Cheatography

HTTPS and Testing Encryption Cheat Sheet

by binca via cheatography.com/44948/cs/13306/

HTTPS: The Basics

SSL/TLS are 2 common options for encrypting HTTP

TLS adds more options for encryption and hashing including 2 different hashes

Secures in transit, but not on either end

Relies on Public Key Infrastructure (PKI) and trust in Certificate Authorities (CA)

When the web browser requests an HTTPS page from the web server, it receives the server's public key. The browser trusts this key because it is signed by the CA.

HTTPS: Attacker Perspective

Prevents listening and manipulating in transit

If control of either side of the tunnel (server or browser); we can decrypt the variable with stolen keys from the server or alter the variables at the browser

Can be used to hide attacker's traffic from NIDS, unless they are configured to perform on-the-fly decryptions, which is unusual due to performance implications

Note: HTTPS accelerators allow HTTPS to terminate on a network device prior to the data reaching the web server allowing for performance gain and enabling IDS to read the traffic.

Testing Weak Ciphers

Does the server support HTTPS and which versions are supported?	SSLv2, SSLv3, TLS1, TLS1.1, TLS1.2
Which ciphers are use and what are the key lengths?	NULL ciphers and lower encryption levels are either weak or plain text

Does the app allow HTTP access to resources that should be protected by HTTPS?

IS the certficate expired or considered invalid by the browser?

Note: Anything below TLS1.2 is older and has issues

OpenSSL

Enables us to generate, sign, manage, and validate certificates as well as make SSL connection directly.

Provides similar acces to SSL that Telnet and Netcat provide to clear text services.

Can test a server configuration and is often already installed.

Test for SSLv2: `$ openssl s_client -connect domain:443 -ssl2`

Test for NULL Cipher: `$ openssl s_client -connect domain:443 -cipher NULL`

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

Nmap NSE to Evaluate Ciphers

Nmap NSE script ssl-enum-ciphers evaluates ciphers supported by an HTTPS server

Categorizes cipher strengths with letter grades A through F

```
$ nmap -p 443 --scri pt= ssl -en um- ciphers domain
```

Qualys SSL Labs

Free, publicly accessible site that will provide a letter grade based on the security of a submitted domain.

Goes beyond the SSL version and basic estimation of the cipher strength.

Similar to SSLDigger.

HTTPS Support on Targets

Strength and "correct" HTTPS support will vary with the needs of the site.

Expired, bad, or other certificate errors should be reported

SSLv3 or earlier, levels of encryption < 128-bits, and weak hashing algorithms like MD5 or SHA-1

Heartbleed (CVE-2014-0160)

OpenSSL vulnerability publicly discovered in 04/2014, unpatched from 03/2012-04/2012

Affected OpenSSL versions 1.0.1 - 1.0.1f, and 1.0.2-beta1

Allows remote reading of 64KB memory chunks directly from a vulnerable OpenSSL server with repeated attempts exposing different chunks of RAM

Nothing is logged on the web server from the attack

Can find usernames, passwords, cookies and more in RAM

CloudFlare ran a challenge to see if the private key could be stolen from a vulnerable server, four people succeeded on the 1st day

Test for presence of heartbleed with **Nmap** `$ nmap -p 443 --scr ipt ssl-he art bleed domain`

Exploit the vulnerability with SensePost's **heartbleed.py** script, which creates a dump file called "dump.bin" by default containing a binary copy of all dumped memory.

Named for the TLS heartbeet extension (RFC 6520), which provides allowed the usage of keep-alive functionality without performing a renegotiation and is the basis for PMTU discovery for DTLS.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com