Implementing certificate based authentication on application level

29. Jan 2021

Tianzhe Yu

tianzhe.yu@ifak.eu

Introduction：Certificate Based Authentication

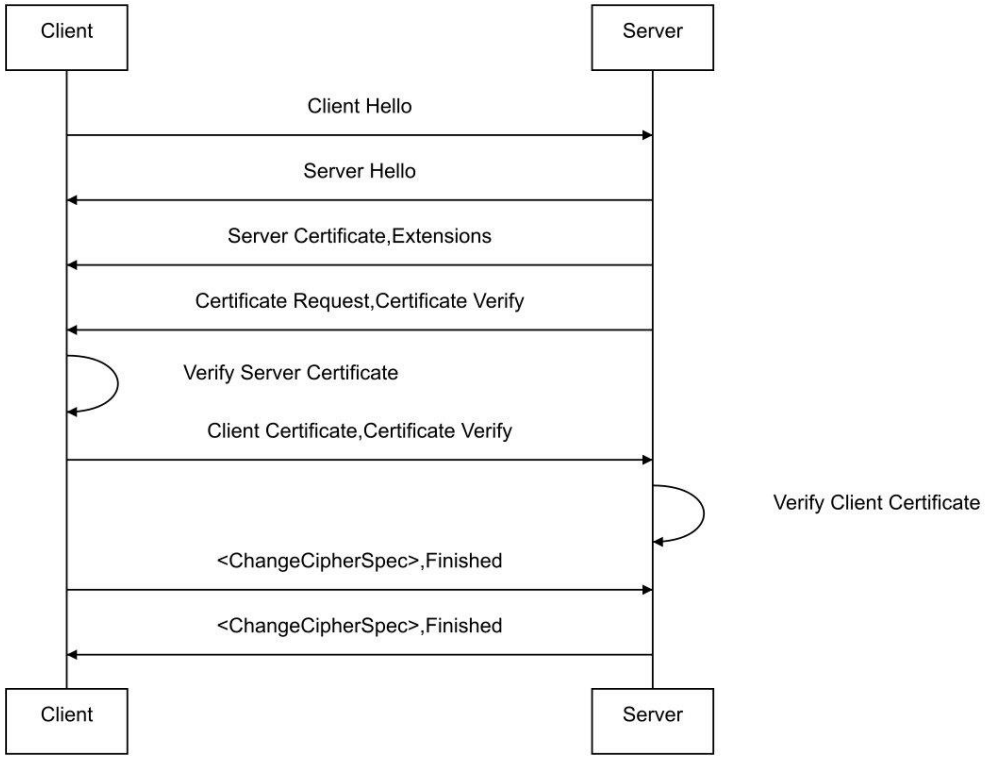


Fig.1 Message Flow of a TLS Connection

- Compare to Password based authentication, it is more secure and scalable
- Widely used in TLS, HTTPS and OAuth
- Certificate is uniquely bond to an ID and a key pair, protected with a digital signature
- Mutual-Authentication by verification of certificates and certificate verify messages

Motivation: Issues with TLS proxy

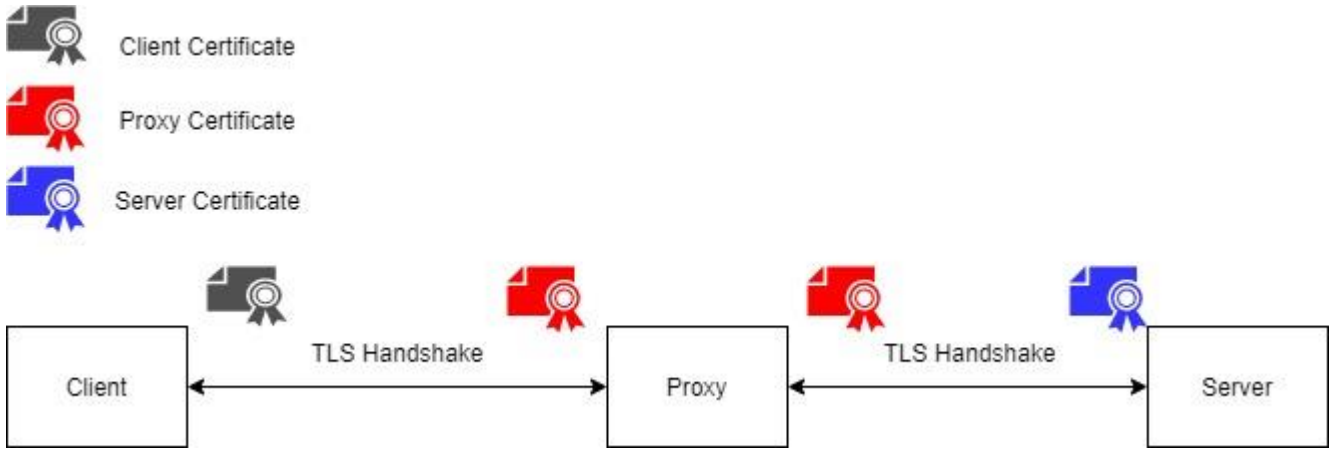


Fig.2 TLS Connection with a Proxy in the middle

- TLS proxy acts as a Man-in-the-Middle and uses its certificate to establish the communication channel with Sever
- This makes certificate based authentication unreliable

Proposed Solution: JWT + x5c

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsIng1dCI6InpIMTdZMWRVcUR5YkdfeFdqUnJMbWtvdXhORT0iLCJ4NWMiOlsiTUlJRHdqQ0NBcXFnQXdJQkFnSVVUVHI2U1d2TktvOTM2OXRjSmlKK1FWMlVKS0V3RFFZSktvWklodmNOQVFFTEJRQXdjakVMTUFrR0ExVUVCaE1DUjBJeEREQUtCZ05WQkFnTUEwRkNRekVNTUFvR0ExVUVCd3dEUVVKRE1SZ3dGZ1lEVlFRS0RBOUhiRzlpWVd3Z1UyVmpkWEpwZEhreEZqQVVCZ05WQkFzTURVbFVJRVJsY0dGeWRHMWxiblF4RlRBVEJnTlZCQU1NREhSbGMzUmZjbTl2ZEY5allUQWVGdzB5TURFeU16RXhNVEEyTXpOYUZ3MHlNekE1TWpjeE1UQTJNek5hTUhJeEN6QUpCZ05WQkFFZVEFrZENNUXd3Q2dZRFZRUUlEQU5CUWtNeEREQUtCZ05WQkFjTUEwRkNRekVNTUFvR0ExVUVDd3dDUjJ4dlltRnNNJRk5sWTNWeWFYUjVNUll3RkFZRFZRUUxEQTFKVkNCRVpYQmhjblJ0Wlc1ME1SVXdFd1lEVlFRRERBeDBaWE4wWDNKdmIzUmZZMkV3Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRRHZhaldvL3cvQk0zYzhuL25pazd0KzRPK0NzU1NqSHJ5RVFEWG80SUlieHB4QlU1YllSS2lqYUtORlWbW03SG9yN0gxRmMyaW5PQkp4RHE0RDFCMjdNV2J4ZHdHOTUrMi9LeHJjdUJxeC9PN0VXVW1Ea3JOQjE5TUJzeW83dVlzZm4xY3FHRUhSdklkakIxLTE1sRFNFZHRQdHMrSjhzSWRvR1ArRjgyeHdsMjNFL0RjTjd1NTFpa1ZIdlhhOHFrVEFSYUt5K1h1YXdVbkYyUURUQll6dm4zL1JzU1JsNXBOakRQRW5qRnJDckZmb0hSQW0xVTRRa0J5bFhnWGUweVZTNUI4ZUk2UjJUam5wNmU5U0g4OXFKQ3ByY2VLOWY4VlNmdjJicnZKRG9vRkh0QVNTbC9abjRWem1OUTFxdkw4dXIzZ2hBMjRFRE41cXQybjlrWlVIQWdNQkFBR2pVREJPTUIwR0ExVWREZ1FXQkJURUZBdHJNaTIvbmFWa2p5ZGNZWTNNK2dQVVlqQWZCZ05WSFNNRUdEQVdnQlRFRkF0ck1pMi9uYVZranlkY1lZM00rZ1BVWWpBTUJnTlZIUk1FQlRBREFRSC9NQTBHQ1NxR1NJYjNEUUVCQ3dVQUE0SUJBUUNDOFIrWmtIdHpHTVNmeTUyTndaMlJoUFdWK091T0dZZlNTR0ZmOE1pZjhkZnI3bFUvUHBkcHVxQmxOWmpSeXdNaGRpaFFHNFZzSVY3OHNZc1kzQkFTN1ZVQ3AyWk9rVVd6QTZOUEdPUVFsK05MY285ZnQ1YUIwRnE3anhwak9Rbk1LYzBkdW1IWEpIM21rSXZMV2tZRXhEN3FrTHhqeXBNbnlKQTR5ZzhhVUl0NHJXSUkrcFAvdUpHZFV4OGYybEVjQ1NUWmhINWhKbFJsTml1eG51dTZsQUl0RGFhVC9laGVRT0dyTnY5cXdnMVhEVlp4bm51aWpBQWJvb0p6Q0xmdnhONDNNMEJTR1hiRWVCamxreXgzTGNpdXNkeDlUQlkzTzNaUXA4THhKNlFHM0xyOG9kWkgyZjU3ZUlwR1lHQUJSUWtMUjQyaWVjUmZwNWxTeVNBciJdfQ.

eyJpc3MiOiAiY2xpZW50Lmp3dCIsICJzdWIiOiAiY2xpZW50Lmp3dCIsICJhdWQiOiAiaHR0cHM6Ly90YXJnZXR1cmwvZW5kcG9pbnQiLCAianRpIjogIjVkMTAxMWMxLTI4MDgtNDI0OC05MWFlLTQyOWY0MTgzMTA2ZiIsICJpYXQiOiAxNjA5NDE3MDI0LCAibmJmIjogMTYwOTQxNzAyNCwgImV4cCI6IDE2MDk0MTg4MjR9.

69kbLwwYh4EzHDTU2eZ3v1JNd6kSKPqqtkXAL325mcb5w6smWYDs5dthmYssdlm-hTG9nLcItvkWts4petbwi--DQMfsX2bxtRjBaxu6eU_-dx3GUnvLbsSWq0W80_XHbvSiVAnBGlj4mEw1p_uEGgRorh6jwr46SjsF0UnIya_qVX4F0fA3u1dSTOERCf1AYxtd1ud5lGT3JPHiFSeCqGu_bEugmXZYilPSKkjy09irPKK65a6isCJdMf8PkmLMD01hdncBbVMt_S3PYHruxrprZHxKhLBQtqGrGf5BzQh4cVNWLYS-axuj4WhTV-_3ozECS0q6_fK6EVMYoR8r6A

Header

```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5t": "zH17Y1dUqDybG_xWjRrLmkouxNE=",
  "x5c": [
    Certificates in BASE64 encode
  ]
}
```

Body

```
{
  "iss": "client.jwt",
  "sub": "client.jwt",
  "aud": "https://targeturl/endpoint",
  "jti": "5d1011c1-2808-4248-91ae-429f4183106f",
  "iat": 1609417024,
  "nbf": 1609417024,
  "exp": 1609418824}
```

Signature

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload), PrivateKey
)
```

Field	Description
Header	
alg	The algorithm used which is used to sign the JWT
typ	Type of token, in this case “JWT”
x5c	X.509 Certificate Chain. It should be an JSON array of certificate chain presented as base64 encoded strings from leaf certificate to root certificate
x5t	Base64url-encoded SHA-1 thumbprint of leaf certificate
Payload	
jti	A unique identifier of JWT
sub	Subject, should be client_id
iat	Issued at, the time stamp then the token is created
nbf	Not valid before
exp	Expiration time
iss	Issuer of the token
aud	Audience, normally the endpoint URL of authentication server
Signature	
Sign (base64UrlEncode(header) + "." + base64UrlEncode(payload), privateKey bond with end certificate in x5c , alg)	

Table1. Details of proposed JWT

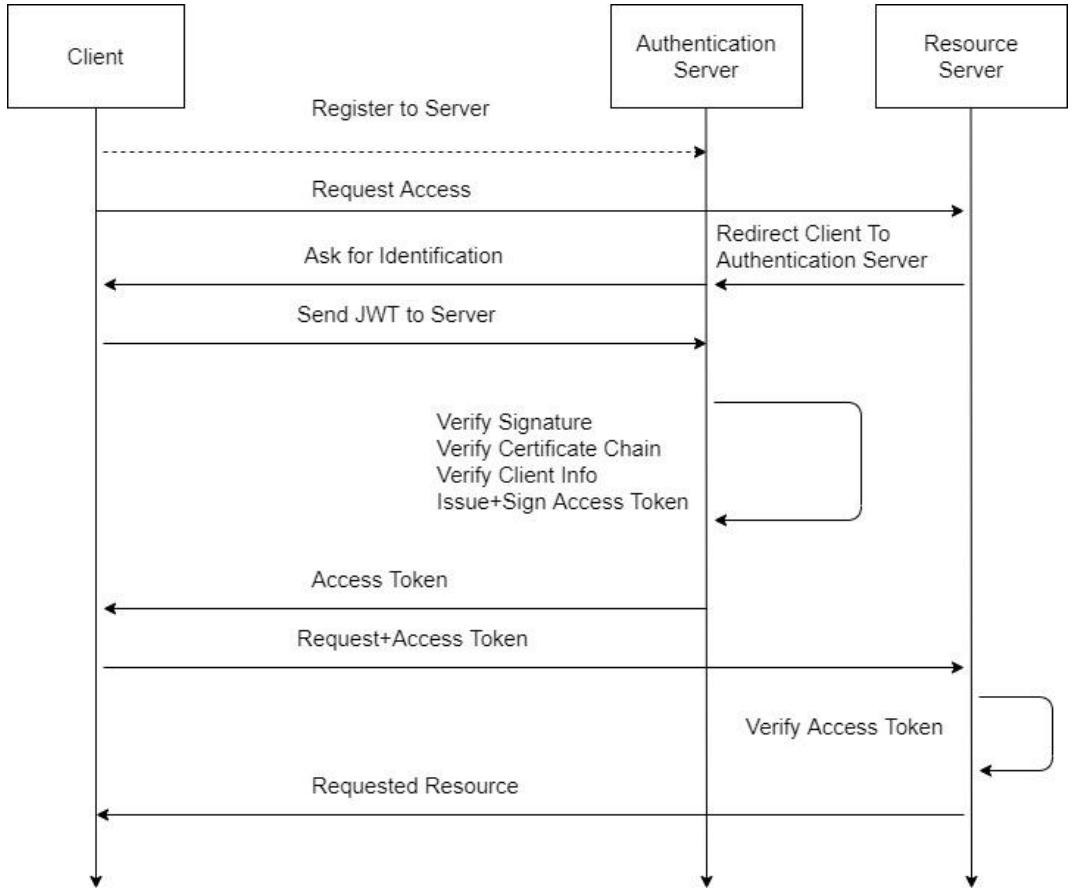
Mapping to mutual TLS

Mutual TLS	Proposed Solution	Usage
Hello	Alg,typ	Define used algorithms
Certificate Chain	X5C	Authentication
Certificate Verify	jti, iat,aud Signature	Authentication& Integrity
Extension	Aud, Body	Other Infos

Table2. Mapping Proposed Solution to TLS

- Bring certificate authentication to application level
- Compatible with proxies & other transport protocols
- Using of JWT is friendly to applications & developers

Demonstration: How it works with application



1. Client and root certificate need to be registered to Server e.g. through an out of band channel
2. When a client wants to get access to resource, it will be redirected to an authentication server
3. Client generates and signs a JWT when it is asked to present its identification
4. Authentication server verifies signature of JWT, certificate chain contains in the x5c, time stamp and client info
5. When verified, authentication server may issue an access token in JWT format, signed with authentication server's certificate
6. Client presents the access token to resource server
7. Resource server verifies the access token and grant client access.

Fig.3 Flaw chart of proposed solution

Links:

AASX Package Explorer with OpenID Client

- https://github.com/admin-shell-io/aasx-package-explorer/releases
- Connect to REST „openid3“

Python Script (Branch tianzhe)

- https://gitlab.com/TZYu30/pyaassecurity/-/tree/tianzhe/tokenRequester

JWT parser

- https://jwt.io/

OpenID Server

- https://admin-shell-io.com:50001/.well-known/openid-configuration
- https://github.com/admin-shell-io/aasx-IdentityServer4

AASX Server

- https://admin-shell-io.com:5101/

ASSX Server Data (with test token as query parameter)

- https://admin-shell-io.com:51311/aas/CPE18/submodels/Identification/table?eyJhbGciOiJSUzI1NiIsImtpZCI6IkJENTVCQ0RGRDdEQzQzQTZCQUNENDI2RTZFQzFFMThBRUMzQ0UzNzVSUzI1NiIsInR5cCI6ImF0K2p3dCIsIng1dCI6InZWVzgzOWZjUTZhNnpVSnVic0hoaXV3ODQzVSJ9.eyJuYmYiOjE1ODY3OTMxOTYsImV4cCI6MTU4Njc5Njc5NiwiaXNzIjoiaHR0cHM6Ly9oMjg0MTM0NS5zdHJhdG9zZXJ2ZXIubmV0OjUwMDAxIiwiYXVkIjoicmVzb3VyY2UxIiwiY2xpZW50X2lkIjoiY2xpZW50Lmp3dCIsInVzZXJOYW1lIjoiYW9yemVsc2tpQHBob2VuaXhjb250YWN0LmNvbSIsInNlcnZlck5hbWUiOiJpZGVudGl0eXNlcnZlci50ZXN0LnJzYSIsInNjb3BlIjpbInNjb3BlMSJdfQ.YYYtlKab6PZfVAI0qESNNp-jeRrILs0ZMNuxmCsLWKLJhSrMeDS2N5V5q9U7S4xBWLUomIdfEV0HFgELfg2NY36Dd6EUSQ7qiPQHIGJZBnFmHI4UlVdd3Hoa_yjYu3OQ6ZnMJSwH0cIxcijIcXs__VB_traym-JzTssSM_3LGTvtfKfn4oIvMto_nzJgJtSrW5_K_S4XNX_Zi85sQE3tg9SZkxiXyvRl81kIk5gaFYdpjqM4pgRqW55jVUqRst6sAYyuCziK-OSfI_575D6GQTaZAf_vKl8I2tP-EMqVRkfo5xU2PaA1gELFtS0NuDmzWmxhupYbRNKBNH7J8bXxZw

More

- http://admin-shell-io.com/screencasts/

Thank You