

RAPPORT TRAVAUX PRATIQUE – SUJET 4

Authentification et Autorisation

Étape 0 – état des lieux

Logiciels utilisés :

- PhpStorm
- Postman

Récupération du code fourni sur GitHub avec les liens suivants :

<https://github.com/laurentgiustignano/R6.A.05-TP-Secu1>

<https://github.com/laurentgiustignano/R6.A.05-TP-Secu1-auth>

<https://github.com/laurentgiustignano/R6.A.05-TP-Secu1-data>

Teste du code effectuer sur Postman avec les liens suivants :

<http://localhost:3000/secu>

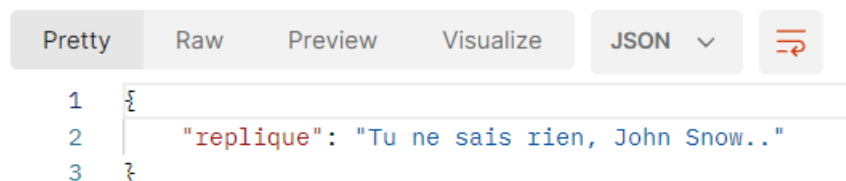
<http://localhost:3000/dmz>.



```

1  {
2    "replique": "Ca pourrait être mieux protégé..."
3  }
    
```

Body Cookies Headers (6) Test Results



```

1  {
2    "replique": "Tu ne sais rien, John Snow.."
3  }
    
```

Étape 1 – « De base... »

Résultat obtenu après avoir suivi les différentes étapes :

<input checked="" type="checkbox"/>	Authorization	Basic VHlyaW9uOndpbmU
	Key	Value

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "replique": "Un Lannister paye toujours ses dettes !"
3  }
  
```

Étape 2 – Prouves qui tu es !

Cette commande me sert à ouvrir mon server openssl

openssl s_server -accept 4567 -cert server.crt -key server.key -www -state

GET https://localhost:4567

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	Key	Value
	Key	Value

body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize HTML

```

1  <HTML><BODY BGCOLOR="#ffffff">
2  <pre>
3
4  s_server -accept 4567 -cert server.crt -key server.key -www -state
5  This TLS version forbids renegotiation.
6  Ciphers supported in s_server binary
7  TLSv1.3 :TLS_AES_256_GCM_SHA384 TLSv1.3 :TLS_CHACHA20_POLY1305_SHA256
8  TLSv1.3 :TLS_AES_128_GCM_SHA256 TLSv1.2 :ECDHE-ECDSA-AES256-GCM-SHA384
9  TLSv1.2 :ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 :DHE-RSA-AES256-GCM-SHA384
10 TLSv1.2 :ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 :ECDHE-RSA-CHACHA20-POLY1305
11 TLSv1.2 :DHE-RSA-CHACHA20-POLY1305 TLSv1.2 :ECDHE-ECDSA-AES128-GCM-SHA256
12 TLSv1.2 :ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 :DHE-RSA-AES128-GCM-SHA256
13 TLSv1.2 :ECDHE-ECDSA-AES256-SHA384 TLSv1.2 :ECDHE-RSA-AES256-SHA384
14 TLSv1.2 :DHE-RSA-AES256-SHA256 TLSv1.2 :ECDHE-ECDSA-AES128-SHA256
15 TLSv1.2 :ECDHE-RSA-AES128-SHA256 TLSv1.2 :DHE-RSA-AES128-SHA256
16 TLSv1.0 :ECDHE-ECDSA-AES256-SHA TLSv1.0 :ECDHE-RSA-AES256-SHA
17 SSLv3 :DHE-RSA-AES256-SHA TLSv1.0 :ECDHE-ECDSA-AES128-SHA
18 TLSv1.0 :ECDHE-RSA-AES128-SHA SSLv3 :DHE-RSA-AES128-SHA
19 TLSv1.2 :RSA-PSK-AES256-GCM-SHA384 TLSv1.2 :DHE-PSK-AES256-GCM-SHA384
20 TLSv1.2 :RSA-PSK-CHACHA20-POLY1305 TLSv1.2 :DHE-PSK-CHACHA20-POLY1305
  
```

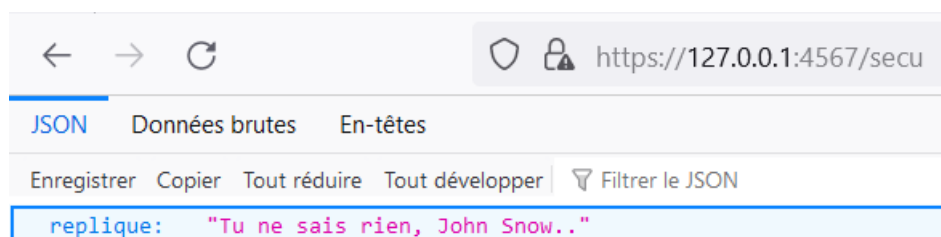
```

▼ GET https://localhost:4567/
Warning: Self signed certificate

▼ Network
  ▶ addresses: {...}
  ▼ tls: {...}
    reused: false
    authorized: false
    authorizationError: "DEPTH_ZERO_SELF_SIGNED_CERT"
  ▶ cipher: {...}
    protocol: "TLSv1.3"
    ephemeralKeyInfo: {}
  ▼ peerCertificate: {...}
    ▶ subject: {...}
    ▼ issuer: {...}
      country: "AU"
      stateOrProvince: "Paris"
      locality: "Paris, Paris"
      organization: "Paris, But"
      organizationalUnit: "But"
      commonName: "quentin"
    validFrom: "Feb 15 15:59:26 2024 GMT"
    validTo: "Feb 12 15:59:26 2034 GMT"
    fingerprint: "7A:3E:44:72:3E:68:99:F7:8E:1E:E3:70:8A:F3:0F:98:A1:6B:D1:C2"
    serialNumber: "42775ffbb325b6098794acca8d82d4abc18b2955"
  ▼ Request Headers

```

Je suis bien en https :



Étape 3 – Un jeton dans la machine

Je n'ai pas réussi à faire cette étape.