Secure SDN Authentication
(DNS based PKI model)

Author:
Hosnieh Rafiee
ietf(at)rozanak.com
Summary

Problem: No flexibility for PKI model

Solution: Combination of DANE, DNSSEC and DDNS to enable:

- Automatic update of certificates
- Enable Tenants to manage and assign resources themselves
- No need to maintain and administrate a/more PKI server(s) as well as DNS server
  - Only maintenance of DNS server is enough (Reduce CapEx)
Existing PKI Model

Current Situation

- Only Operator admin need to control the PKI model

- All keys need to be under the control of operator admin for authorization management

- No categorization for keys
  - App1.Tenant1 PK
  - App2.Tenant2 PK
  ...

Operator1 Admin

Tenant1

Tenant2

Third party

Third party
Enable Tenants to manage and assign resources themselves

**Current Situation**

- All keys need to be under the control of operator admin for authorization management
- No categorization for keys
  - App1.Tenant1 PK
  - App2.Tenant2 PK

**Improvement**

1. Each Tenant controls the information of his own zone
2. Keys also can be updated by either automatically or manually by the key owner
Enable Tenants to manage and assign resources themselves

- All keys need to be under the control of operator admin for authorization management - no categorization for keys
- App1. Tenant1 PK
- App2. Tenant2 PK

1- Each Tenant controls the information of his own zone
2- Keys also can be updated by either automatically or manually by the key owner

- Each customer access its own zone and can update key for its own resources
- Operator1 can define some access control templates for tenants and assign to them
- Each Tenant can assign access control itself to third party without major dependency to operator1
Problem with the existing PKI Model

- SDN authentication is usually based on TLS or certificates
- Problem with certificates based authentication
  - Self-Signed Certification (Spoofing, MITM attacks, Key management)
  - Public CA (PKI)
    - Compromised CA → compromised all resources that uses that CA database
    - Single operator accessibility and dependency to the main admin of the CA to define and control keys and other resources → Disallow resell of a part of the network in multi-tenancy
    - Self-update of certificates are not possible
  - Local CA (PKI)
    - This is similar to public CA but only the chance of having compromised CA is lower
Advantages of The Proposed Model - I

- Improve the existing PKI models, for SDN and NFV use cases
  - Reduce the scope of possible attacks on PKI mode (multi-tenancy and remove the need for maintenance and administration of PKI servers).

- The use of existing protocols and existing infrastructure
  - DNS (RFCs 1034,1035), DANE (RFC 6698), DDNS (RFC 3007)

- Provide a secure authentication model for different components of SDN and NFV solutions. Two example scenarios:
  - vCPEs controlled by ISPs who are the customers of operator
  - A part of vEPCs infrastructure sold to a customer and resold to third parties that they want also to resell it to end customers (IPsec keys can be updated via this model)

- Allow each tenant to control access (authorization) on own resources with no dependency to the operators. → Solve the high level authorization problem for SDN ad NFV solutions
A Solution for Hierarchical Multi-Tenancy Problem

- Allows operators to sell part of their infrastructure to their customers
- Allows their customers to re-sell a part of their leased infrastructure to third parties
- Allows third parties to re-sell their leased resources to end customers

Example of sell and re-sell of the resources
SDN Example: Enable Tenants to manage and assign resources themselves

Operator1 defines different parent policy templates and store them in its resource policy database

**Step1: Agreement between Tenant1 and Operator** to use Operator1’s resources
- Operator 1 defines Tenant1 zone and assign the policy indices to this Tenant1 which identifies its access control (only index the whole policy is in resource policy DB)

- DNS is a powerful database
- Tenant 1 will not have any dependency to operator1 for modifying authorization information to its third party
- Quick authorization in the same step as authentication by orchestrator (DNS proxy)
SDN Example – Automatic Key Update by Third party

Third party wants update its keys and TLSA record

Third party generates a key pair and ask Tenant1 to sign it.

Tenant1

I want to update my zone with thirdparty.tenant1 sub zone and assign policy indices 1, 2, 3

Extended DDNS

Request signed by Tenant1

Ack

Orchestrator service (operator 1) (DNS proxy)

I want to update my TLSA and PK

Tenant1 signed my values (Extended DDNS)

Ack

Ack Update

Verification of request and update Tenant1 zone

Verification of the request and update thirdparty.tenant1 with PK and TLSA value of thirdparty

Verification of the request and update thirdparty.tenant1 with PK

Local DNS

Ack
SDN Example – Third party Gain Access to Resources

Third party via using an application wants to configure its resources in southbound via a SDN controller.

The TLS session expires after RTT elapsed or by sending an end session request message.

PP= Parent policy Index(es)
vCPE Example Scenario

- vCPE assigns IP address and controls devices inside the “Home Network”
- End User can configure its vCPE via its web user interface
- Operator can configure vCPE according the network changes via SDN controller without sending any technician to home of end user to configure the CPE
- All authentication among these different components is based on keys and certificates
- vDNS is the PKI storage and authorization indexes
Problem: No flexibly for PKI model

Solution: Combination of DANE, DNSSEC and DDNS to enable:

- Automatic update of certificates
- Enable Tenants to manage and assign resources themselves
- No need to maintain and administrate a/more PKI server(s) as well as DNS server
- Only maintenance of DNS server is enough (Reduce CapEx)

Thank you!