Password: 
server#show clock
*10:02:12.131 PST Mon Jul 2 2012
```

What Wireshark Sees

```
Stream Content

User Access Verification
Username: ...!.....user1
Password: 
server>en
Password: 
server#show clock
*10:02:12.131 PST Mon Jul 2 2012
server#!Starting Application Now
server#show clock
*10:02:42.169 PST Mon Jul 2 2012
```

```
saved by %9^v^9!v5:95= |gfldlbdxg\ov...v.98v.#:vdvdgfd
%3$ 3$u
```
onePK Agent ↔ Application Interactions

Examples:
- Getting 50'000 ACLs from an Element
- Syslog Messages
- RIB Changes
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Getting 50'000 ACLs from an Element
- Syslog Messages
- RIB Changes

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address

Examples:
- Setting 50'000 ACLs on an Element
- Get Element Version
- Set Interface Address
What will YOU program?
### Portfolio of API, Languages and Abstractions

**Network Programming with onePK and Embedded Network Automation**

<table>
<thead>
<tr>
<th>Native Network OS Embedded Automation</th>
<th>Advanced Network OS Embedded Scripting</th>
<th>Structured API</th>
<th>Object Oriented API</th>
<th>Higher-Level Abstractions / Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event-/Expression- MIB, PfR, IPSLA Thresholds, Embedded Event Manager Applets, …</td>
<td>Tcl, Python, Embedded Event Manager, EASy, …</td>
<td>onePK C</td>
<td>onePK Java</td>
<td>onePK Libraries REST, XMPP, Design Patterns, OMNI Controllers, …</td>
</tr>
</tbody>
</table>

---

**Choice and Flexibility of Implementation**

Network Programming – SDN

Network Automation – Embedded Automations
Cisco ONE – Current Key Components …

Network
Survivability → Manageability → Automation → Autonomy

Business Operations

APIs and Agents

ONE Controller
Open Daylight Controller

Domain Controllers

Virtual / Overlay Networks

Nexus 1000v  CSR 1000v  ASA 1000v  VSG  vWAAS  ++

APIs and Agents

CloudConnectors

UCS-E  Virtual Containers

For Your Reference

Cisco ONE – Current Key Components …

Network
Survivability → Manageability → Automation → Autonomy

Business Operations

APIs and Agents

ONE Controller
Open Daylight Controller

Domain Controllers

Virtual / Overlay Networks

Nexus 1000v  CSR 1000v  ASA 1000v  VSG  vWAAS  ++

APIs and Agents

CloudConnectors

UCS-E  Virtual Containers
Resources and References
References – Programmable and Cloud-Intelligent

Cisco Scripting Community: [www.cisco.com/go/ciscobeyond](http://www.cisco.com/go/ciscobeyond)
Network Programming Hands-On Lab

At CiscoLive: Join us for TECNMS-3601 Advanced Network Automation or

1. Walk up to the WISP Labs
2. Book a seat for:
   - Network Automation – The Basics
   - Network Automation – Intermediate
   - Network Automation – Advanced
   - Network Automation – Smart Call Home
   - Network Programming – The Basics
   - Network Programming – Intermediate
   - Network Programming – Advanced

Partners via PEC (currently being updated):

1. Navigate to http://www.cisco.com/go/pec
2. Click on >Launch
3. Search for ‘%Network Programming%’
4. Enjoy !
Network Automation Hands-On Lab – 3 Levels

2 – Network Automation: Intermediate
Service Planning
Task 211 – Exporting Bulk Statistics
Task 222 – Exporting upon Periodical Time Windows
Service Deployment
Task 221 – Installing an EASY Package – Extensible HTTP Server
Task 222 – Creating an EASY Package – Hello World
Task 223 – Configure Your Network for a onePK Application
Task 224 – EASY Post-Deployment Manageability Verification
Service Monitoring
Task 231 – Monitoring Resources – Using ERM and EEM
Task 232 – Monitor Application Traffic using Flexible NetFlow - NBAR Integration
TBC Task 233 – Monitoring SLA – Using Cisco IOS Auto IPSLA (+ IPSLA ED)
Service Troubleshooting
Task 241 – Detecting Low-TTL Traffic
Task 242 – Capturing Packets – Using Embedded Packet Capture
Task 243 – Capturing Packets – Using the EASY EPC Package
Task 244 – Logging SNMP Traps – Using the EASY Trap Logger

1 – Network Automation: The Basics
Task 101 – Working with EEM Applets – Hello World
Task 102 – Working with EEM Tcl Policies – Hello World
Task 103 – Working with EEM – CLI Event Detector
Task 104 – Working with EEM – Regular Expressions
Task 105 – Simple Command Logging – EEM CLI Event Detector
Task 106 – Custom Interface Failover – using EEM, IPSLA, and Enhanced Object Tracking (EOT)
Task 107 – Editing ASCII Files in IOS
Task 108 – Using the IOS HTTP Client
Task 199 – Sneak Peek: onePK Custom Routing Application

3 – Network Automation: Advanced
Service Planning
Task 311 – EEM System Policies – Tune to Device’s EEM Capacity
Service Deployment
Task 321 – Using Tcl Libraries in EEM Policies – Pure Tcl
Task 322 – Dynamically Configure from HTTP Input – Using a Pure Tcl Library
Task 323 – Protect Your Intellectual Property – Signed Byte Code
Service Monitoring
Task 331 – Send Syslog Upon Route Change – EEM Routing Event Detector
Task 332 – Monitoring Resource Groups – Using ERM and EEM
Task 333 – Collecting Reliable Metrics – Using DCM, IP SLA and EEM
Service Troubleshooting
Task 341 – EEM Applet – Testing and Debugging EEM Policies
Network Programming Hands-On Lab – 3 Levels

1 – Network Programming: The Basics
   Task 511 – Sample App: Deploy and Run a onePK Application from Eclipse
   Task 402 – Sample App: Routing for Dollars – Beyond Normal Routing Metrics
   Task 411 – Compile and Run a onePK C Application using make
   Task 412 – Compile and Run a onePK Java Application using mvn
   Task 421 – Create your own C onePK application
   Task 422 – Create your own Java onePK application

2 – Network Programming: Intermediate
   Task 521 – Application Interaction with Custom CLI
   Task 531 – Application Interaction with Embedded Event Manager

3 – Network Programming: Advanced
   Task 491 – Simulating a Network – IOL and NETMAPs
   Task 492 – Connecting to a Simulated Network – IOLTunnel
   Task 611 – Introspection: Discover Service Sets supported
Master Class Network Programming and Automation
Sustainable Innovation and Differentiation with Cisco Embedded Automation and onePK

How can I successfully use Network Programming?
Step-by-step introduction and adoption strategies

Where do I start – can I do this?

Practical Examples and Hands-on Lab

What have other Customers and Partners done?
Real-Life Examples and Case Studies

What are Cisco ONE and onePK anyhow?

Illustration in the bigger SDN and Automation Context

Join us for the Network Programming Master Class Series
References – Instrumentation and Automation

Device Manageability Instrumentation (DMI) [www.cisco.com/go/instrumentation]

- Embedded Event Manager (EEM): [www.cisco.com/go/eem]
- Cisco Beyond – EEM Community: [www.cisco.com/go/ciscobeyond]
- Embedded Packet Capture (EPC): [www.cisco.com/go/epc]
- Flexible NetFlow: [www.cisco.com/go/netflow] and [www.cisco.com/go/fnf]
- IPSLA (formerly SAA, formerly RTR): [www.cisco.com/go/ipsla]
- Network Analysis Module: [http://www.cisco.com/go/nam]
- Network Based Application Recognition (NBAR): [www.cisco.com/go/nbar]
- Smart Call Home: [www.cisco.com/go/smartcall]
- Cisco Configuration Engine (CCE): [www.cisco.com/go/ciscoce]

- Feature Navigator: [www.cisco.com/go/fn]
- MIB Locator: [www.cisco.com/go/mibs]
Network Programming and Automation
Sustainable Innovation and Differentiation with Cisco Embedded Automation and onePK

Network Programming in a Cisco Open Networking Environment …
… provides Choice and Flexibility of
  - APIs and Abstractions
  - Architectures
  - Deployment Models
… closes the gap between Business Applications and Networks
… enables Operational Savings and New Opportunities
… puts YOU in control

What will YOU Program?
Prosíme, ohodnoťte tuto přednášku.
Děkujeme za pozornost.