I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

A Formal Access Control Model for SE-Floodlight Controller

Abdullah Al-Alaj[1], Ravi Sandhu[1] and Ram Krishnan[2]

[1]Dept. of Computer Science
[2]Dept. of Electrical and Computer Engineering
[1,2]Institute for Cyber Security
[1,2]Center for Security and Privacy Enhanced Cloud Computing (C-SPECC)
University of Texas at San Antonio, TX 78249

SDN-NFV Security 2019
Dallas, Texas, USA, March 27, 2019

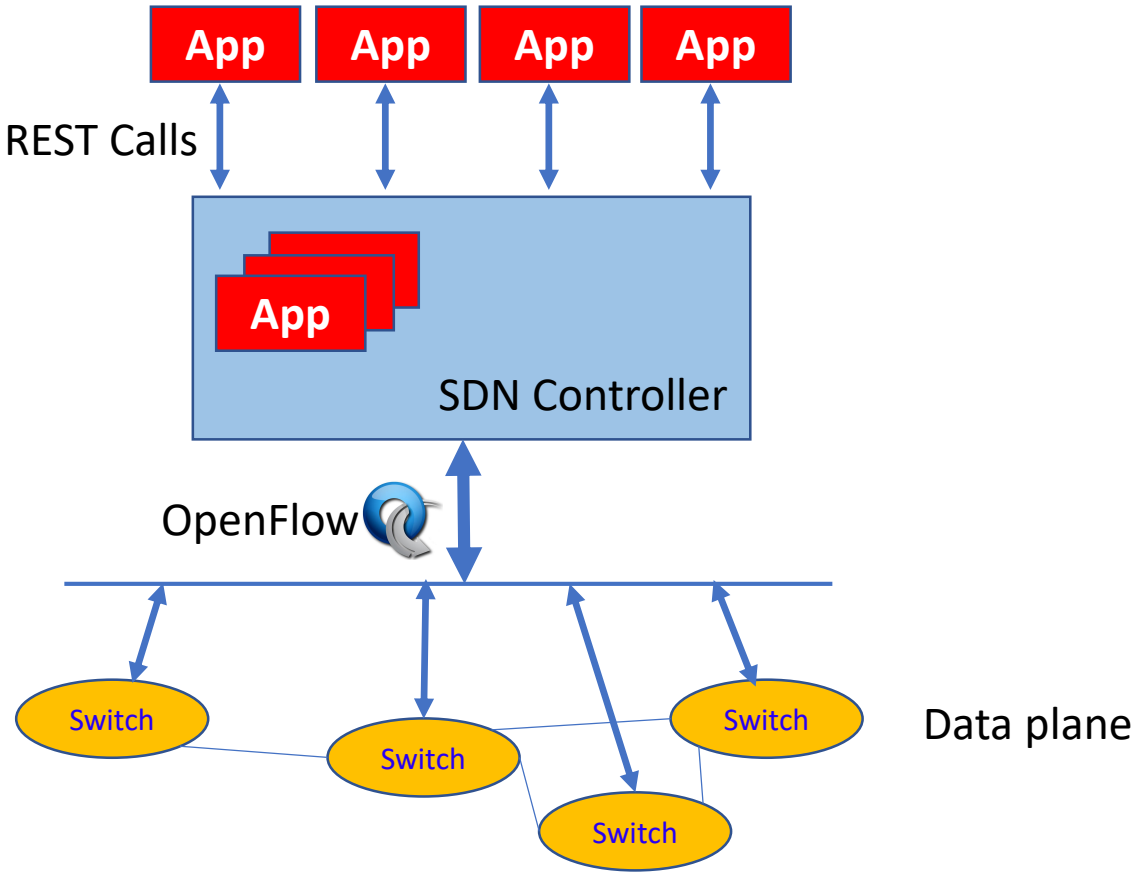
Introduction

- Software Defined Networks (SDN)
- Floodlight
- SE-Floodlight

OpenFlow

- SDN Enabler.

Application Authorization in SDN

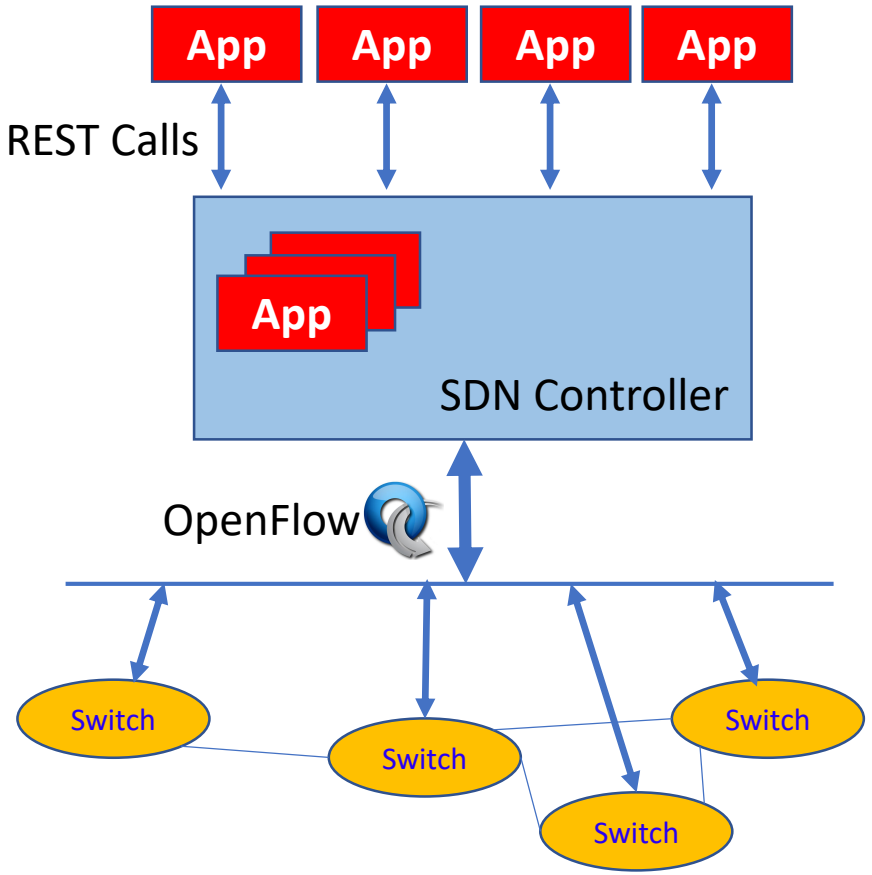


Components of the Formal Access Control Model

- Basic components
  - Apps (A),
  - Roles (R),
  - Data Exchange Operations (DXOP),
  - Types of DXOPs

Apps

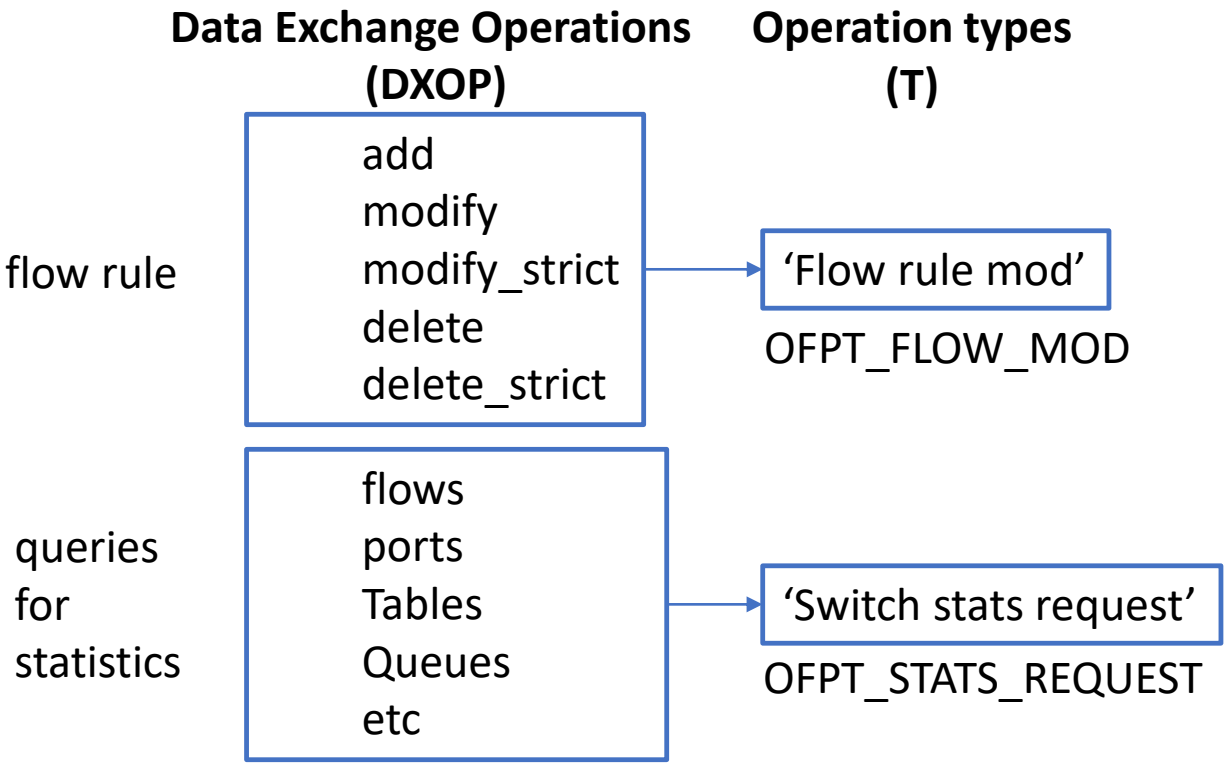
- Two types:
  - Local OpenFlow apps
  - Remote OpenFlow apps



Roles in SE-Floodlight

- Two main purposes:
  - App permission authorization
  - Flow rule conflict resolution.

Data Exchange Operations (DXOP) and Types of DXOPs



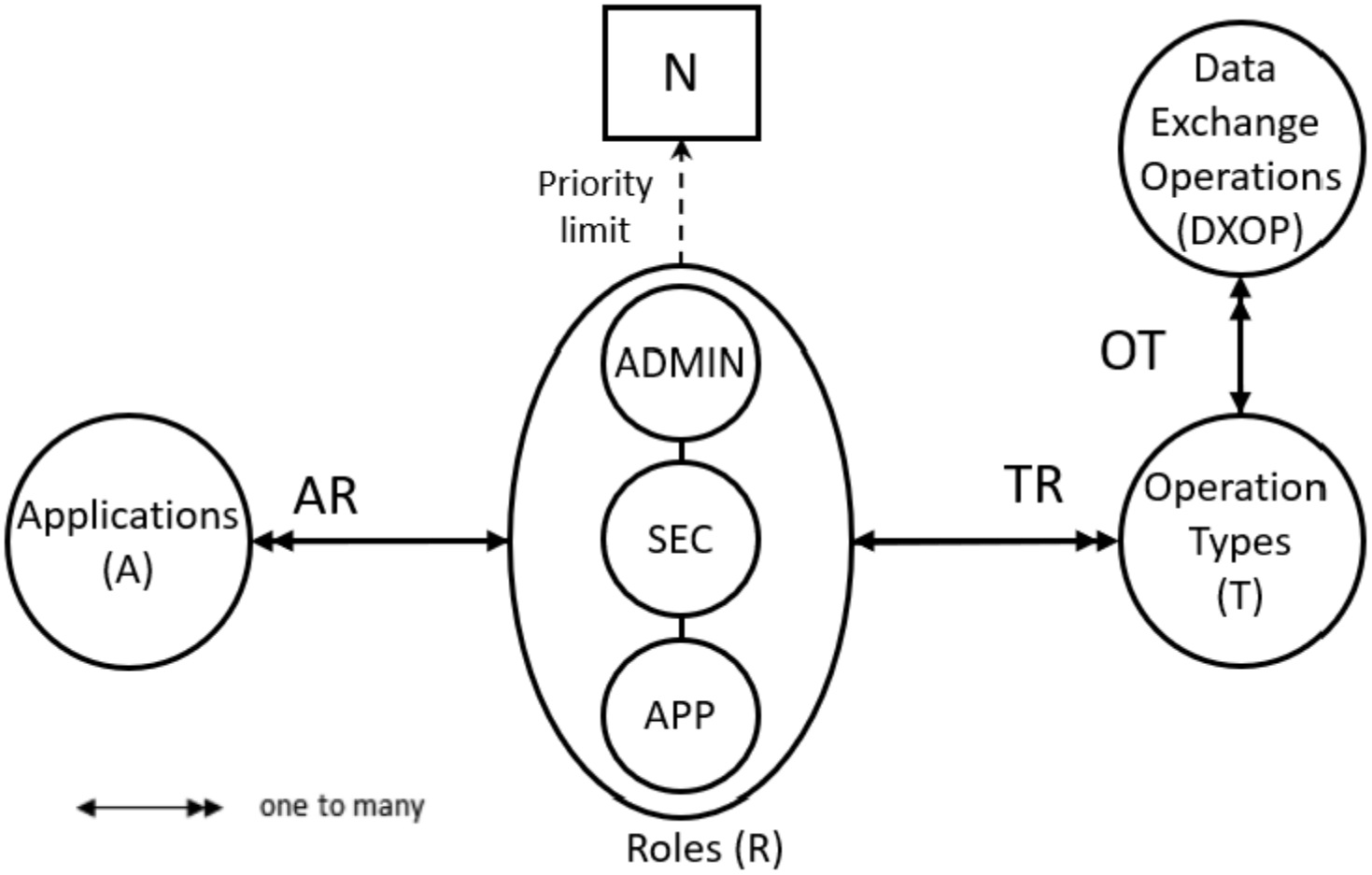
Types of Data Exchange Operations

Type ID	Type of Data Exchange Operation	Minimum Auhorization Role	Open Flow Message Type
t1	Flow removal messages	APP	OFPT_FLOW_REMOVED
t2	Flow error reply	APP	OFPT_ERROR
t3	Echo requests	APP	OFPT_ECHO_REQUEST
t4	Echo replies	APP	OFPT_ECHO_REPLY
t5	Barrier requests	APP	OFPT_BARRIER_REQUEST
t6	Barrier replies	APP	OFPT_BARRIER_REPLY
t7	Switch get config	APP	OFPT_GET_CONFIG_REQUEST
t8	Switch config reply	APP	OFPT_GET_CONFIG_REPLY
t9	Switch stats request	APP	OFPT_STATS_REQUEST
t10	Switch stats report	APP	OFPT_STATS_REPLY
t11	Packet-In return	APP	OFPT_PACKET_IN
t12	Flow rule mod	APP	OFPT_FLOW_MOD
t13	Packet-Out	SEC	OFPT_PACKET_OUT
t14	Vendor actions	ADMIN	OFPT_VENDOR
t15	Vendor features	ADMIN	OFPT_FEATURES
t16	Switch port status	ADMIN	OFPT_PORT_STATUS
t17	Switch port mod	ADMIN	OFPT_PORT_MOD
t18	Switch set config	ADMIN	OFPT_SET_CONFIG

Porras P. A et al. 2015. Securing the Software Defined Network Control Layer. In NDSS.

Authentication & Authorization

Conceptual Authorization Model



Formal Model Definitions **w/o** Flow Rule Conflict Resolution

- Basic Sets and Functions:

A: a finite set of OpenFlow apps.

T: a finite set of types of data exchange operations.

$R = \{ADMIN, SEC, APP\}$: a fixed set of three roles.

$>$: a total order on R where $ADMIN >$ SEC and $SEC > APP$.

$AR \subseteq A \times R$, a many-to-one relation, i.e., $(a,r_1) \in AR \wedge (a,r_2) \in AR \Rightarrow r_1 = r_2$, mapping each app to one role.

$TR \subseteq T \times R$, a many-to-one relation, i.e., $(t,r_1) \in TR \wedge (t,r_2) \in TR \Rightarrow r_1 = r_2$, mapping each operation type to one role.

$DXOP$: a set of possible data exchange operations where each operation $op \in DXOP$ contains a flow rule and a priority if $o =$ 'add flow rule'.

$type$: $DXOP \rightarrow T$, a function specifying the type of each operation. Equivalently viewed as a many-to-one relation $OT \subseteq DXOP \times T$, where $(o,t_1) \in OT \wedge (o,t_2) \in OT \Rightarrow t_1 = t_2$.

- Authorization Rule:

$Authorization\_rule$: $A \times DXOP \rightarrow \{T, F\}$, checks whether $a \in A$ has the right to perform an operation $o \in DXOP$.

$Authorization\_rule\ (a : A, o : DXOP) \equiv (\exists r_1, r_2 \in R \cdot (a, r_1) \in AR \wedge (type(o), r_2) \in TR \wedge r_1 \geq r_2)$.

Formal Authorization Model Definitions **without** Flow Rule Conflict Resolution.

Formal Model Definitions **with** Flow Rule Conflict Resolution

- Basic Sets and Functions:

All basic sets and functions from Table 2.

FR: a set of all possible flow rules where for each $fr_i \in FR$ there should be a priority.

priority_limit: $R \rightarrow \mathbb{N}$, the mapping of role to the highest priority an app in $r \in R$ may assign to its flow rules, where *priority_limit*(*ADMIN*) > *priority_limit*(*SEC*) > *priority_limit*(*APP*).

S: Set of switches in the network slice.

FT: $S \rightarrow 2^{FR}$, the set of flow rules currently in a switch's flow table.

rule: $DXOP \rightarrow FR$, a function that returns the flow rule $fr_c \in FR$ of an operation $op \in DXOP$ given that $type(op) =$'Flow Rule Mod'.

priority: $FR \rightarrow \mathbb{N}$, the mapping of a flow rule $fr_c \in FR$ to its priority.

$RCA(fr_c: FR, pr_c:\mathbb{N}, s_t:S) \rightarrow \{Reject, Add, Exchange\}$, a function uses rule-based conflict analysis described in [16] that returns the result of a request to add of new flow rule fr_c into $FT(s_t)$ submitted with priority pr_c. *'Reject', 'Add', or 'Exchange'* indicates whether fr_c is rejected, added without removing pre-existing rules, or exchanged with a conflicting flow rule $fr_i \in FT(s_t)$, respectively.

- Authorization Rules:

$Authorization\_rule_{op='add\ flow\ rule'} : A \times S \rightarrow \{T, F\}$, checks whether $a \in A$ has the right to insert a flow rule $rule(op)$ into $FT(s_t \in S)$.

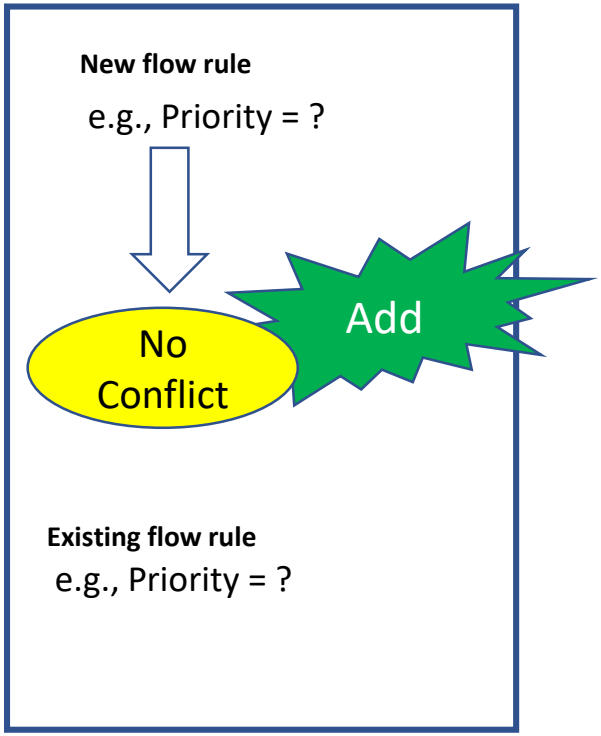
$Authorization\_rule_{op='add\ flow\ rule'}\ (a:A, s_t:S) \equiv (\exists r_1, r_2 \in R \cdot (a, r_1) \in AR \wedge (type(op), r_2) \in TR \wedge r_1 \geq r_2) \wedge (RCA(rule(op), priority(rule(op)), s_t) \in \{Add, Exchange\})$.

$Authorization\_rule_{op \in DXOP - 'add\ flow\ rule'} : A \times S \rightarrow \{T, F\}$, checks whether $a \in A$ has the right to perform a non-flow-rule-insertion operation.

$Authorization\_rule_{op \in DXOP - 'add\ flow\ rule'}\ (a:A, s_t:S) \equiv (\exists r_1, r_2 \in R \cdot (a, r_1) \in AR \wedge (type(op), r_2) \in TR \wedge r_1 \geq r_2)$

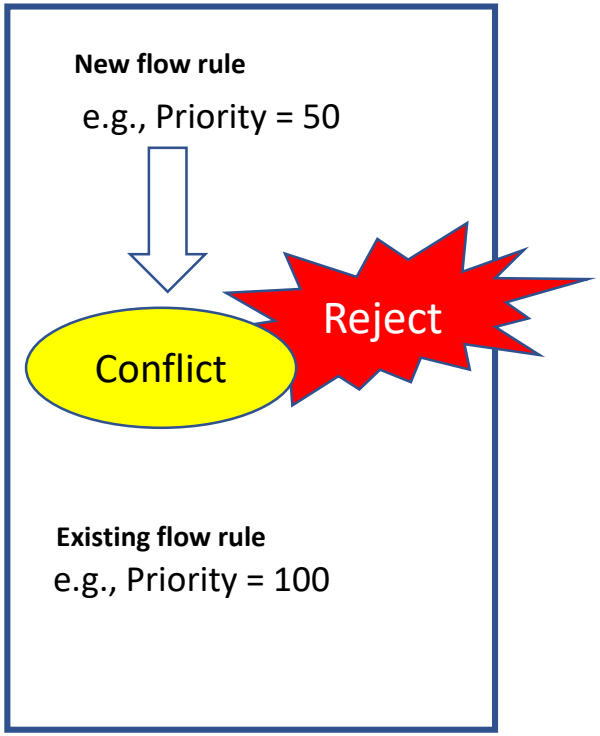
Formal Model Definitions **with** Flow Rule Conflict Resolution.

RCA : Add case

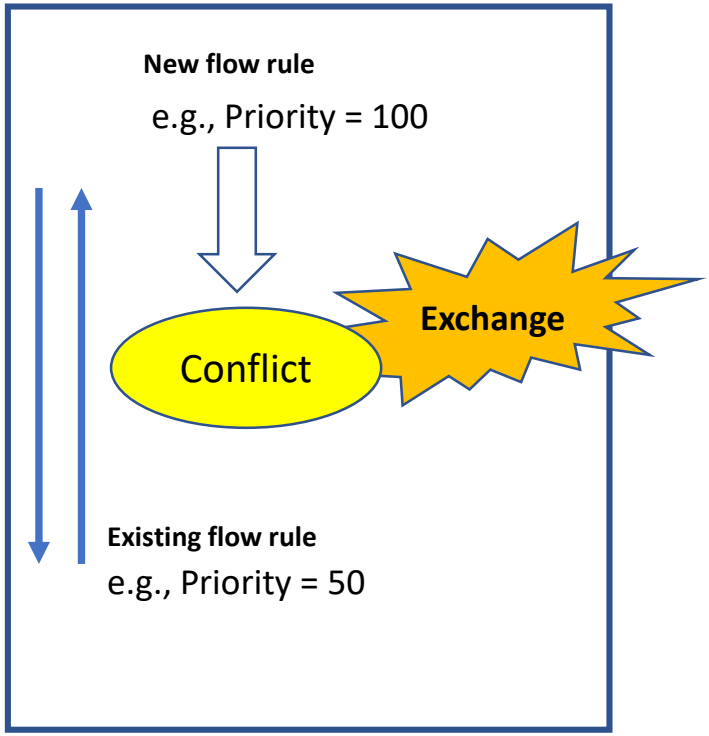

New flow rule
e.g., Priority = ?
No Conflict
Add
Existing flow rule
e.g., Priority = ?

RCA : Reject case

Example



RCA : Exchange case



Administrative Model

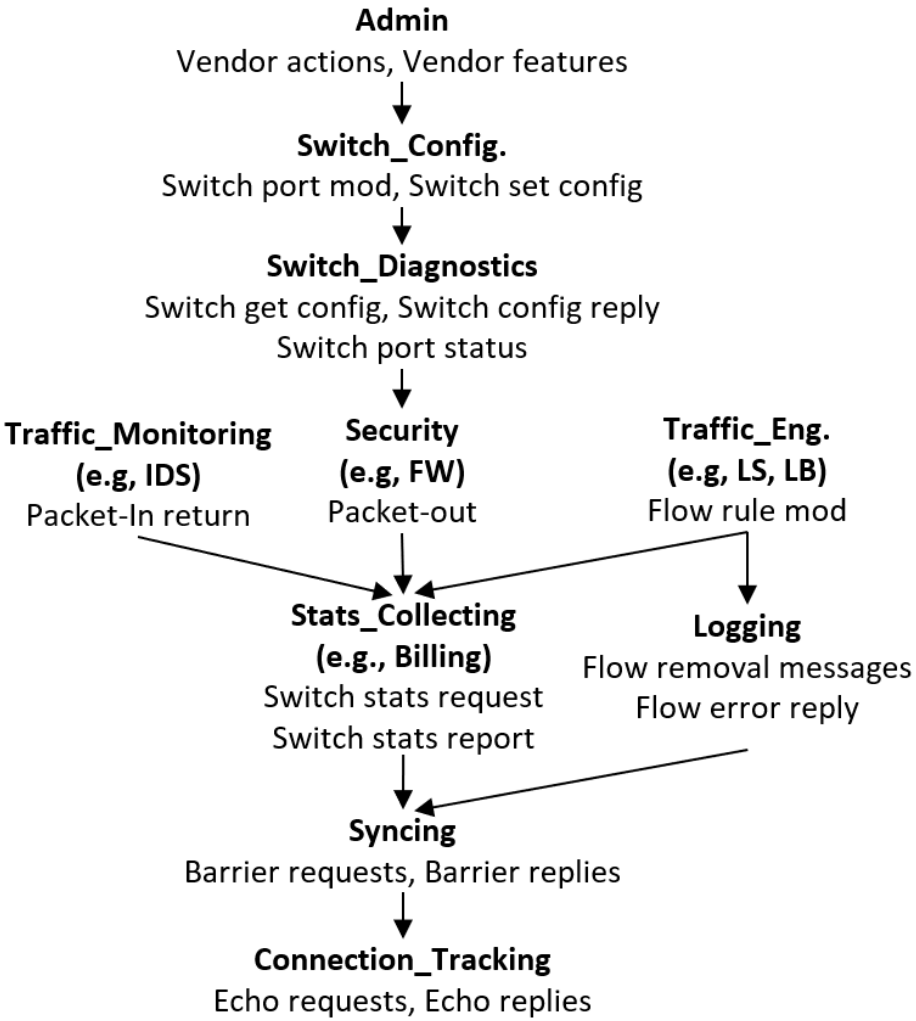
Function	Condition	Update
$addApp(a)$	$a \notin A$	$A' = A \cup \{a\}$
$deleteApp(a)$	$a \in A \wedge (a,r) \in AR$	$AR' = AR \setminus \{(a,r)\}$, $A' = A \setminus \{a\}$
$addType(t)$	$t \notin T$	$T' = T \cup \{t\}$
$deleteType(t)$	$t \in T \wedge (o,t) \in OT \wedge (t,r) \in TR$	$OT' = OT \setminus \{\forall (o,t) \in OT\}$, $TR' = TR \setminus \{(t,r)\}, T' = T \setminus \{t\}$
$addRole(r)$	$r \notin R$	$R' = R \cup \{r\}$
$deleteRole(r)$	$r \in R \wedge (a,r) \in AR \wedge (t,r) \in TR$	$AR' = AR \setminus \{\forall (a,r) \in AR\}$, $TR' = TR \setminus \{\forall (t,r) \in TR\}$, $R' = R \setminus \{r\}$
$assignApp(a,r)$	$a \in A \wedge r \in R \wedge (a,r) \notin AR$	$AR' = AR \cup \{(r,a)\}$
$revokeApp(a,r)$	$a \in A \wedge r \in R \wedge (a,r) \in AR$	$AR' = AR \setminus \{(a,r)\}$
$assignType(t,r)$	$t \in T \wedge r \in R \wedge (t,r) \notin TR$	$TR' = TR \cup \{(t,r)\}$
$revokeType(t,r)$	$t \in T \wedge r \in R \wedge (t,r) \in TR$	$TR' = TR \setminus \{(t,r)\}$
$assignOp(o,t)$	$o \in DXOP \wedge t \in T \wedge (o,t) \notin OT$	$OT' = OT \cup \{(o,t)\}$
$revokeOp(o,t)$	$o \in DXOP \wedge t \in T \wedge (o,t) \in OT$	$OT' = OT \setminus \{(o,t)\}$

Use Case - Configuration

- Five apps

$A = \{LS, LB, NIP, FW, OC\}$,
$R = \{APP, SEC, ADMIN\}$ with a total order $>$ on R ,as defined in Table 2,
$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}\}$, as labled in Table 1,
$AR = \{(LS, APP), (LB, APP), (NIP, SEC), (FW, SEC), (OC, ADMIN)\}$,
$TR = \{(t_i, APP), (t_{13}, SEC), (t_j, ADMIN) | (t_i \in T | 1 \leq i \leq 12, t_j \in T | 14 \leq j \leq 18))$,
$DXOP = \{'add\ flow\ rule', 'packet\ in', 'flow\ stats', 'packet\ out'\}$,
$Type('add\ flow\ rule') = '\ Flow\ rule\ mod', Type('packet\ in') = '\ Packet - In\ return',$
$Type('flow\ stats') = '\ Switch\ stats\ request' = '\ Switch\ stats\ report', Type('packet\ out') = '\ Packet - Out',$
$AuthorizationRule(LS, '\ add\ flow\ rule') = true, AuthorizationRule(LB, '\ add\ flow\ rule') = true,$
$AuthorizationRule(FW, '\ add\ flow\ rule') = true,$
$AuthorizationRule(LS, '\ packet\ in') = true, AuthorizationRule(LB, '\ packet\ in') = true, AuthorizationRule(NIP, '\ packet\ in') = true,$
$AuthorizationRule(FW, '\ packet\ in') = true\ AuthorizationRule(OC, '\ packet\ in') = true,$
$AuthorizationRule(LB, '\ flow\ stats') = true, AuthorizationRule(FW, '\ packet\ out') = true.$

Refined Role Hierarchy



Conclusion and Future Work

- A formal authorization model for SDN apps.
- An administration model.
- A configuration of the formal model in a use case scenario of five apps.
- A refined Role hierarchy.

- Some future goals:
  - Extension of SE-Floodlight access control model to cover all controller resources.
  - An access control model following the NIST RBAC concept.
  - Fine-grained access control using ABAC within a holistic view to SDN resources.

Thank you!
Questions?

abdullah.al-alaj@utsa.edu