* Cryptography related issues
(Testing for weak cryptography)

1. Weak ssl/tls protocols, ciphers, keys and insufficent transport layer protection

- Common issues:
  1. ssl/tls protocol, ciphers, keys and renegotain must be properly configured.
  2. certificate validty must be ensured
  3. Software exposed must be updated due to possibility of known vulnerabilities.
  4. usage of secure flag for session cookie.
  5. HSTS
  6. presence of both http and https.
  7. presence of mixed https and http content in the same page, which can be used to leak information.
  8. sensitive data transmitted in clear text.

1. ssl/tls protocol, ciphers, keys and renegotain must be properly configured.
   - ① weak protocols must be disabled.
     (ssl v2 is enabled then there is a vulnerabilities)
   - ② if reneogition is possible then there is a vulnerabilities.
     (it should be disabled)
   - ③ if rsa or dsa is used the key must be at least 1024 bits encrypted
     (if not then there is a vulnerability)
   - ④ key must be generated with proper entropy.
   - ⑤ md5/rc4 should not be in used
   - ⑥ server should be protected from beast as well as CRIME attack.

✗ with nmap you can recongition ssl service.

```
> nmap -sV --reason -PN -n --top-ports 100 target-ip
```

✗ Checking for certificate information, weak ciphers and sslv2 via nmap

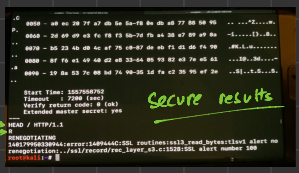
```
> nmap --script ssl-cert,ssl-enum-ciphers -p 443,465 target-ip
```

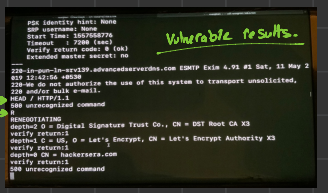
* checking for client-initiated renegotiation and secure renegotiation via openssl

> openssl s_client -connect target:443

HEAD / HTTP/1.1

R



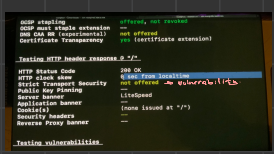


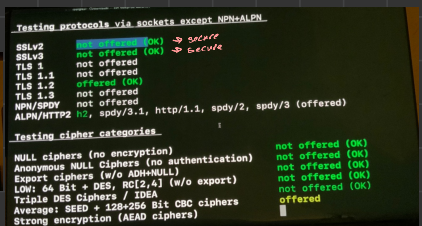
* Testing ssl/tls vulnerabilities with sslyze script

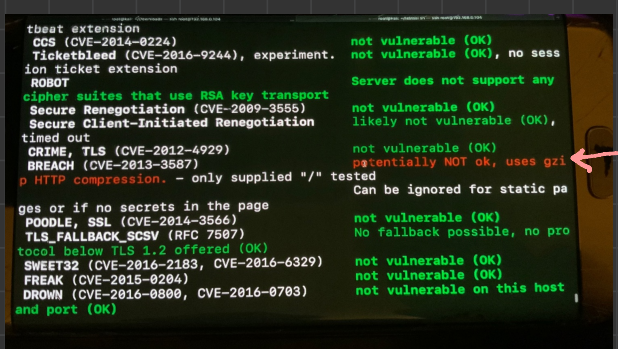
> sslyze --regular target:443

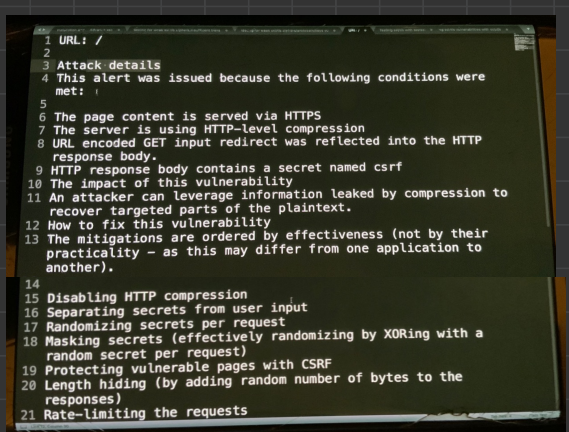
* Testing ssl/tls vulnerabilities with testssl script

> bash testssl.sh target









X SSl cookie without secure flag.

① Sign up in any application and app will generate a cookie
② look in request body for cookie section, and there should be a secure flag. (parameter)

* poodle attack (SSL v3 attack)

an attacker may be able to exploit this proble to conduct mitm attack and decrypt communication between the affected services and client.

```
>nmap -sV --version-light -Pn --script ssl-poodle -p 443 -iL Subdomains.txt
```

25, 465, 587, 993, 995, 21

```
PORT    STATE SERVICE       VERSION
443/tcp open  ssl/http-proxy Varnish http accelerator
| ssl-poodle:
|   VULNERABLE:                ← Vulnerable results
|   SSL POODLE information leak
|     State: VULNERABLE
|     IDs:  OSVDB:113251  CVE:CVE-2014-3566
|           The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and
other
|           products, uses nondeterministic CBC padding, which makes it
easier
|           for man-in-the-middle attackers to obtain cleartext data vi
a a
|           padding-oracle attack, aka the "POODLE" issue.
|     Disclosure date: 2014-10-14
|     Check results:
|       TLS_RSA_WITH_AES_128_CBC_SHA
|     References:
|       http://osvdb.org/113251
|       https://www.openssl.org/~bodo/ssl-poodle.pdf
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
```

* Enumerate cipher

```
>nmap --script ssl-enum-ciphers -p 443 target
```

```
Host is up (0.16s latency).

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   SSLv3:
|     ciphers:
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: server
|     warnings:
|       CBC-mode cipher in SSLv3 (CVE-2014-3566)
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     compressors:
```

```
> curl -v3 -X Head target → to test
```

or you can use SSLlabs.com

```
Nmap done: 1 IP address (1 host up) scanned in 18.15 seconds
root@kali:~# curl -v3 -X HEAD https://home.tomtom.com
Warning: Setting custom HTTP method to HEAD with -X/--request may no
t work the
Warning: way you want. Consider using -I/--head instead.
*   Trying 185.5.121.54...
* TCP_NODELAY set
* Connected to home.tomtom.com (185.5.121.54) port 443 (#0)
* OpenSSL was built without SSLv3 support  → it is vulnerable
* Closing connection 0
curl: (4) OpenSSL was built without SSLv3 support
root@kali:~#
```

2. Oracle padding
3. Testing for senstive informations sent via unencrypted channel.