IETF94
2 Nov. 2015
Yokohama
SDNRG WG

Secure SDN Authentication & Authorization for Multi-tenancy

(DNS based PKI model)

Author:

Hosnieh Rafiee

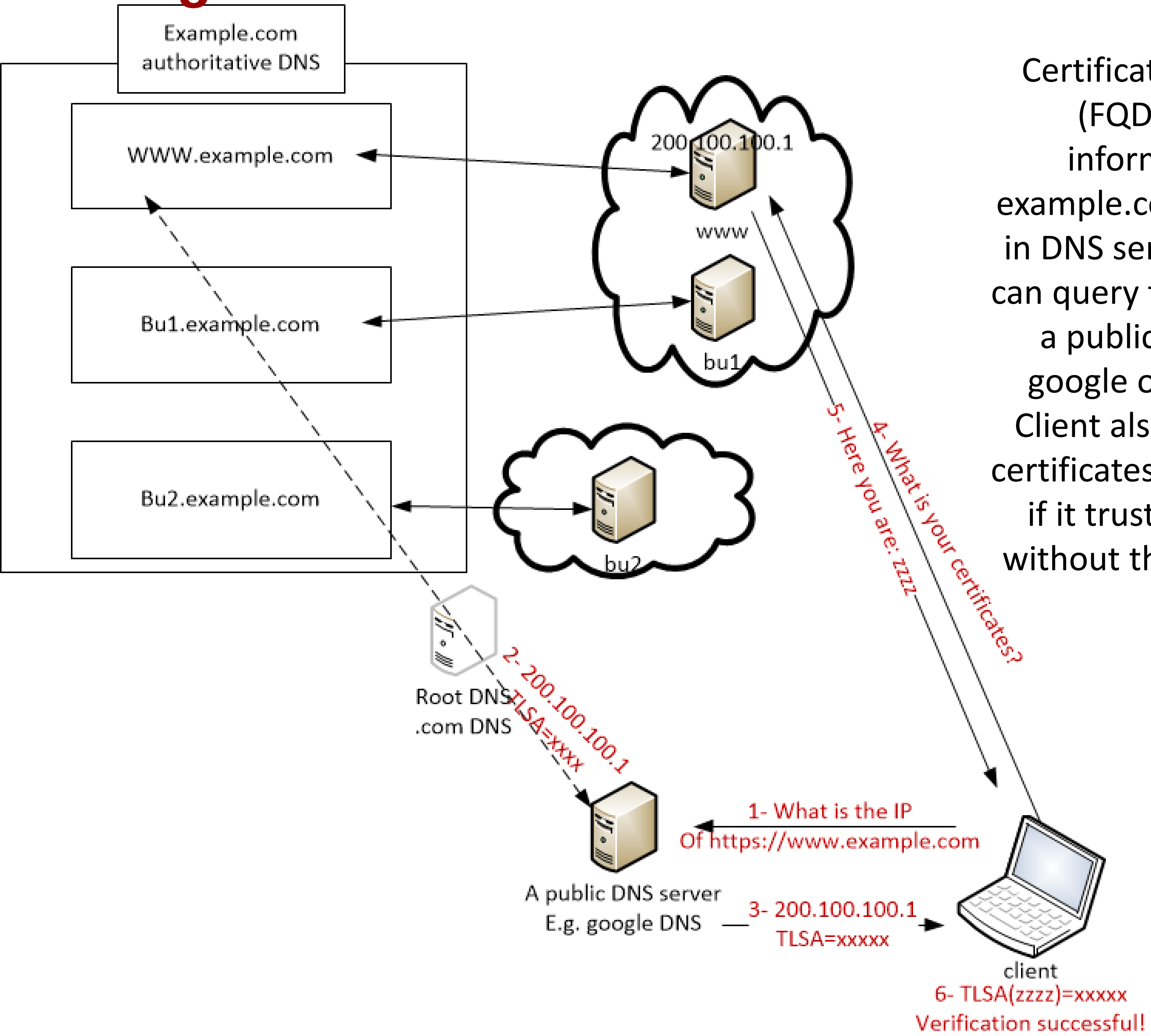
Ietf{at}rozanak.com

Motivation

Problem: secure SDN authentication, authorization and binding of authentication and authorization for multi-tenancy environment

- Solution: Using DANE, DDNS → flexibility in certificates management.
  - ✓ 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)

Background - the use of DNS and DANE in Internet



Certificates, IP, host name (FQDN), and other information of each example.com server is stored in DNS server where a client can query this information via a public DNS server like google or ISP DNS server. Client also can validate the certificates using DANE (TLSA) if it trusts the DNS server without the need of a public CA

How we want to use DANE?

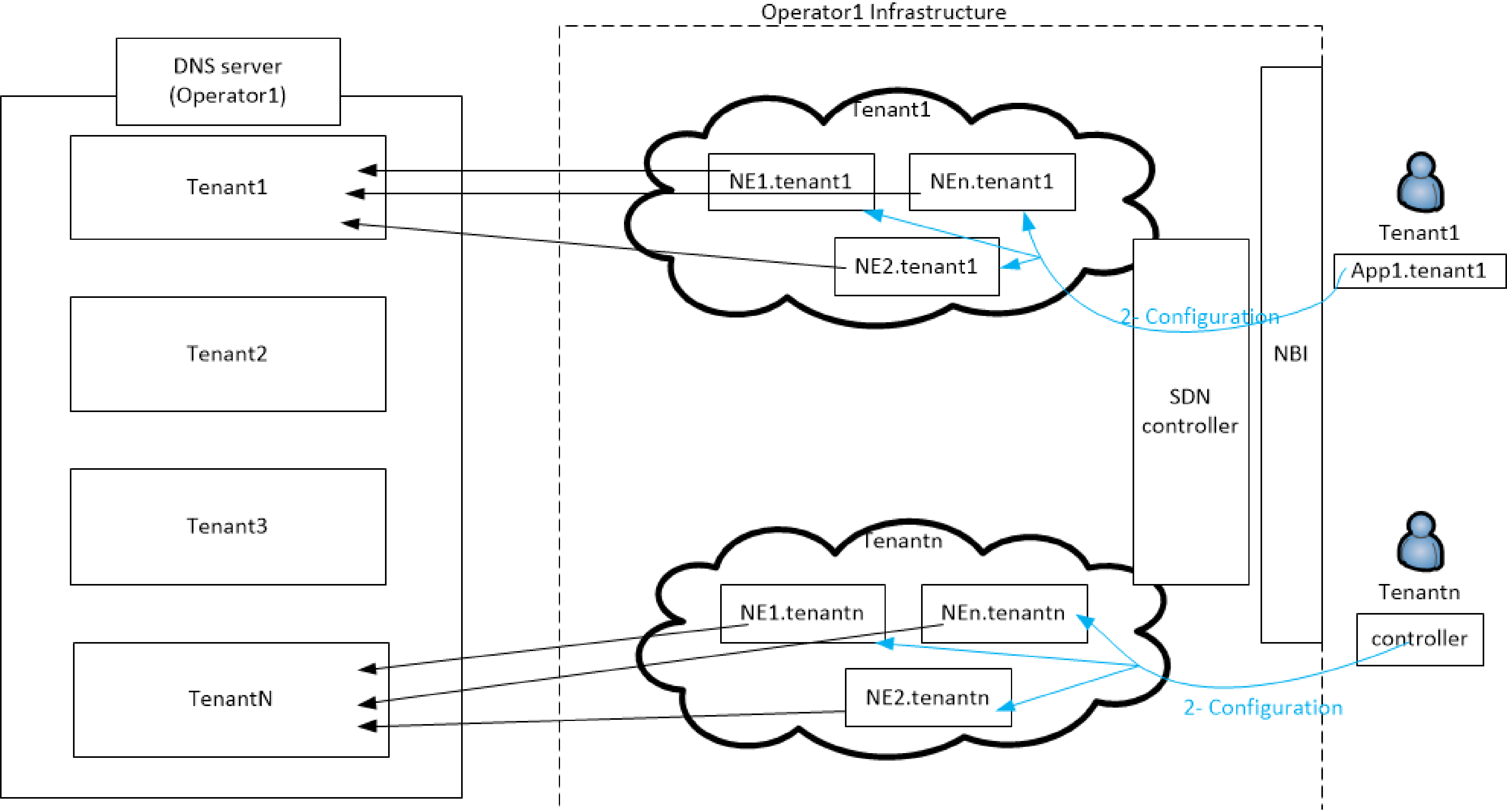
How someone looked at this approach!

The existing DNS system
Is like tree based database
DANE allows each leaves to store the certificates
of a server

What we want to do →

Not only use DANE but also
Bind authentication (certificates)
with authorization

How to use DNS and DANE in SDN solution



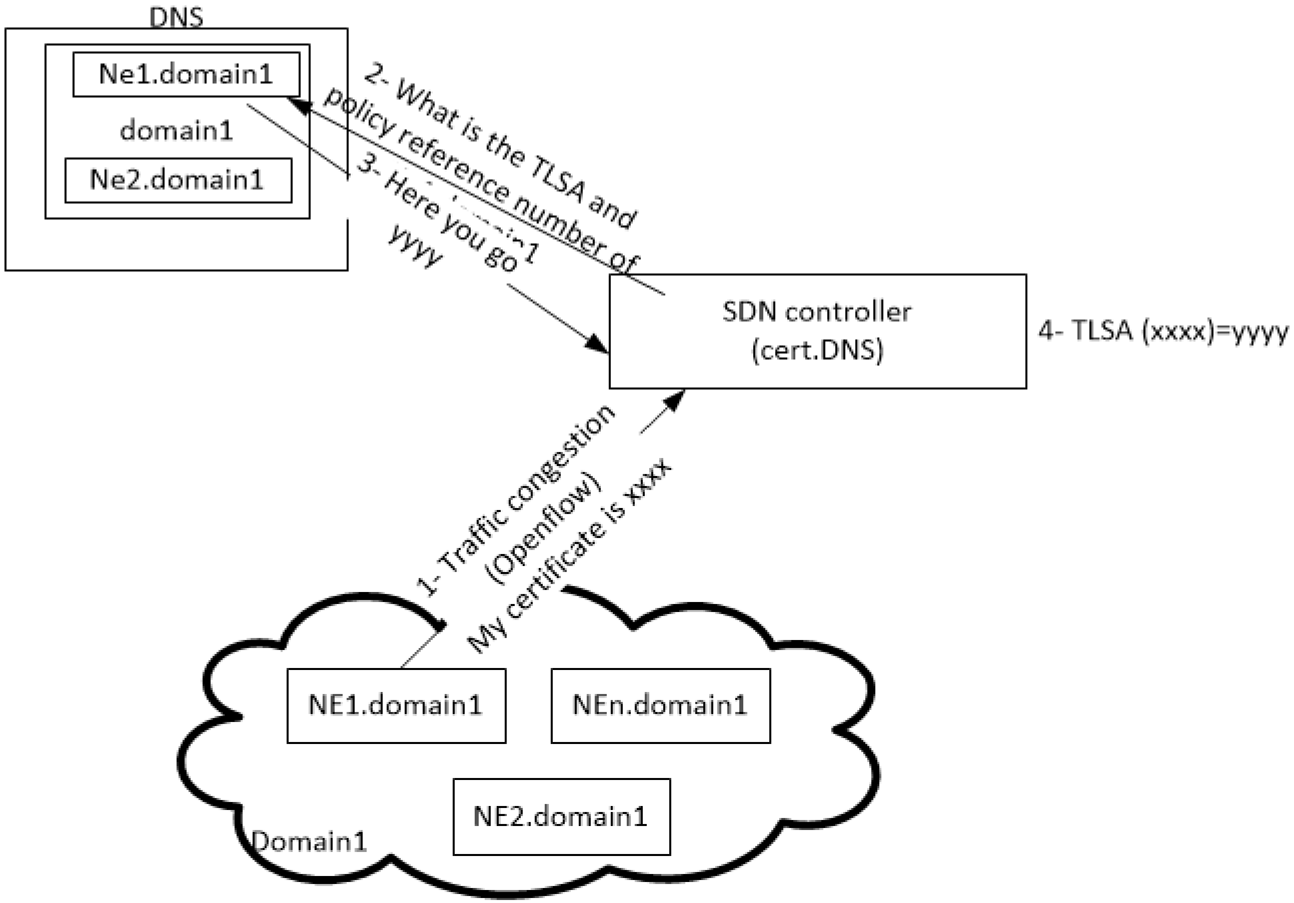
After the agreement with operator, the trust between tenant and operator domain is established and the domain for tenant is created where the certificates of tenant1 is stored in operator DNS, the reference number(s) of resource policy (authorization) is stored in tenant domain where tells what resources can be accessible by this tenant.

Tenant might have their own SDN controller to control their own resources or they might use an application to access SDN controller in operator domain → all authentication is based on DANE

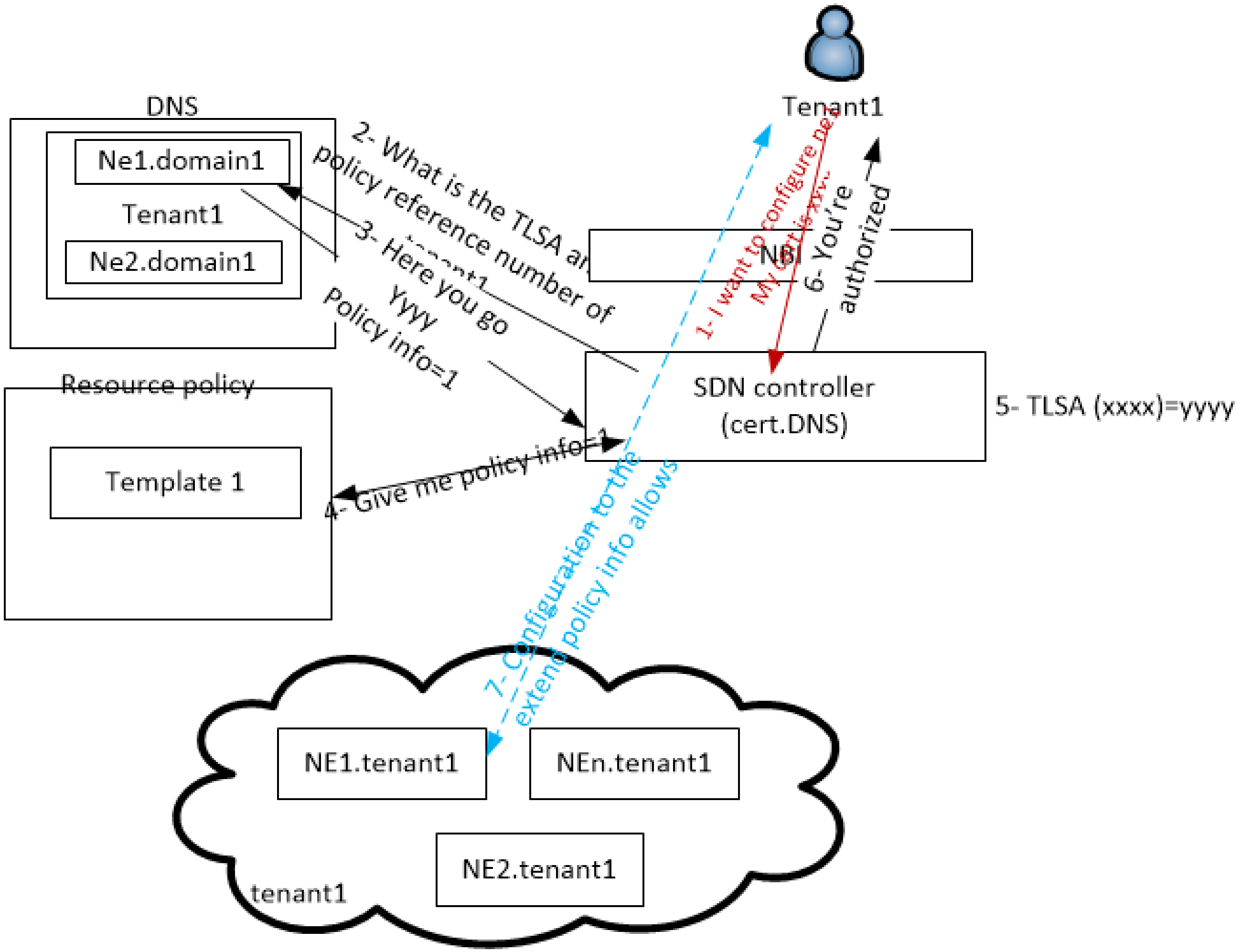
Why to use this approach?

- DNS is flexible and it is working for years on internet
- It is the nature of DNS to allow resource isolation with minimum efforts → admin of www can be separate from admin of bu1 in our first example
- There are already protocols for querying DNS server, updating records, etc.
- To have a flexible network, we can use domains instead of IPs for different components of SDN solution → the use of DNS is inevitable
- Storing keys inside the network elements is not flexible and scalable but this is a way of certificates management
  - How many keys should be stored in each network elements to allow different components of SDN controller to communicate?

Example Scenario 1 – Authentication of NE to SDN controller


DNS
Ne1.domain1
domain1
Ne2.domain1
2- What is the TLSA and policy reference number of
3- Here you go yyyy
SDN controller (cert.DNS)
4- TLSA (xxxx)=yyyy
1- Traffic congestion (Openflow) My certificate is xxxx
NE1.domain1
NEn.domain1
NE2.domain1
Domain1

Example Scenario 2 – Authentication of a tenant to configure any network element



Current status of this work

- Presented in other standardization groups in ONF
- SDN is only an example scenario for this way of authentication and authorization → possible also to use it for NFV

Conclusion

- Need flexibility in multi-tenancy environment for SDN solution?

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

Thank you!