

Greffe de cœur pour OpenSSL

JSSI

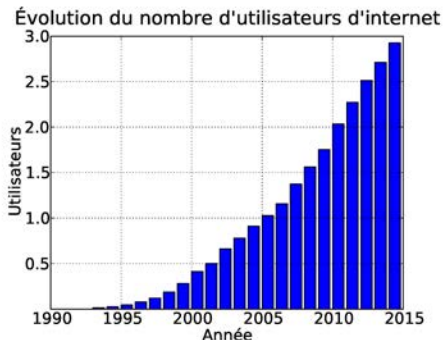
Ministère de la Défense

13 Novembre 2014

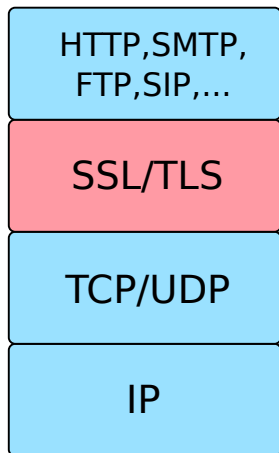


Évolution de l'exposition des utilisateurs

- Trafic total : ~ 24 To/s
- Utilisateurs de plus en plus exposés :
 - ▶ Transactions financières
 - ▶ Mails
 - ▶ VoIP
 - ▶ Vie privée



SSL/TLS



- Authentification
- Confidentialité
- Intégrité



OpenSSL™

Cryptography and SSL/TLS Toolkit



OpenSSL : petit rappel...

- Création : 23/12/98
- Bibliothèque cryptographique + implémentation SSL/TLS
- Open source - communauté mondiale
- Versions Unix/Linux, Microsoft Windows, OpenVMS
- Certification : FIPS 140-2
- Utilisation : Apache + Nginx = 66% du marché

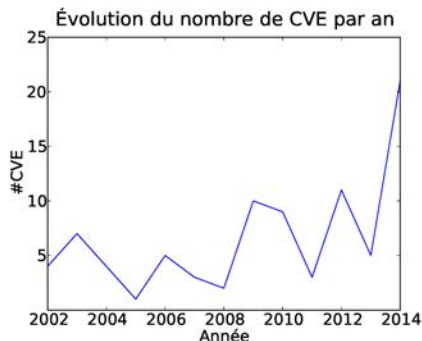


La sécu dans tout ça



Quelques petites failles...

- Timing attacks
 - ▶ Clés privées RSA - 2003
 - ▶ Clés symétriques - 2013
- Deni de service
 - ▶ Parsing ASN.1 - 2003
 - ▶ OCSP stapling - 2011
- Affaiblissement cryptographique
 - ▶ Debian - 2008
 - ▶ Injection CCS - 2014



Heartbleed

OpenSSL Heartbleed Zero-day vulnerability



Heartbeat

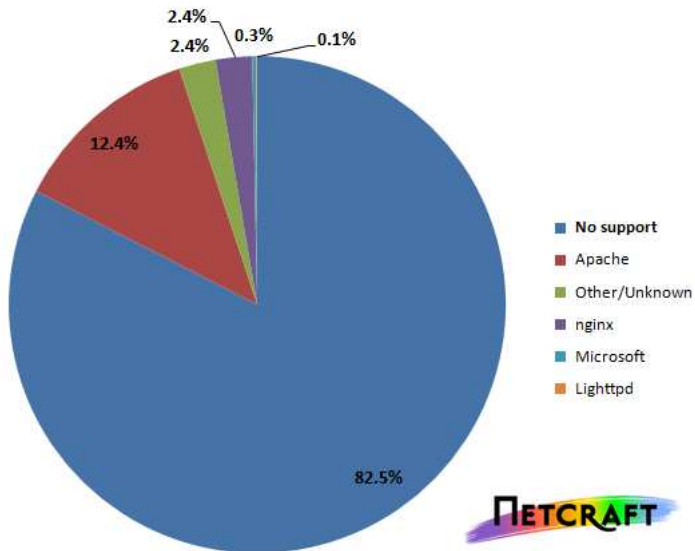


- DTLS : TLS sur UDP
- RFC 6520 : TLS/DTLS Heartbeat Extension
- Objectifs :
 - ▶ Éviter la renégociation : keep-alive
 - ▶ Path MTU discovery pour le DTLS

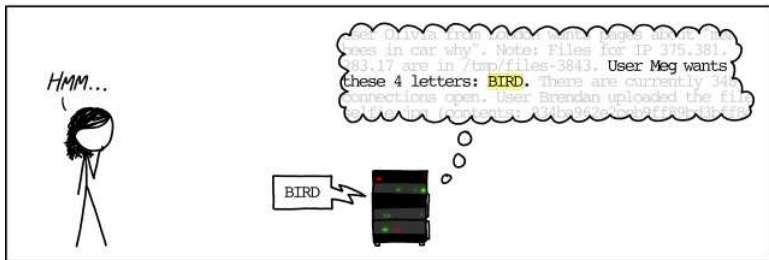
Type (1 octet)	Taille du payload (2 octets)	payload	padding
-------------------	---------------------------------	---------	---------



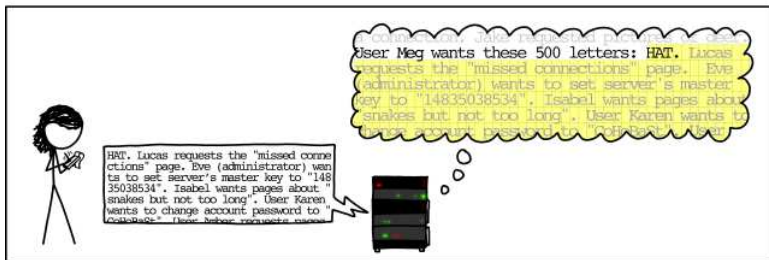
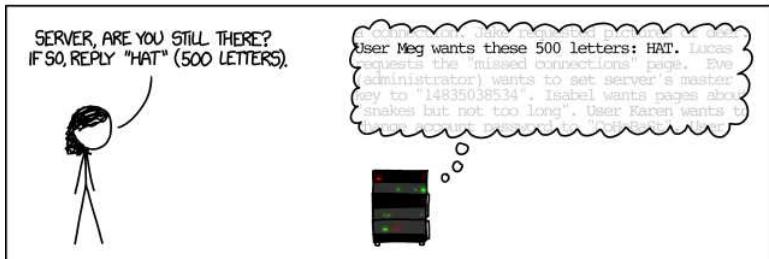
TLS Heartbeat Extension Support by IP Address



Heartbleed expliqué par xkcd



Heartbleed expliqué par xkcd



Le petit bug

```
/* Read type and payload length first */
```

```
hbtype = *p++;  
n2s(p, payload);
```

```
pl = p;
```



Le petit bug

```
/* Read type and payload length first */
```

```
if (1 + 2 + 16 > s->s3->rrec.length)  
    return 0; /* silently discard */
```

```
hbtype = *p++;  
n2s(p, payload);
```

```
if (1 + 2 + payload + 16 > s->s3->rrec.length)  
    return 0; /* silently discard per RFC 6520 */
```

```
pl = p;
```



- Codenomicon et Google
- Fuzzing : Codenomicon's Defensic SafeGuard

FUZZING 101

§ CODENOMICON

- `openssl-1.0.1[a-f]`

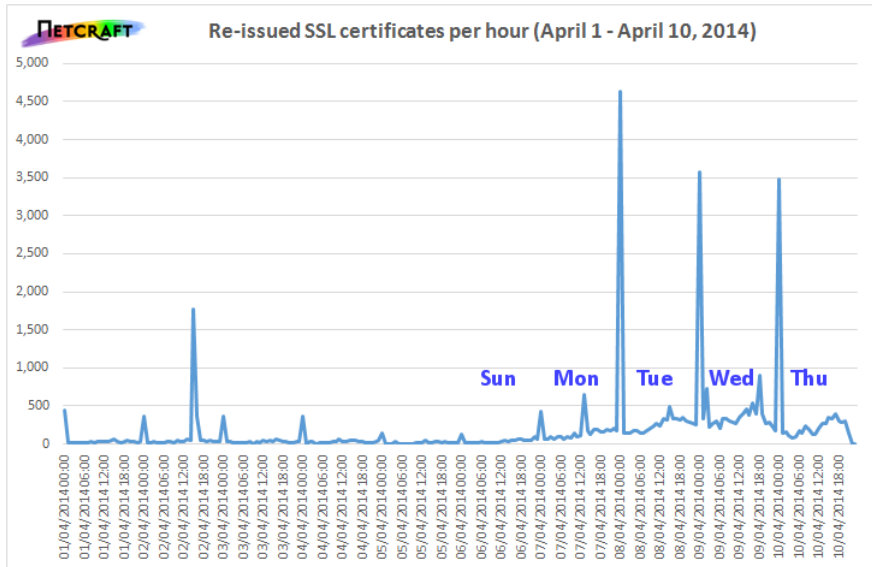


Chronologie des événements

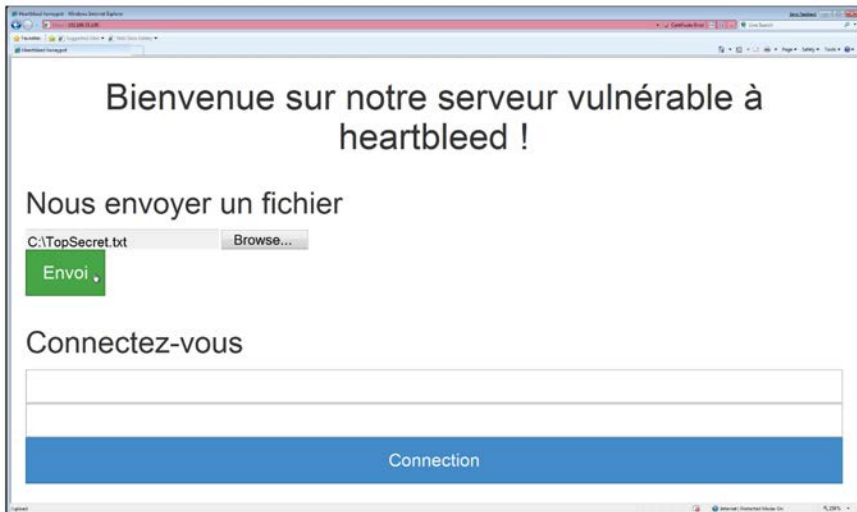
- 18 Juin 2010 Draft RFC 6520
- 31 Décembre 2011 Rajout du code pour Heartbeat
- 14 Mars 2012 OpenSSL 1.0.1
- 3 Décembre 2013 Réservation du CVE
- 5 Avril 2014 Heartbleed.com enregistré
- 7 Avril 2014 Annonce de la faille, nouvelle version d'OpenSSL
- 8 Avril 2014 RHEL, CentOS, Ubuntu, Debian, etc
- 11 Avril 2014 Vol de la clé privée de Cloudflare
- 17 Juin 2014 Sites populaires encore exposés $> 10^4$



Ré-émission de certificats début Avril



Exploitation : envoi de fichier



Exploitation : envoi de fichier

The image shows a Wireshark capture of a TLSv1 session. The packet list pane shows several packets, with packet 9 selected. The packet details pane shows the structure of the TLSv1 record, including the Application Data protocol. The packet bytes pane shows the raw hex and ASCII data.

No.	Time	Source	Destination	Protocol	Length	Info
4.0.000772000		192.168.35.1	192.168.35.130	TLSv1		438 Client Hello
6.0.002752000		192.168.35.130	192.168.35.1	TLSv1		211 Server Hello, Change Cipher Spec, Encrypted Handshake Message
8.0.003493000		192.168.35.1	192.168.35.130	TLSv1		125 Change Cipher Spec, Encrypted Handshake Message
9.0.008363000		192.168.35.1	192.168.35.130	TLSv1		1514 Application Data
10.0.008386000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]
11.0.008395000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]
12.0.008402000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]
13.0.008413000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]
16.0.008818000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]
17.0.008840000		192.168.35.1	192.168.35.130	TCP		1514 [TCP segment of a reassembled PDU]

Frame 9: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: Vmware c8:00:01 (08:50:56:c8:00:01), Dst: CadmusCo 0d:33:be (08:00:27:0d:33:be)
Internet Protocol Version 4, Src: 192.168.35.1 (192.168.35.1), Dst: 192.168.35.130 (192.168.35.130)
Transmission Control Protocol, Src Port: 52385 (52385), Dst Port: https (443), Seq: 432, Ack: 146, Len: 1448

Secure Sockets Layer
- TLSv1 Record Layer: Application Data Protocol: http

```
0000 08 00 27 0d 33 be 00 50 56 c8 00 01 08 00 45 00 ..^..P V.....E.  
0010 05 dc ff 63 40 00 40 06 6d e4 c0 a8 23 01 c0 a8 ...C0.@. m...#...  
0020 23 82 cc a1 01 bb 3c 93 ed 95 f5 83 0d d5 80 10 #.....<.....  
0030 00 7b c4 45 00 00 01 01 08 0a 00 0d 14 bd 00 06 {...E.....  
0040 e9 d2 17 03 01 00 20 fa 05 29 33 f5 c7 91 32 bc .....<.....}3..2.  
0050 73 de 07 cb 42 41 fb 1b b3 fd 50 a0 b5 62 bf 43 s...BA...P..b.C  
0060 ae 85 14 65 b4 49 e1 17 03 01 40 20 10 ec 97 35 ...e.I...@...5  
0070 93 c2 4b 14 09 a0 a3 b6 84 c0 87 fe e6 0d 47 6a ..R.....<.....G}  
0080 7a 6e a2 0f 7c 05 d6 ce 5e 8a 8a 6b 0e 00 a8 93 ..T.....q*..I.  
0090 e0 c0 cf 54 e2 c9 c4 94 be 71 2a c4 71 54 21 a5 ...T.....q*..I.  
00a0 ac 65 86 4e 29 59 81 2e 2a 25 db 93 f3 dc e5 be ..e.N)Y...%.....  
00b0 41 4b de 1a 82 44 ee bc 5f 27 8e ec 1e bf a5 ca AK...D...<.....  
00c0 13 75 8b e0 b4 6c 10 f8 7b 27 22 fd 09 26 f4 86 ..u...l...{...<.....6..  
00d0 7f 3c 54 1f a9 0f 2a 49 6c fe 5c ba 8a e1 ad 8e <T...<I l\.....  
00e0 45 a9 4f 06 a0 c1 dd cf b2 60 ab 53 c0 ec 5b 85 E.O.....<S...{.  
00f0 5e 94 b8 14 11 17 12 bf 19 3e f4 9b 7d c3 1c 0d ^.....<...>.....  
0100 d5 a0 2e c3 76 57 8f 2c 5f ff d5 52 ce d5 86 89 .....VM...<R.....  
0110 7d ab f9 ad fc a2 fe 0d 3a a4 77 29 05 d0 f2 68 }.....<:w...rh  
0120 a4 1e bb 0e 0e f2 52 08 d5 01 c7 67 74 11 f2 55 .....R...<gt...U  
0130 16 3e 93 b4 46 bb f2 d7 73 02 69 0e 22 d8 a0 cf >..F...<S...i...  
0140 40 3d 03 8e 25 30 bf 13 1a 94 85 16 f3 10 4c 66 @...<N0...<LT  
8150 40 32 0e 29 5c cf 82 ce 5c 70 7c 76 07 74 4c 77 # 9 <...<A
```



Exploitation : envoi de fichier

The screenshot shows a Wireshark capture of a TLSv1.1 connection. The packet list pane shows several packets, with packet 104 selected, which is a continuation of data (2962 bytes) from the previous packet. The packet bytes pane shows the raw data in hex and ASCII. A red box highlights the text "TOPSECRET" in the ASCII view, and a red arrow points from this box to a callout box on the right.

==== TOPSECRET ====
Les habitants de Rouen utilisent OpenSSL, c'est normal, ils aiment la crypto. Ils aiment aussi le brouillard et la pluie.



Exploitation : identifiants

Bienvenue sur notre serveur vulnérable à heartbleed !

Nous envoyer un fichier

Browse...

Envoi

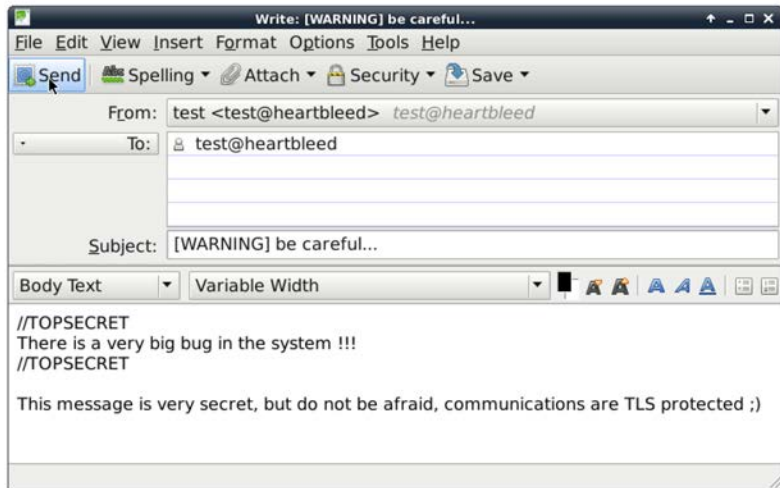
Connectez-vous

.....

Connection



Exploitation : mails (SMTPS)



Exploitation : mails (SMTPS)

The image shows a Wireshark capture of an SMTPS session. The packet list pane shows several TLS-related packets, with packet 41 selected. The packet details pane shows the transmission control protocol (TCP) and internet protocol (IP) headers. The packet bytes pane shows the raw data of the selected packet, which is a TLS message. The message content is displayed in a hex dump and ASCII view. The ASCII view shows the following text:

```
.....  
..chase t=ISO-88  
59-l; fo rmat=flo  
wed..Con tent-Trans  
nsfer-En coding:  
7bit..... //TOPSEC  
RET..The re is a  
very big bug in  
the syst em !!!  
//TOPSEC RET...T  
his mess age is v  
ery secr et, but  
do not b e afraid  
, comm unications  
are ..T LS prote  
cted ;). .....[  
Q.A.5w. .5..u.H
```

An arrow points from the text in the ASCII view to a red speech bubble containing the following text:

```
//TOPSECRET  
There is a very big bug in the system !!!  
//TOPSECRET  
This message is very secret,  
but do not be afraid,  
communications are TLS protected ;)
```



Exploitation : clé privée du serveur

- RSA :
 - ▶ On récupère le certificat et donc $N = pq$
 - ▶ Pour chaque bloc P de $|N|/2$ octets consécutifs, on regarde si $P|N$
 - ECDSA :
 - ▶ On récupère le certificat et donc $D_s = [d_s].G$
 - ▶ Pour chaque bloc d de $|d_s|$ octets consécutifs, on regarde si $[d].G = D_s$
- Grand nombre de réponses nécessaires (plusieurs Go)
 - "Stresser" le serveur



Impact

Quoi ?

- Serveurs (HTTP, FTP, SMTP, ...)
- VPN
- TOR : 380 noeuds bloqués (12%)
- Applications mobiles
- Set top boxes

Qui ?

- ~ 17% des 10000 plus gros sites web
- ~ 500000 sites en tout
- Facebook, Google, Yahoo, Wikipedia, etc



Que faire ?

Utilisateur d'un site affecté

- ① Une fois le site sûr : changer de mot de passe
- ② Mettre à jour sa liste de révocation de certificats

Entreprise

- ① Mise à jour
- ② Ré-émission du certificat
- ③ Imposer le changement de mot de passe pour tous les utilisateurs



À l'avenir

- Analyse de code : clint, valgrind
- Tests :
 - ▶ Unitaires
 - ▶ Fuzzing
- Plus de relecteurs
- Aider la communauté OpenSSL : 1300 bugs ouverts, 500000 lignes
- Réduire la surface d'attaque, enlever des fonctionnalités
- LibreSSL, BoringSSL, polarSSL

