Establishing a Session Database for SDN Using 802.1X and Multiple Authentication Resources
Joint Meeting of the VDE/ITG Sections 5.2.2 & 5.2.4

Frederik Hauser, Mark Schmidt, Michael Menth
frederik.hauser@uni-tuebingen.de

http://kn.inf.uni-tuebingen.de
Motivation

► Authentication and authorization (AA) as common mechanisms for securing communication networks
  ▪ Status quo: coarse-granular access permissions
  ▪ E.g., VLAN tagging, network admission rules, ...
► Extensive work on more fine-granular network control systems using software-defined networking (SDN)
  ▪ Stateful and indentity-centric access permission rules
  ▪ E.g., Ethane, Resonance, Kinetics, ...
  ▪ Requires a network-wide session database and reliable authentication mechanisms
► Most widely used AA mechanisms in SDN
  ▪ Static MAC-address-to-identity mapping
  ▪ Web frontend authentication
Most widely used standard for port-based authentication and authorization in access networks (e.g., eduroam)

Encompasses frontend and backend authentication

- Frontend authentication: Extensible Authentication Protocol (EAP)
- Backend authentication: Remote Dial-In User Service (RADIUS)
EAP and RADIUS in 802.1X

Initialization of AA

Identity-based AA

Authentication

1. EAPOL-Start
2. EAP-Response / Identity
3. **Authentication** using EAP (EAP-MD5, EAP-TTLS, ...)
4. Port authorized

EAP and RADIUS are used in 802.1X for port authorization and authentication. The process involves:

1. Supplicant initiates the connection with the Authenticator using EAPOL.
2. The Authenticator sends an EAP-Request / Identity to the Supplicant.

Authentication using EAP (EAP-MD5, EAP-TTLS, ...) is performed between the Supplicant and Authenticator.
Limitations of 802.1X

► Dependence on RADIUS or Diameter for backend AA
  ▪ AA data is stored on external resources (e.g., SQL database)
  ▪ Unnecessary overhead for specific setups

► Change of authorization
  ▪ Backend AA does not support unsolicited messages, changes in authorization cannot be applied to existing sessions
    – Extension to RADIUS specified in RFC 5176, but hardly implemented

► Stateless property of RADIUS
  ▪ Unlimited number of concurrent authorized network access by using one credential
    – Simultaneous-Use extension, but hardly deployed
    – Session management by little standardized RADIUS accounting messages, SNMP, Finger or telnet

Frederik Hauser, Mark Schmid, Michael Menth: Establishing a Session Database for SDN Using 802.1X and Multiple Authentication Resources, ITG Joing Meeting 5.2.2 & 5.2.4, 05.12.2016
Related Work

- Usage of hostapd (FAUCET, AuthFlow, FlowIdentity, …)
  - EAP packets are forwarded to an instance of hostapd (user-space 802.1X authenticator)
  - Channel between the SDN controller and hostapd to report successful authentication attempts

- Protocol extensions to 802.1X (FlowNAC)
  - EAPOL-in-EAPOL encapsulations for authentication and authorization of different applications running on a network host
  - Requires changes of all 802.1X components
Objectives of this Work

► Use 802.1X for authentication and authorization in SDN
  ▪ Compliance to standards (no modification on network endhosts or RADIUS-based AA infrastructure required)
  ▪ Solve major shortcomings of current 802.1X infrastructures

► Architectural approach
  ▪ AA module (AAM) serving as application for an SDN controller
    – Legacy 802.1X authenticator (authenticator mode)
    – Alternate AA resources (authentication server mode)
  ▪ Session database
**Authenticator Mode**

**Authenticator mode**
- AAM adopts the functionality of a legacy 802.1X authenticator
- No need for implementing specific EAP types (e.g., EAP-TLS, …)
- AAM implements mechanisms to translate authorization data (e.g., a VLAN tag) into corresponding SDN rules
- **Scenario:** use an existing RADIUS infrastructure for AA in SDN

---

Frederik Hauser, Mark Schmid, Michael Menth: Establishing a Session Database for SDN Using 802.1X and Multiple Authentication Resources, ITG Joing Meeting 5.2.2 & 5.2.4, 05.12.2016
Integrating alternative AA resources: **Authentication server mode**

- AAM acts as authenticator and authentication server
  - Implementation of EAP type specifics required
  - Implementation of AA resource interfacing required
- **Scenario:** use AA resources (e.g., a SQL database or an LDAP server) for AA in SDN
Choosing between authenticator and authentication server mode and different AA resources

- Port-based selection
  - AA resource is selected using the switch port identifier

- Identity-based selection
  - AA resource selection according to the EAP (outer) identity
Network-Wide Session Database

- Session database contains information about all authenticated and authorized identities

- AAM triggers actions on the session database
  - Session removal (e.g., in case of a port down event)
  - Reauthentication

- External applications can interact with the AAM by using communication techniques like REST interfaces

```
{ test@group1.local : { max_sessions : 1, sessions : (  
    { aaa_time : Mo 13 Jun 2016 14:16:26 CEST,  
      aaa_method : Radius(ip=10.0.20.100, meth=EAP-MD5),  
      phys_port : OF-Switch(ip=10.0.20.222, port=1),  
      assigned_vlans : (10)},  
    { aaa_time : Mo 13 Jun 2016 14:18:31 CEST,  
      aaa_method : Radius(ip=10.0.20.100, meth=EAP-MD5),  
      phys_port : OF-Switch(ip=10.0.20.222, port=2),  
      assigned_vlans : (10)},  
  )
}
```
Prototypical Implementation & Functional Validation (1/2)

Prototypical implementation for the Ryu SDN controller framework
- Python-based 802.1X authenticator implementation
  - Extension of \textit{dpkt} (network packet generation and parsing) by RADIUS and EAP protocol handlers

Testbed
- OpenFlow switches
  - Hardware-based OpenFlow switches
    - HP Enterprise 2920 (firmware version 16.01.0006)
      - 100% EAPoL frame droppings (incompatibility issue)
    - Zodiac FX (firmware version 0.66)
      - Expected behaviour
  - Software-based OpenFlow switches
    - Open vSwitch (version 2.4.0)
      - Expected behaviour
- KVM-/QEMU-based virtual machines
Various tests within a functional validation scenario

- E.g. usage of multiple AA resources, session database, ...
Outlook & Conclusion

► Outlook: Scalability of the AAM
  ▪ AAM load highly dependent on the mode of operation (authenticator or authentication server mode)
  ▪ Approach: deploy the AAM as network function
    – AA processes are independent and can be parallelized
    – SDN controller installs flow rules for EAPoL frame forwarding to particular AAM instances that interact with the network-wide session database

► Conclusion
  ▪ Proposition to adopt 802.1X for authentication and authorization in SDN with multiple AA resources and a network-wide session database
  ▪ Design and prototypical implementation of the AA module (AAM) for running within the Ryu SDN controller framework