Multi-Layer Access Control for SDN-based Telco Clouds

(result of a joint CELTIC research project called SASER – SAve and SEcure Routing)

Bernd Jaeger[1], Christian Röpke[2], Iris Adam[1], Thorsten Holz[2]

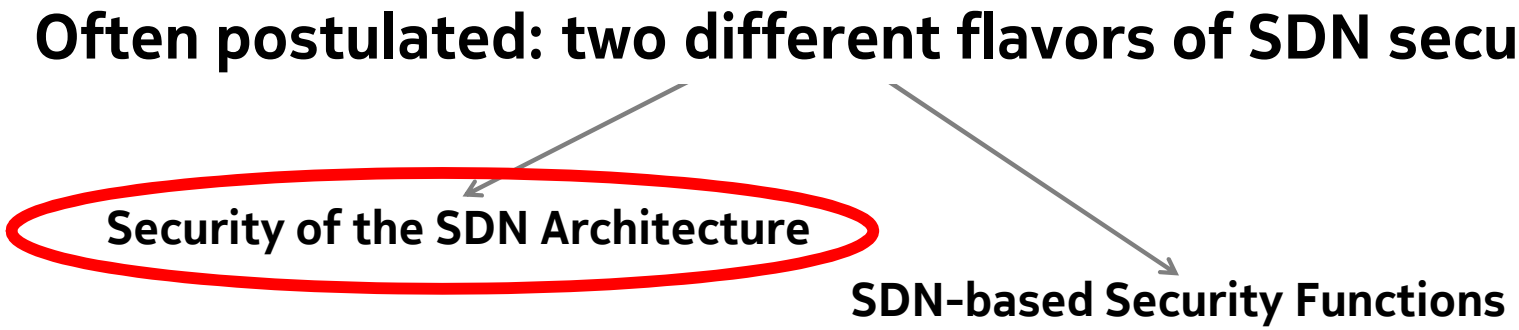
1: Nokia Networks

2: Ruhr-University Bochum

NOKIA

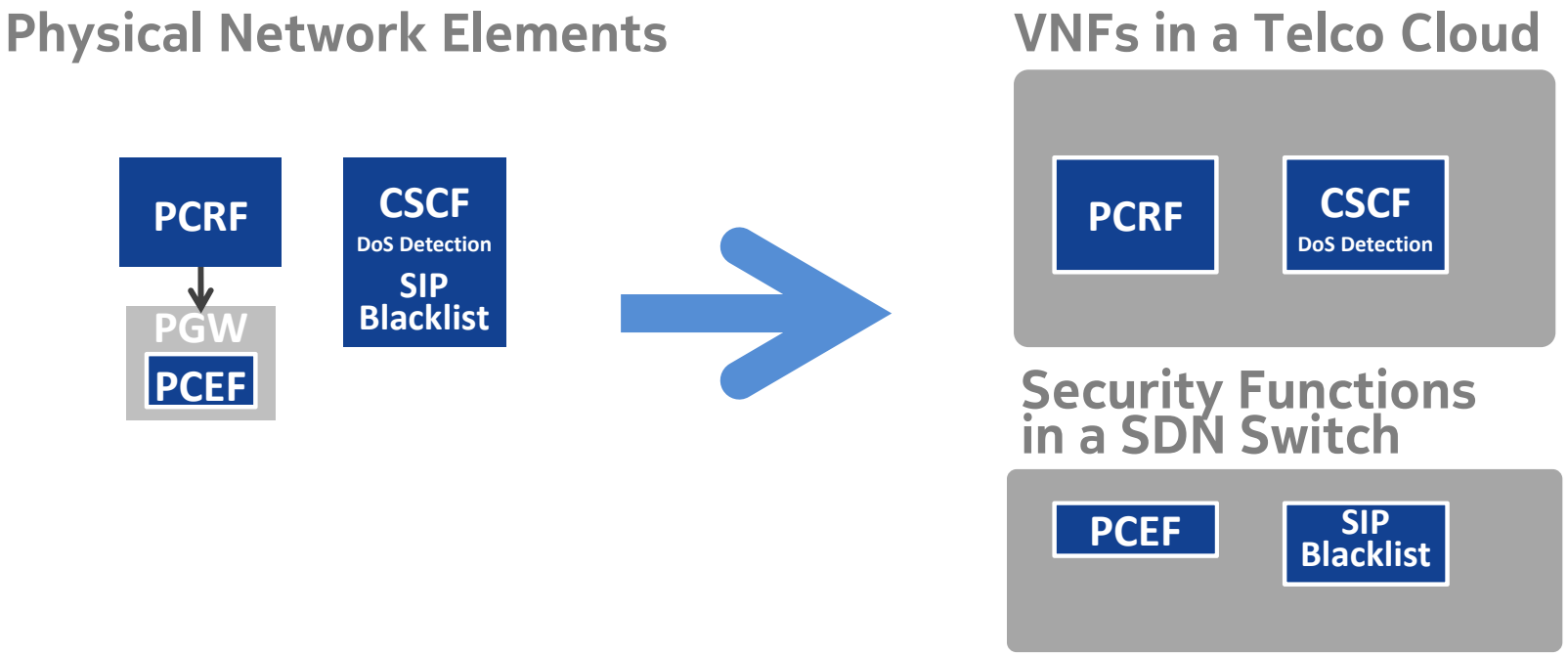
hgi Horst Görtz Institut für IT-Sicherheit

SDN Security

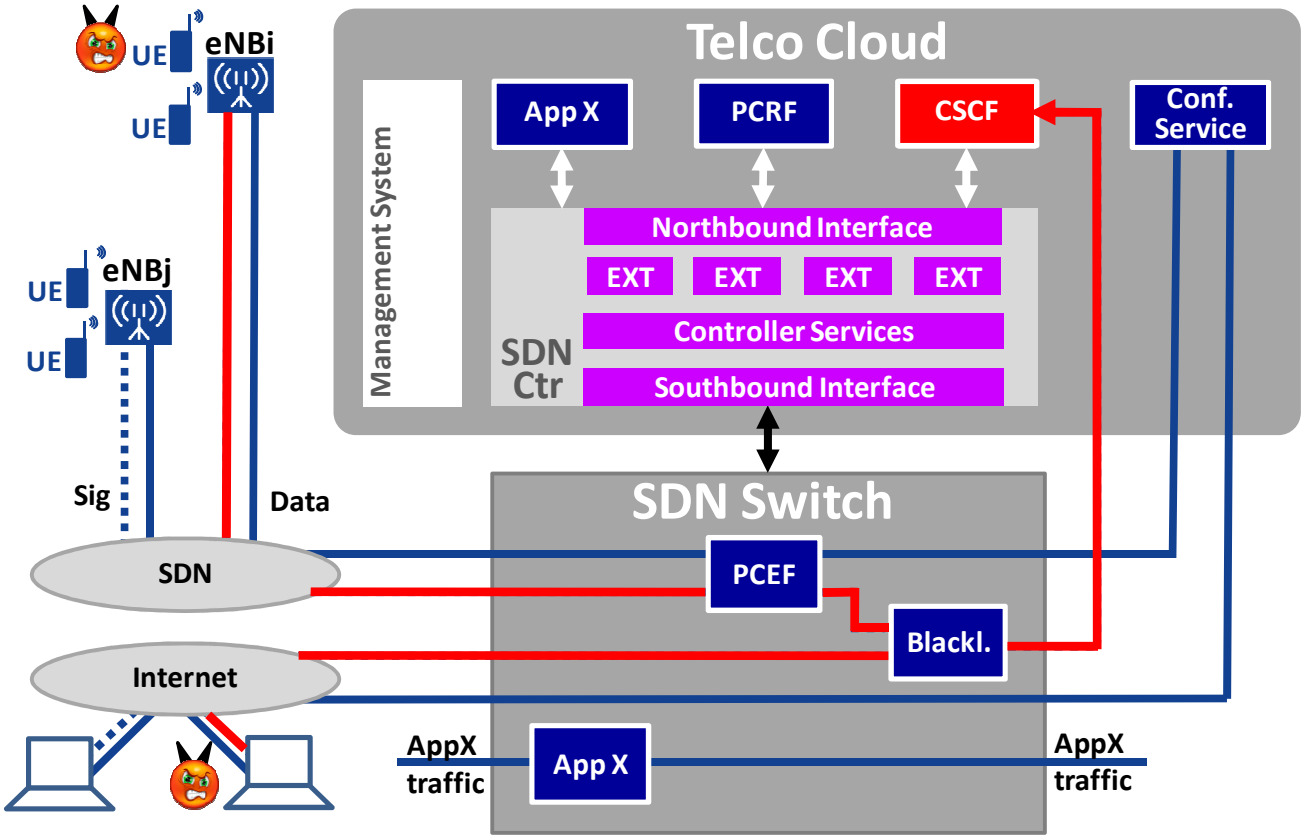


- The analysis combines both aspects at the example of a SIP signaling SDN network with focus on the security of the SDN architecture
- Assumed are multiple applications on top of a single SDN controller, controlling a number of (partly) chained security functions in a SDN switch
- The applications and the SDN controller are assumed to run in a telco cloud with the worst-case threat that an application gets compromised and then acts maliciously

Split of Physical Network Elements into VNFs and SDN-based Security Functions



Simplified Example: Mobile-Internet Conference



UE - User Equipment
eNB – eNodeB (base station)
Sig – Signaling

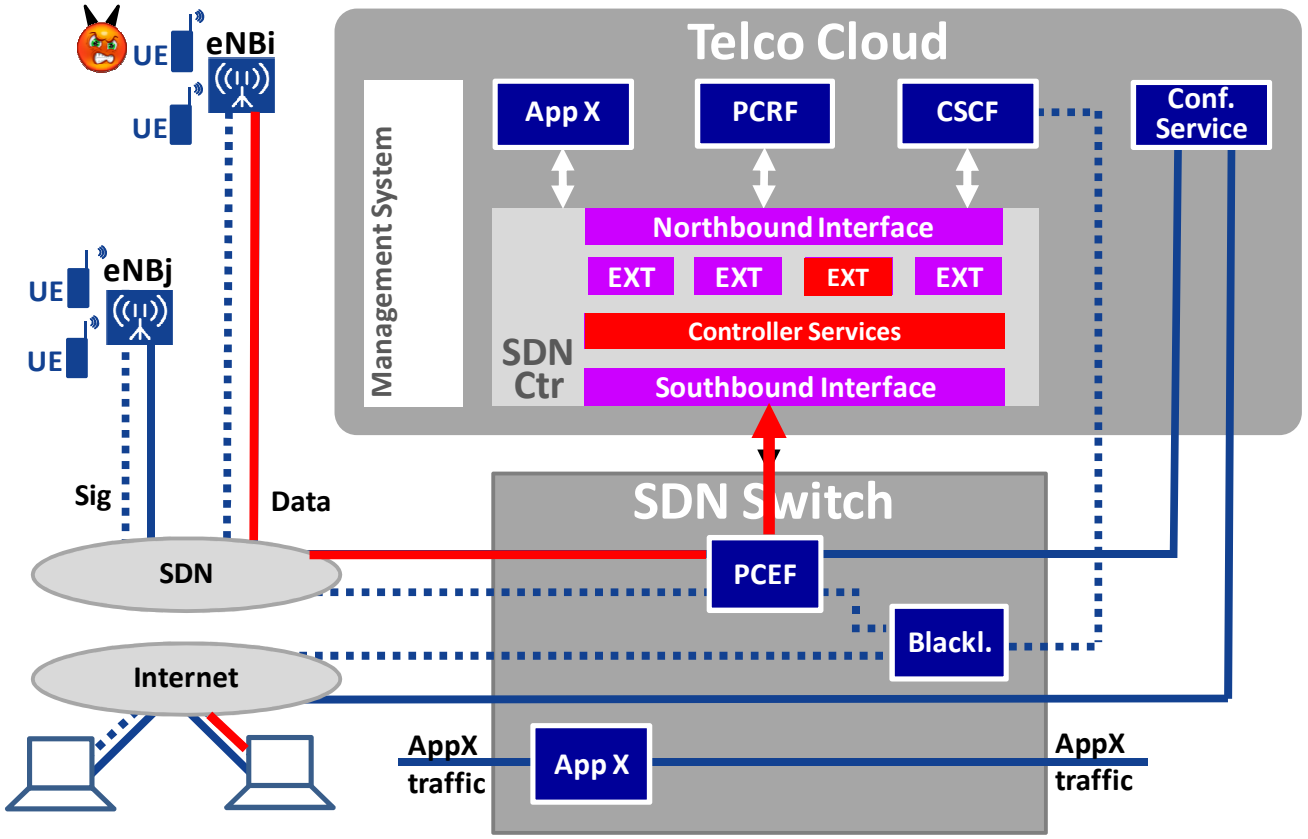
PCRF – Policy & Charging Rules Function
PCEF – Policy & Charging Enforcement Function
CSCF – Call Session Control Function

SDN Ctr – SDN Controller
EXT - Extension
Blackl. - Blacklist

Attacker Model

- **Malicious end user systems exploit vulnerabilities in cloud applications**
- Attackers exploit SDN controller vulnerabilities by sending specially crafted packets to an SDN switch triggering it to delegate these packets to the SDN controller
- Cloud applications respectively SDN controller extensions – intentionally or unintentionally – attack and infect other cloud applications or SDN controller components

Simplified Example: Mobile-Internet Conference



UE - User Equipment
eNB – eNodeB (base station)
Sig – Signaling

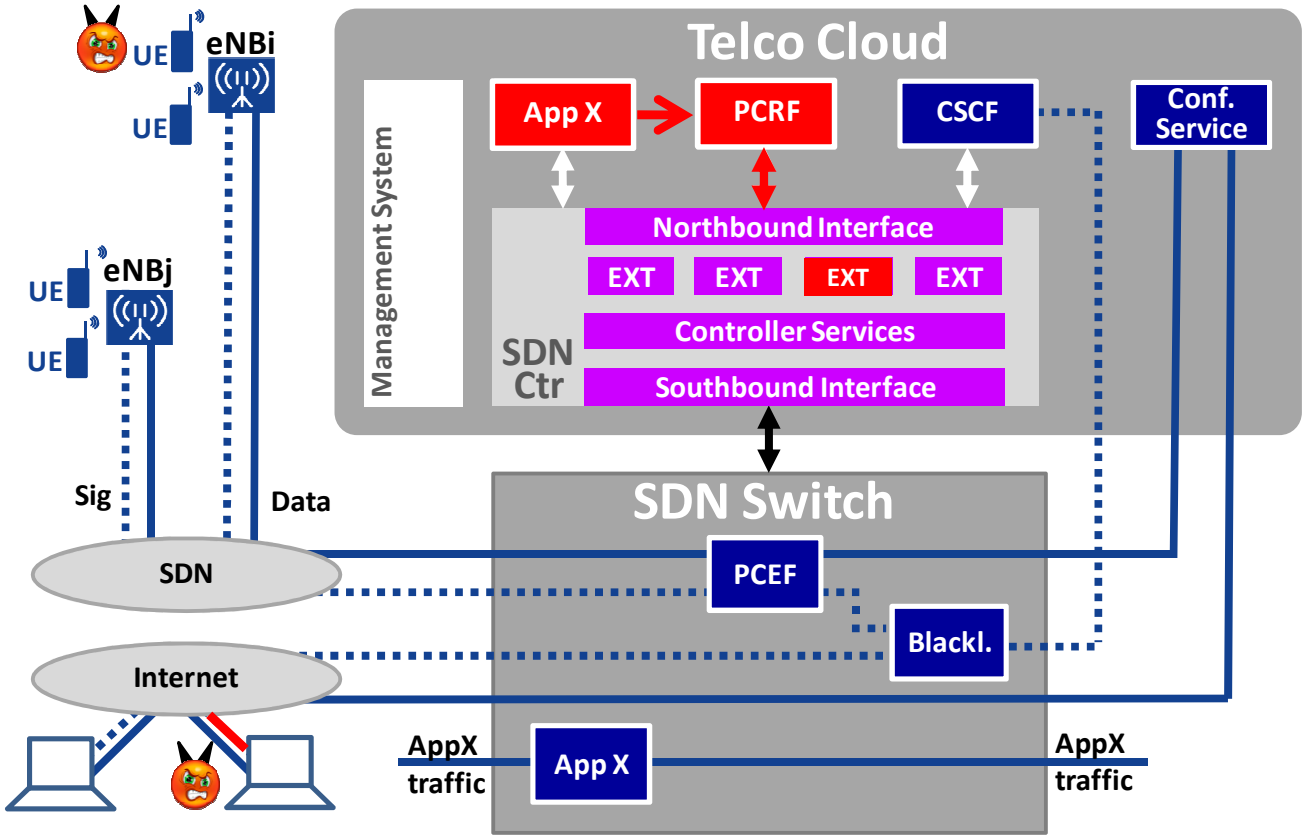
PCRF – Policy & Charging Rules Function
PCEF – Policy & Charging Enforcement Function
CSCF – Call Session Control Function

SDN Ctr – SDN Controller
EXT - Extension
Blackl. - Blacklist

Attacker Model

- Malicious end user systems exploit vulnerabilities in cloud applications
- **Attackers exploit SDN controller vulnerabilities by sending specially crafted packets to an SDN switch triggering it to delegate these packets to the SDN controller**
- Cloud applications respectively SDN controller extensions – intentionally or unintentionally – attack and infect other cloud applications or SDN controller components

Simplified Example: Mobile-Internet Conference



UE - User Equipment
eNB – eNodeB (base station)
Sig – Signaling

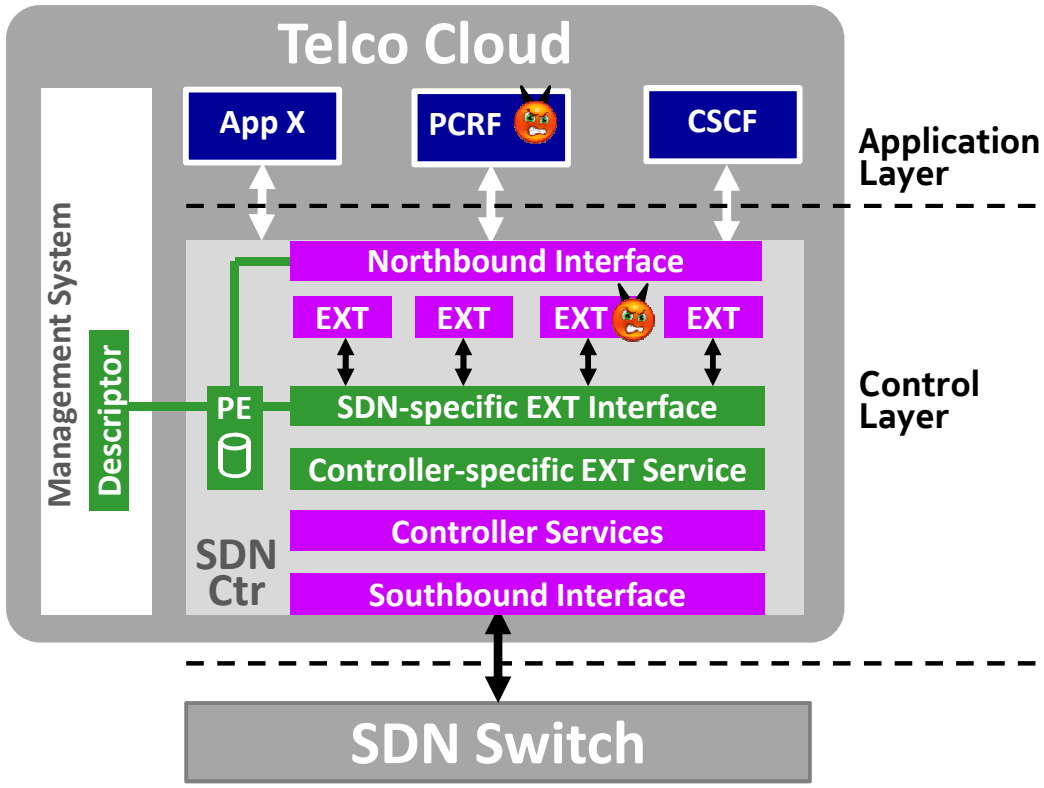
PCRF – Policy & Charging Rules Function
PCEF – Policy & Charging Enforcement Function
CSCF – Call Session Control Function

SDN Ctr – SDN Controller
EXT - Extension
Blackl. - Blacklist

Attacker Model

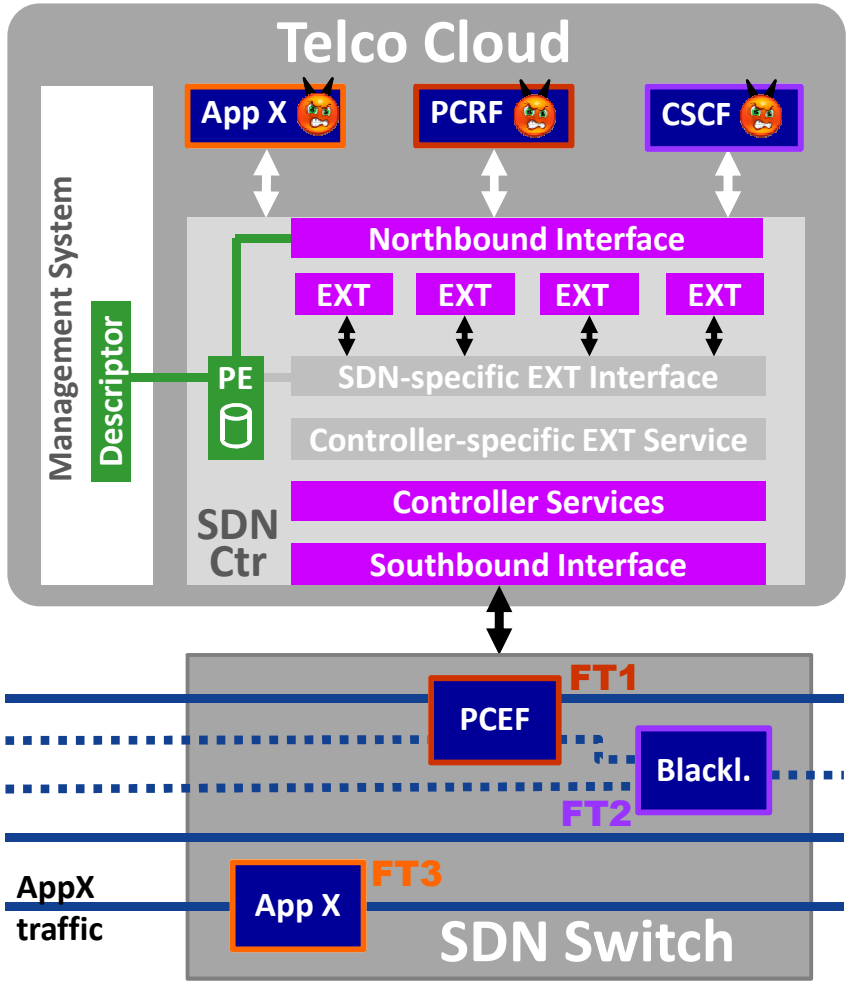
- Malicious end user systems exploit vulnerabilities in cloud applications
- Attackers exploit SDN controller vulnerabilities by sending specially crafted packets to an SDN switch triggering it to delegate these packets to the SDN controller
- **Cloud applications respectively SDN controller extensions – intentionally or unintentionally – attack and infect other cloud applications or SDN controller components**

Multi-Layer Access Control



- A Policy Enforcement (PE) unit provides protection against malicious behavior of northbound applications and SDN controller extensions, provided by means of a descriptor from an independent management system
- On Application Layer the Policy Enforcement unit restricts the allowed instruction set according to an application profile
- On Control Layer the allowed instruction set of SDN controller extensions is reduced by high-level permissions in an SDN controller independent fashion
- SDN controller independence can be achieved by providing an additional layer that adapts the high-level permissions to the respective SDN controller specifics

Assigning Applications to Forwarding Tables



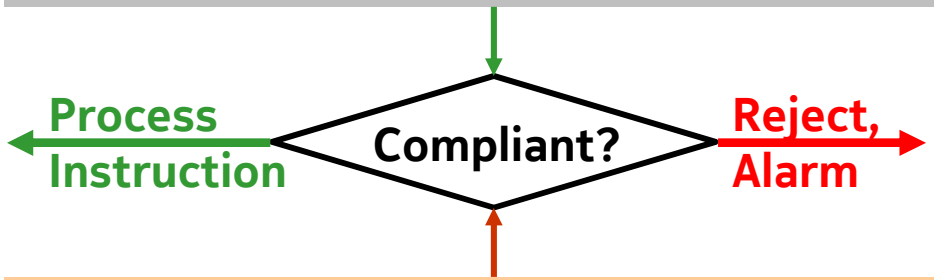
- A first step to increase security is to separate the flow rules of the respective applications into separate Forwarding Tables (FT)
- With that Forwarding Tables are decoupled unless they work on the same traffic stream. If so, they are still able to affect each other.
- But even if the Forwarding Tables are completely decoupled from each other, they are still not protected against attacks because each of the applications in the Telco Cloud can be potentially compromised and then send malicious instructions to the SDN controller
- This may e.g. result in DoS attacks (drop *.*), in illegitimate service consumption attacks, in manipulation of traffic integrity or in eavesdropping attacks by copying the traffic to an unauthorized destination

Working of Policy Enforcement - 1

Descriptor

Descriptor: Application Profile; Method Multiple Pipelined Forwarding Tables

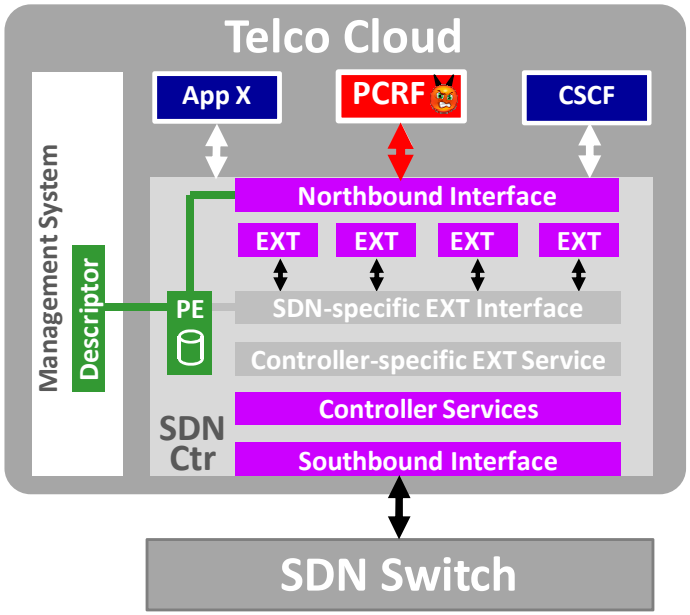
Source Address	Prot.	Prio	Action	Dest Addr.
Application: PCRF, SE-Threshold=100; assign Forwarding Table 1 (FT1)				
Single Element of user address range	SIP/Voice	Prio2	drop	-
All Elements of user address range	SIP	Prio1	goto FT2	-
All Elements of user address range	Voice	Prio1	forward	Conf. Server
Application: CSCF, SE-Threshold=100; assign Forwarding Table 2 (FT2)				
Single Element of user address range	SIP	Prio2	drop+error	-
All Elements of user address range	SIP	Prio1	forward	CSCF



Forw. Table 1

Forwarding Table 1 (FT1) NBI Commands

Source Address	Prot.	Prio	Action	Dest Addr.
Anton of user address range	SIP/Voice	Prio2	drop	-
Bernhard of user address rang	SIP/Voice	Prio2	drop	-
...				
Zora of user address range	SIP/Voice	Prio2	drop	-
All users of user address range	SIP	Prio1	goto FT2	-
All users of user address range	Voice	Prio1	forward	Conf. Server



NBI Instruction:
All users of user address range Voice Prio1 **drop**

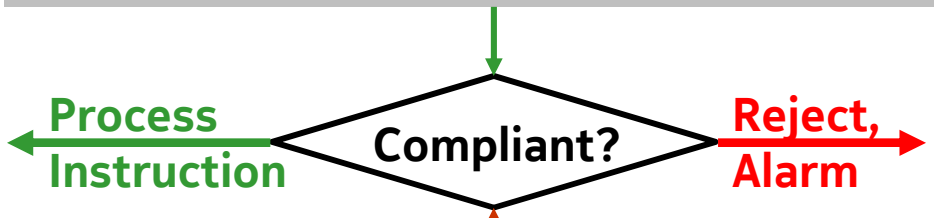
Reject, because 'drop' not allowed !

Working of Policy Enforcement - 2

Descriptor

Descriptor: Application Profile; Method Multiple Pipelined Forwarding Tables

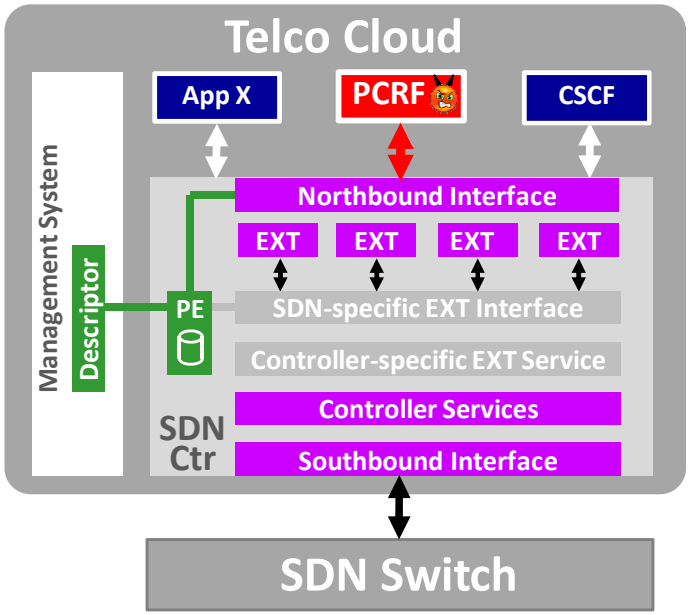
Source Address	Prot.	Prio	Action	Dest Addr.
Application: PCRF, SE-Threshold=100; assign Forwarding Table 1 (FT1)				
Single Element of user address range	SIP/Voice	Prio2	drop	-
All Elements of user address range	SIP	Prio1	goto FT2	-
All Elements of user address range	Voice	Prio1	forward	Conf. Server
Application: CSCF, SE-Threshold=100; assign Forwarding Table 2 (FT2)				
Single Element of user address range	SIP	Prio2	drop+error	-
All Elements of user address range	SIP	Prio1	forward	CSCF



Forw. Table 1

Forwarding Table 1 (FT1) NBI Commands

Source Address	Prot.	Prio	Action	Dest Addr.
Anton of user address range	SIP/Voice	Prio2	drop	-
Bernhard of user address rang	SIP/Voice	Prio2	drop	-
...				
Zora of user address range	SIP/Voice	Prio2	drop	-
All users of user address range	SIP	Prio1	goto FT2	-
All users of user address range	Voice	Prio1	forward	Conf. Server



NBI Instruction:

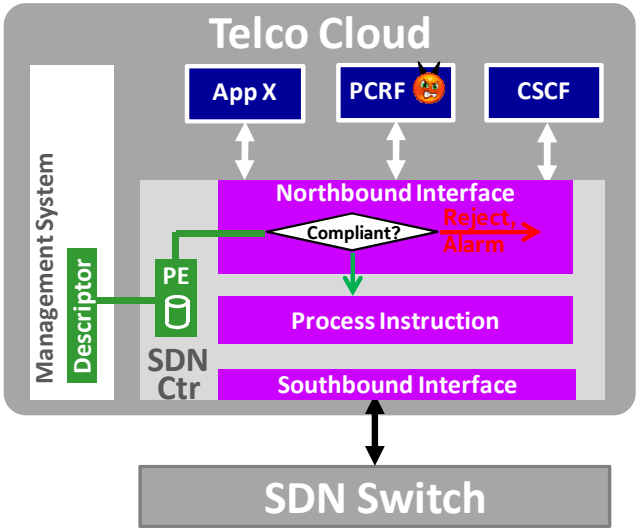
User 1 of user address range	SIP	Prio2	drop	Allow
User 2 of user address range	SIP	Prio2	drop	
.....				
User 100 of user address range	SIP	Prio2	drop	
---------- Threshold ----------				
User 101 of user address range	SIP	Prio2	drop	Reject, Alarm
---------- Alarm! ----------				

Applicability of Policy Enforcement

A Descriptor can make use of all Forwarding Table entries such as described below:

Match Fields	Priority	Counters	Instructions	Timeouts

- Match Fields: to match against packets (e.g. ingress port, source address, destination address, protocol, ports, optionally metadata specified by a previous Forwarding Table)
- Priority: matching precedence of flow entry
- Counters: updated when packets match
- Instructions: to modify the action set or pipeline processing
- Timeout: maximum time before flow is expired by the switch



- An application profile provided by means of a descriptor can improve resistance against malicious instructions in case of compromised applications significantly
- The presented method can help against all attacks that are verified during Forwarding Table matching
- The presented method can not prevent against attacks in the ‚normal fucntion spectrum' of an application (e.g. block a single user that has paid his invoice and not exceeded his data volume)
- This method is most appropriate for networks with a specified topology and well defined application functionalities, which applies to telco networks

Challenges On Control Layer

(i) Current sandbox systems for SDN controllers depend on low-level permissions

- Examples
  - System call level permissions: **sys_execve, sys_init_module, …**
  - Java API level permissions: **java.lang.RuntimePermission "accessDeclaredMembers", …**
- ➢ Difficult to understand for SDN controller operators
- ➢ 50-80% of network outages are caused by human errors
  - ➢ Sandbox misconfiguration is likely to happen in practice
  - ➢ Operators may simply grant all permissions to avoid malfunctioning

(ii) Current sandbox systems are controller-specific

- Sandbox system must be implemented for each SDN controller separately
- Interfaces and use will likely differ for each implementation
- ➢ Would not one access control system for all SDN controllers be better?

High-Level Permissions

Basic set of high-level permissions:

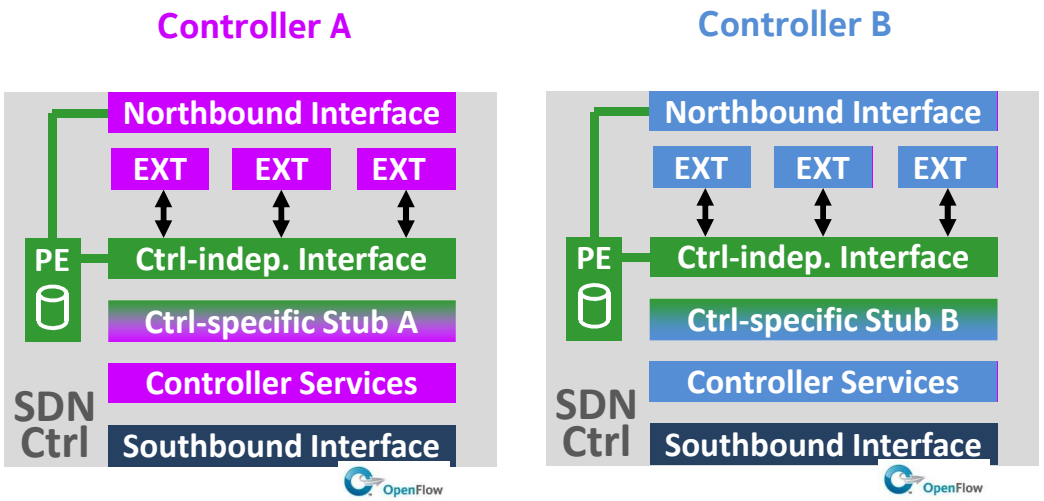
- readTopology, readStatistics
- addForwardingTableEntry, delForwardingTableEntry
- recvDataPkt, sendDataPkt

- Easy to understand by SDN controller operators
  - ➢ Only necessary permissions are granted for a SDN controller extension
    - ➢ Malicious or vulnerable components are restricted to a minimum set of critical operations

Example:

- A load balancer extension may need
  - recvDataPkt, readTopology, readStatistics, add/delForwardingTableEntry
    - ➢ With these permissions, it could rule the network
- We therefore need further restrictions, i.e., on switch and forwarding entry level:
  - Load balancer is only allowed to access, e.g., switches 1, 2, 6 and 12
  - Load balancer is only allowed to program forwarding entries on these switches, e.g., which are associated with the load balancer‘s backend server

Controller-Independent Access Control

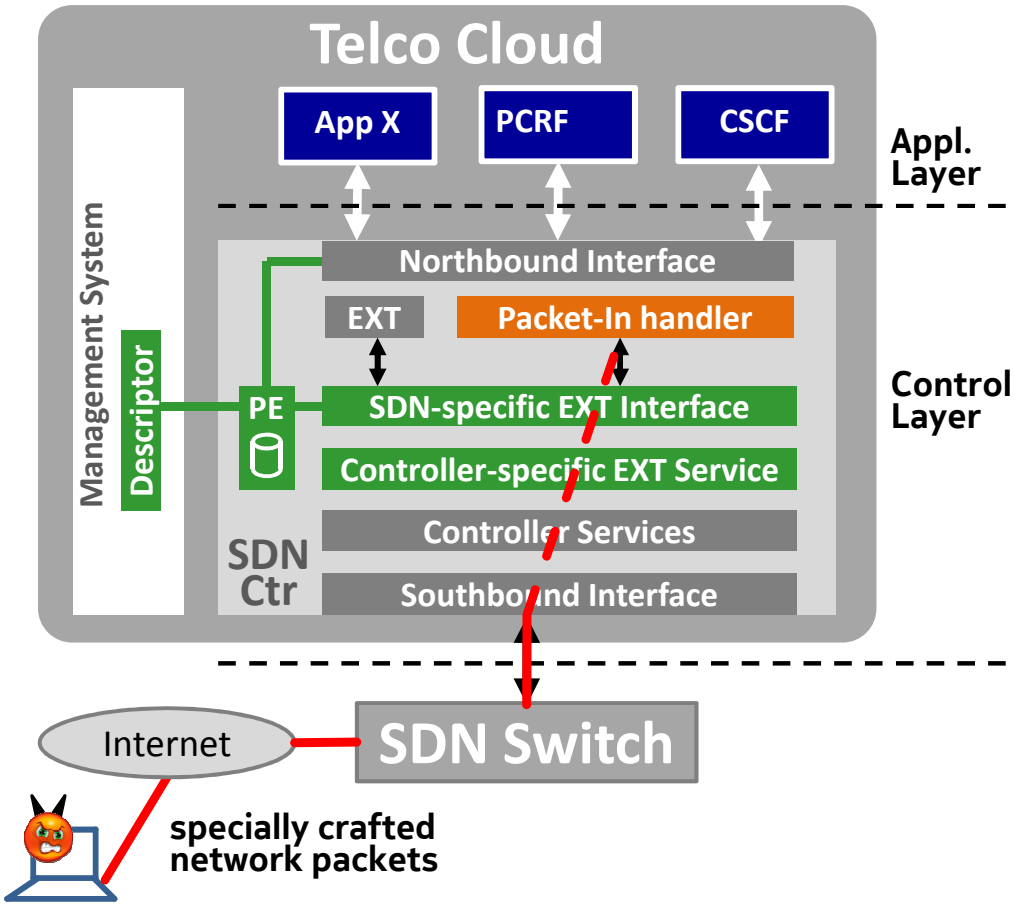


- We add an additional layer providing a controller-independent interface to high-level and SDN-specific controller services
- Interface is implemented by a controller-specific service using controller services
- Example:
  - Controller A implements high-level operations by its specific services
  - Controller B implements the interface layer by services of this controller

➢ This allows access control for a wide range of SDN controllers

➢ Extensions must not be implemented for each SDN controller separately

Control Layer Protection



Scenario

- A packet-in handler extension is vulnerable
- Specially crafted packets trigger a switch to delegate them towards this extension

Access Control Functioning:

- When exploiting this vulnerability, an attacker may be able to perform critical operations:
  - Add forwarding entries allowing the attacker to connect to internal servers.
  - Redirect internal traffic to the attacker‘s IP
- Our system mitigates such control layer attacks:
  - Attacker can only perform a reduced set of critical operations
  - With it, an attacker can only manipulate a reduced set of switches
  - On such switches, an attacker can only manipulate certain forwarding table entries

Conclusions

- Telecom providers started building telco clouds based on the emerging technologies of NFV and SDN
- Cloud-based functions such as telco applications and SDN controllers are potentially vulnerable and may easier be compromised because they are logically and not physically separated
- Therefore leaving cloud applications and SDN controller extensions with unlimited access to critical operations can result in the adverse misuse of such operations
- As a consequence, SDN end user traffic can be manipulated, copied or dropped
- We improve the security of such SDN end user traffic by restricting access to corresponding and harmful operations on both, the application and the control layer
- With our improvements, telco cloud providers are able to mitigate various attack scenarios on multiple layers

Contact

Nokia, Munich:

bernd.jaeger@nokia.com (Application Layer Access Control, main contact)
iris.adam@nokia.com (Application Layer Access Control)

Ruhr University Bochum:

christian.roepke@rub.de (Control Layer Access Control, main contact)
thorsten.holz@rub.de (Control Layer Access Control)

Back-Up Slides

ETSI NFV Reference Architecture