SDN-RBAC: An Access Control Model for SDN Controller Applications

Abdullah Al-Alaj\textsuperscript{1}, Ram Krishnan\textsuperscript{2} and Ravi Sandhu\textsuperscript{1}

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

4\textsuperscript{th} IEEE International Conference on Computing, Communication and Security (ICCCS’2019)
Rome, Italy, October 10-12, 2019
Agenda

• Introduction
• Access Control for SDN
• SDN-RBAC Model
• App Sessions in SDN-RBAC
• SDN-RBAC System Architecture
• Use Case and Configuration
• Performance Evaluation
• Conclusion and Future Work
Introduction

Application Plane
- Routing
- Firewall
- Load Balancing
- Intrusion Prevention
- Network Visualization
- Other

REST/Java APIs

Control Plane
- Open Interface: needs control

Network Services
- Topology Service
  - Entry Pushing
- Routing Service
  - Device Management
- Statistics Collection
  - Link Discovery
- Switch Management
  - Other

Virtual Network Resources
- Topology
- Flow tables
- Switches
- Port
- Statistics
- Traffic payloads
- Configurations
- Hosts
- Links
- Devices
- VLANs
- Other

OpenFlow Protocol

Data Plane (Infrastructure)
Access control for SDN

- **Access control problem:**
  - Control which subjects (SDN apps) can access which objects (virtual network resources) for performing which actions (SDN operations).

- **Key issues for SDN include:**
  - Reducing network exposure to attack surface.
    - Apply principle of least privileges for SDN apps.
    - Minimize active permissions available for an SDN app.
  - Facilitate administration of access control.

- **Challenges:**
  - Handling sessions of controller apps (no direct user interaction).
  - Implementing access control with minimal change to controller’s code.
SDN-RBAC: Conceptual Model

App examples:
- Routing app
- Load Balancing
- Topology Visualizer
- Network Debugger
- etc.

Role examples:
- Routing
- Device Handler
- Bandwidth Monitoring
- Link Handler
- Port Handler
- etc.

Operation examples:
- Get Port BW Statistics
- Insert Flow to Switch
- get All Devices
- etc.

Session examples
- deep packet inspection session
- transmission rate monitoring session
- web-traffic filtering session
- shortest path precomputation session
- traffic redirection session
- etc.

Object Type example:
- PORT-VLAN-5, PORT-VLAN-10
- LINK-CS, LINK-ACC
- HOST-TENANT-X, HOST-TENANT-Y
- etc.
## SDN-RBAC: Formal Definitions

### Basic Element Sets
- **Sets**
  - `APPs`, `ROLES`, `OPS`, `OBS` and `OBTS`, a finite set of OpenFlow apps, roles, operations, objects and object types, respectively.
  - `PRMS = 2^{OPS \times OBTS}`, the set of permissions.
  - `SESSIONS`, a finite set of sessions.

### Assignment Relations
- `PR \subseteq PRMS \times ROLES`, a many-to-many mapping permission-to-role assignment relation.
- `AR \subseteq APPS \times ROLES`, a many-to-many mapping app-to-role assignment relation.
- `OT \subseteq OBS \times OBTS`, a many-to-one relation mapping an object to its type.

### Mapping Functions
- `assigned_perms(r : ROLES) \rightarrow 2^{PRMS}`, the mapping of role `r` into a set of permissions. Formally, `assigned_perms(r) \subseteq \{p \in PRMS|(p, r) \in PR\}`.
- `app_sessions(a : APPS) \rightarrow 2^{SESSIONS}`, the mapping of an app into a set of sessions.
- `session_app(s : SESSIONS) \rightarrow APPS`, the mapping of session into the corresponding app.
- `session_roles(s : SESSIONS) \rightarrow 2^{ROLES}`, the mapping of session into a set of roles. Formally, `session_roles(s) \subseteq \{r \in ROLES|(session_app(s), r) \in AR\}`.
- `type : OBS \rightarrow OBTS`, a function specifying the type of an object, where `(o, t_1) \in OT \land (o, t_2)\in OT \Rightarrow t_1 = t_2`.
- `avail_session_perms(s : SESSIONS) \rightarrow 2^{PRMS}`, the permissions available to an app in a session = \( \bigcup_{r \in session_roles(s)} assigned_perms(r) \).

---

*Used directly in checkAccess system function.*
### SDN-RBAC: Specifications of System Functions

#### Session Creation/Deletion

- **Function**: `createSession(a : APPS, s : SESSIONS, ars : 2ROLES)`
- **Authorization Condition**: \( ars \subseteq \{ r \in \text{ROLES} \mid (a, r) \in \text{AR}\} \land s \notin \text{SESSIONS} \)
- **Update**: \( \text{SESSIONS}' = \text{SESSIONS} \cup \{s\}, \text{app_sessions}'(a) = \text{app_sessions}(a) \cup \{s\}, \text{session_roles}'(s) = ars \)

#### Adding/Dropping Active Role

- **Function**: `deleteSession(a : APPS, s : SESSIONS)`
- **Authorization Condition**: \( s \in \text{app_sessions}(a) \)
- **Update**: \( \text{SESSIONS}' = \text{SESSIONS} \setminus \{s\}, \text{app_sessions}'(a) = \text{app_sessions}(a) \setminus \{s\}, \text{session_roles}'(s) = \text{session_roles}(s) \setminus \{r\} \)

- **Function**: `addActiveRole(a : APPS, s : SESSIONS, r : ROLES)`
- **Authorization Condition**: \( s \in \text{app_sessions}(a) \land (a, r) \in \text{AR} \land r \notin \text{session_roles}(s) \)
- **Update**: \( \text{session_roles}'(s) = \text{session_roles}(s) \cup \{r\} \)

- **Function**: `dropActiveRole(a : APPS, s : SESSIONS, r : ROLES)`
- **Authorization Condition**: \( s \in \text{app_sessions}(a) \land r \in \text{session_roles}(s) \)
- **Update**: \( \text{session_roles}'(s) = \text{session_roles}(s) \setminus \{r\} \)

#### Access Check

- **Function**: `checkAccess(s : SESSIONS, op : OPS, ob : OBS)`
- **Authorization Condition**: \( \exists r \in \text{ROLES} : r \in \text{session_roles}(s) \land ((\text{op}, \text{type}(\text{ob})), r) \in \text{PR} \)

---

*retrieving the object type*
Sessions in SDN-RBAC

• Two types:
  • Atomic network sessions
    • Self-contained task definition.
  • Dependent network sessions.
    • Inter-session dependency
    • Conduct inter-session interaction at runtime.
Methods for Inter-session Interaction for SDN-RBAC

Atomic sessions

Two sessions access shared data

Conditional session creation

Interaction via inter-session interaction APIs

Active role set sent from master to slave sessions

<table>
<thead>
<tr>
<th>Atomic sessions</th>
<th>Two sessions access shared data</th>
<th>Conditional session creation</th>
<th>Interaction via inter-session interaction APIs</th>
<th>Active role set sent from master to slave sessions</th>
</tr>
</thead>
</table>

--- creates a session (From the creator to the created session).

--- access shared data.

--- session interaction via session interaction API.

w/r : read/write operation.

c : condition that triggers session creation.

I : session interaction API (managed by the system).

a : active role set sent along with session creation request.
Examples of conditions for session creation:
- bandwidth consumption cap exceeded,
- new device detected,
- at system start-up.
- etc.

Session handling APIs usage examples:
- Getting names of all active sessions
- Getting active role set of a session.
- Getting session’s status.
  - e.g., idle time, up time, etc.
- Passing information and notifications between sessions.
  - e.g., results of calculations.
Session Handling Approaches

- Who is responsible of specifying:
  - (T) the tasks and corresponding sessions.
  - (C) the condition for session creation/deletion.
  - (A) the active role set.
  - (R) role to be added/dropped during execution.

DD – determined by developer at design-time.
CR – determined by controller at run-time.
SR – determined by session at run-time.
SDN-RBAC: System Architecture

Application Plane

Network App

Controller Services

Protected Objects

SDN-RBAC Framework

AspectJ Hooking

Request by session

Result to session

Authorized request

Access decision (grant/deny)

Available session permissions

SDN-RBAC Policy (PIP)

Request Evaluation & Decision (PDP)

Session’s access request

Topology Service

Routing Service

Statistics Collection

Link Discovery

Device Management

Other

Topology

Flow Tables

Statistics

Links

Devices

Other

Read/Write

Result

Service APIs

Service Agents

Event Listeners

Service APIs

Service Agents

Event Listeners

I·C·S
The Institute for Cyber Security

Center for Security and Privacy
Enhanced Cloud Computing

Computer Science

University of Texas at San Antonio
Use Case: Data Usage Manager
(A Multi-session App)

Data Cap Analysis Session

- Read Port BW Statistics
- Identify BW Violations (If cap exceeded)
- Get All Devices
- Identify Violating device

Data Cap Enforcing Session

- Read Black List
- Insert Flow Rule (If new device)

Every 60 seconds

Every 5 seconds

Update Black Listed Devices

Requires BW Monitoring Role

Requires Flow Mod Role

Requires Device Handler Role
Use Case: Configuration in SDN-RBAC

- **Use case sets:**
  - $APPS = \{DataUsageCapMngr\}$.
  - $ROLES = \{Device Handler, Bandwidth Monitoring, Flow Mod\}$.
  - $D = \text{set of all network devices}$.
  - $FT = \text{set of all flow tables in all switches}$.
  - $PS = \text{set of all port statistics in all switches}$.
  - $OBS = \{D, FT, PS\}$.
  - $OBTS = \{DEVICE, PORT-STATS, FLOW-TABLE\}$.
  - $OT = \{(D, DEVICE), (PS, PORT-STATS), (FT, FLOW-TABLE)\}$.

- **Permissions:**
  - $PRMS = \{p_1, p_2, p_3\}$ with
    - $p_1 = (\text{getAllDevices, DEVICE})$,
    - $p_2 = (\text{getBandwidthConsumption, PORT-STATS})$,
    - $p_3 = (\text{InsertRule, FLOW-TABLE})$.

- **Permissions assignment:**
  - $PR = \{(p_1, Device Handler), (p_2, Bandwidth Monitoring), (p_3, Flow Mod)\}$.
  - $\text{assigned_perms(Device Handler)} = \{p_1\}$,
    - $\text{assigned_perms(Bandwidth Monitoring)} = \{p_2\}$,
    - $\text{assigned_perms(Flow Mod)} = \{p_3\}$.

- **Role assignment:**
  - $AR = \{(DataUsageCapMngr, Device Handler), (DataUsageCapMngr, Bandwidth Monitoring), (DataUsageCapMngr, Flow Mod)\}$.

- **Sessions:**
  - $SESSIONS = \{DataUsageAnalysisSession, DataCapEnforcingSession\}$.
  - $\text{app_sessions(DataUsageCapMngr)} = \{DataUsageAnalysisSession, DataCapEnforcingSession\}$.
  - $\text{session_app(DataUsageAnalysisSession)} = \{DataUsageCapMngr\}$,
    - $\text{session_app(DataCapEnforcingSession)} = \{DataUsageCapMngr\}$.

- **Active role sets:**
  - $\text{session_roles(DataUsageAnalysisSession)} = \{Device Handler, Bandwidth Monitoring\}$.
  - $\text{session_roles(DataCapEnforcingSession)} = \{Flow Mod\}$.

---

1 Sets with this mark in the table include minimum elements enough to understand the use case. Remaining elements are avoided for more convenience and readability.

The configuration of the $DataUsageCapMngr$ and its two sessions as a use case in SDN-RBAC.
Demonstration in Floodlight: Data Cap Manager App

**Snapshot 1**
```
roller.statistics.IStatisticsService.getBandwidthConsumption, PORT-STATS)
The method net.floodlightcontroller.topology.ITopologyService.getAllLinks
is called by session net.floodlightcontroller.datausage.mngr.DataUsageAnalysisSession
Unauthorized access requested by session (DataUsageAnalysisSession)
Reason: None of session active roles contains a corresponding permission
Active roles set for this session: [Device Handler, Bandwidth Monitoring]
16:36:32.630 INFO [n.f.l.l.LinkDiscoveryManager:Scheduled-3] Sending LLDP packets out of a
```

Snapshot of authorization check result for `getAllLinks()` operation requested by `DataUsageAnalysisSession` - **Access Denied.**

**Snapshot 2**
```
The method net.floodlightcontroller.statistics.IStatisticsService.getBandwidthConsumption
is called by session net.floodlightcontroller.datausage.mngr.DataUsageAnalysisSession
requested by session (DataUsageAnalysisSession)
The method net.floodlightcontroller.topology.ITopologyService.getAllLinks
```

Snapshot of authorization check result for `getBandwidthConsumption()` operation requested by `DataUsageAnalysisSession` - **Access Granted.**
Average execution time required by SDN-RBAC components to finish checking 50 operations with varying number of roles.

On average: 0.031 ms overhead for 50 operations.
Controller with SDN-RBAC Performance Evaluation

Average total execution time required to finish the 50 operations called 1000 times including and excluding SDN-RBAC.

![Graph showing execution time vs number of roles](image)

Execution time (ms)

Number of roles

- Floodlight Only
- 0 roles
- 1 roles
- 2 roles
- 3 roles
- 4 roles
- 5 roles
- 6 roles
- 7 roles
- 8 roles
- 9 roles
- 10 roles
- 11 roles
- 12 roles
- 13 roles
- 14 roles
- 15 roles
- 16 roles
- 17 roles
- 18 roles
- 19 roles

Execution time ranges from 2700 ms to 2950 ms.
In this work:

• A formal model (SDN-RBAC) for SDN controller apps.
• Methods for Inter-session Interaction.
• Different approaches for handling session instances of an app.
• Implementation of the model, as proof-of-concept prototype, in Floodlight platform.
• We used hooking techniques without any change to the code of Floodlight native modules.
• We show the system’s usability using a test app with multi-session execution.
• Performance evaluation with various number of roles.

Future research

• Hierarchical priority groups for conflict resolution between apps operations.
• Role-based administration of SDN-RBAC and its extensions.