Heartbleed Debugged – OpenSSL & CVE-2014-0160

René 'Lynx' Pfeiffer

DeepSec GmbH
https://deepsec.net/, rpfeiffer@deepsec.net

FH Eisenstadt

DEEPSEC

Vorstellung

- Studium der Physik
- PGP Benutzer seit 1992 (aus denselben Gründen wie heute)
- selbstständig seit 1999 in der Informationstechnologie
- seit 2001 Lehrtätigkeit bei Firmen und dem Technikum Wien
- seit 2008 in der Geschäftsführung der DeepSec Konferenz
- seit 2010 in der Geschäftsführung der Crowes Agency OG

Motivation – Ist OpenSSL wichtig?

- OpenSSL ist wichtige Sicherheitskomponente
- weit verbreitet in Applikationen & Embedded Devices
- OpenSSL implementiert nicht nur SSL/TLS
  - *full-strength general purpose cryptography library*
  - X.509 für PKI
  - ...
- Fehler in OpenSSL betreffen daher viele Anwendungen
  One Bug to rule them all, One Bug to find them; One Bug to bring them all and in the darkness bind them.

Geschichte von OpenSSL

- offizieller Start am 23. Dezember 1998 (Version 0.9.1)
- Motivation: Werkzeug für Kryptographie, Fokus Internet
- `mod_ssl v1` stammt von April 1998 (auf Basis Apache-SSL)
- OpenSSL Entwicklungsteam
  - 11 Developer (zwei angestellt)
  - schreiben, modifizieren, testen 500.000+ Zeilen Code

OpenSSL Fähigkeiten

- Verschlüsselungsalgorithmen
  AES, Blowfish, Camellia, SEED, CAST-128, DES, IDEA, RC2, RC4, RC5, Triple DES, GOST 28147-89
- Hash-Algorithmen
  MD5, MD4, MD2, SHA-1, SHA-2, RIPEMD-160, MDC-2, GOST R 34.11-94
- Public Key Verfahren
  RSA, DSA, Diffie–Hellman Schlüsseltausch, Elliptic curve, GOST R 34.10-2001
- minimale Certificate Authority (CA) Implementation
- S/MIME E-Mail Verarbeitung

SSL/TLS Überblick

- Secure Sockets Layer (SSL) von Netscape erfunden (1995)
- Transport Layer Security (TLS) in IETF Nachfolger seit 1999
- SSL/TLS soll Kommunikation im Internet absichern
  - Authentisieren von Servern (und optional Clients)
  - Verschlüsselung
- bekannter als Hypertext Transfer Protocol Secure (HTTPS)
- Bestandteil vieler anderer Protokolle

Datagram Transport Layer Security (DTLS)

- SSL/TLS ist für TCP Datenströme gedacht
- Datagram Transport Layer Security (DTLS) für
  - User Datagram Protocol (UDP) RFC 6347
  - Datagram Congestion Control Protocol (DCCP) RFC 5238
  - Stream Control Transmission Protocol (SCTP) RFC 6083
  - Secure Real-time Transport Protocol (SRTP) RFC 5764
- Anwendungsgebiete beispielsweise Multimedia und Virtual Private Network (VPN)
- OpenSSL unterstützt DTLS

TLS Heartbeat Extension

- RFC 6520 Heartbeat Extension für TLS/DTLS (Februar 2012)
  - Keepalive Funktionalität ohne Renegotiation
  - Methode für Path MTU (PMTU) Discovery für DTLS
- Heartbeat Extension implementiert in OpenSSL (2011)
- funktioniert mit TLS und DTLS gleich
- Code präsent von 23. Dezember 2011 an
- OpenSSL 1.0.1 am 14. März 2012 publiziert (*stable*)
- *Window of Exposure* des Fehlers
  - 14. März 2012 bis 7. April 2014
  - Codebase OpenSSL 1.0.1 bis 1.0.1f

Heartbeat Dialog

1. Client sendet Heartbeat Paket mit Payload d und Längeninfo l
2. Server generiert Antwort mit
   - Payload d von Client
   - mit der Länge l
3. Problem: Client darf lügen
   - Payload *Coffee*
   - Länge 65535 Byte
4. $\longrightarrow$ Server allokiert 65535 Byte, füllt *Coffee* in Puffer

Heartbeat DTLS/TLS Pakete

Heartbeat sent to victim

SSLv3 record:

Length
4 bytes

Heartbeat Message:

Type	Length	Payload data
TLS1_HB-Request	65535 bytes	1 byte

Victim's response

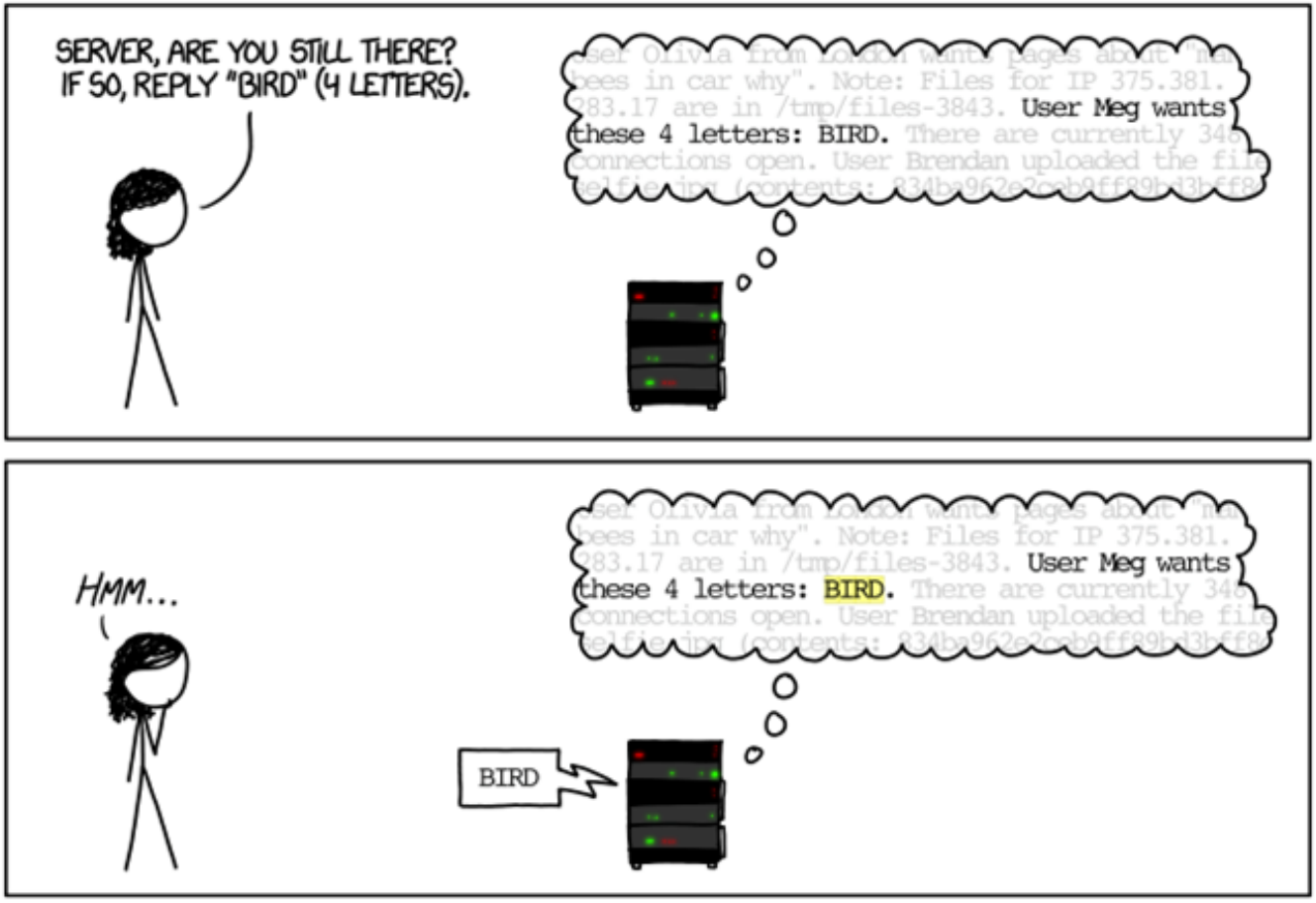
SSLv3 record:

Length
65538 bytes

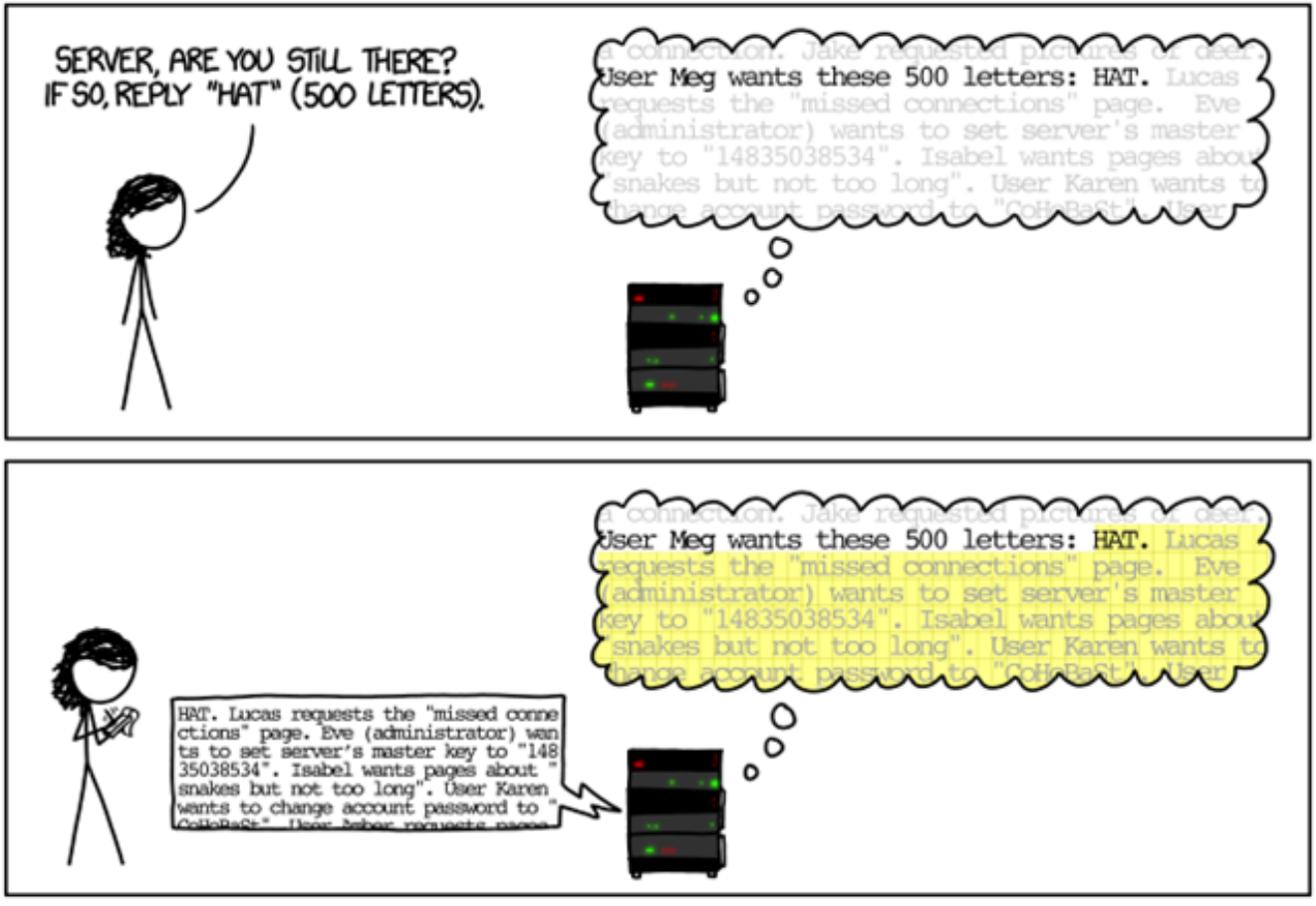
Heartbeat Message:

Type	Length	Payload data
TLS1_HB-Response	65535 bytes	65535 bytes

OpenSSL Heartbleed Bug in Aktion



OpenSSL Heartbleed Bug in Aktion



Eigenschaften des Exploits

- TLS Heartbeat Pakete erfordern keine Layer 7 Interaktion
  $\longrightarrow$ erscheint nicht in Applikations-Logs!
- Angreifer kann *ständig* Pakete schicken
  - Länge 65535, Payload 1 Byte
  - Server schickt Payload + 65534 Byte aus Speicher
- ausgelesene Daten abhängig von
  - Belastungsprofil des Servers
  - Belegung der OpenSSL Puffer (*kein* `calloc()`)
- Auslesen beschränkt auf OpenSSL Daten

Betroffene Systeme

- Netzwerkkomponenten (Firewall, Access Points, Router, Switches, . . .)
- Webseiten und Online Dienste
- Datenbanken
- Kommunikationssysteme (VoIP Telefone, Instant Messaging, . . .)
- TOR Anonymisierungsnetzwerk (mit Randbedingungen)
- Betriebssysteme (Firmware, PC, Tablets, Smartphones, . . .)
- Embedded Systems
- VPN Systeme
- . . .

Datenlecks

- Benutzerdaten
- Logins und Paßworte
- Session Cookies
- SSL Zertifikate und private Master Schlüssel
- CVE-2014-0160 kann SSL/TLS brechen, entweder
  - alle Sessions durch Master Schlüssel oder
  - nur zukünftige Sessions, wenn Folgenlosigkeit / Perfect Forward Secrecy aktiv
- *reverse Heartbleed* $\longrightarrow$ Server attackiert Client

Widerruf von Zertifikaten

- Heartbleed erfordert Widerruf von Zertifikaten
- PKI der CAs nicht für Menge vorgesehen
  - $> 5 \cdot 10^5$ ausgetauschte Zertifikate **und** private Schlüssel
  - Widerruf muß an alle Clients kommuniziert werden
  - Online Certificate Status Protocol (OCSP) hat Schwächen / wird ignoriert
- Austausch kann logistisches Problem sein
- Austausch muß **nach** Upgrade von OpenSSL geschehen!

OpenSSL Review

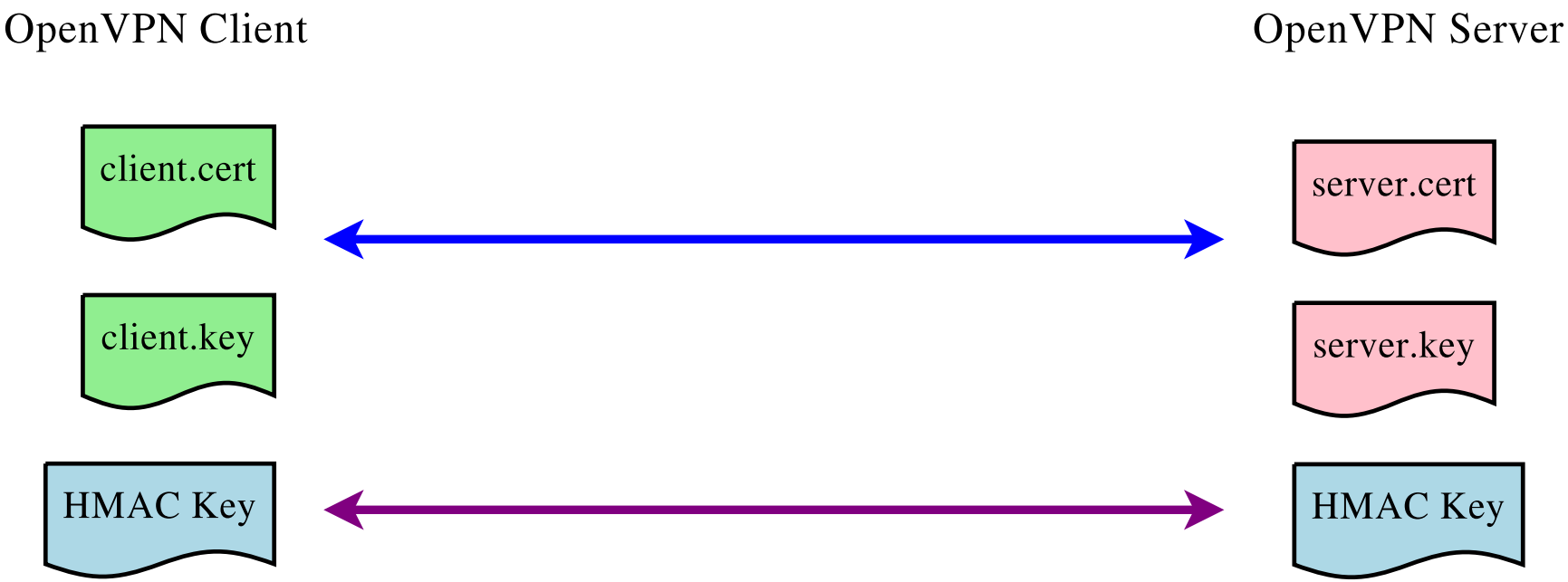
- OpenSSL Entwickler starker Kritik ausgesetzt
  - Fehler jahrelang nicht behoben (Beispiel)
- OpenBSD Team startete OpenSSL Valhalla Projekt
  - OpenSSL 1.0.1g hat $\approx$ 338.000 Zeilen Code
  - 90.000+ Zeilen Code entfernt (OpenVMS, DOS, 16-Bit Windows, Big Endian i386/AMD64, FIPS 140-2, . . .)
  - 500.000+ Zeilen `diff` Patches zu Original
  - Fork heißt LibReSSL
  - Portable Version geplant (analog wie bei OpenSSH)
- Google und Facebook spenden Resourcen für OpenSSL Entwicklung
- Google hat eigenen Fork namens BoringSSL

Alternative Bibliotheken

- Botan
- Bouncy Castle
- cryptlib
- CyaSSL
- GnuTLS
- Java Secure Socket Extension (JSSE) von Oracle®
- LibReSSL
- MatrixSSL
- Network Security Services (NSS)
- PolarSSL
- Security Support Provider Interface (SSPI) von Microsoft®
- Secure Transport von Apple
- SharkSSL

Aber: Keine Implementation ist fehlerfrei.

Konfigurationen zur Vermeidung – OpenVPN



Offene Probleme

- 12.043 der Alex Top 800.000 SSL/TLS Webseiten nicht gepatcht (Stand 20. Mai 2014)
- weitere Systeme noch nicht gepatcht, Heartbleed wird bleiben
- SSL/TLS ist kritische Komponente
  - *defence in depth* nur bedingt/nicht möglich
  - Code Teil der Infrastruktur
- Linux Foundation gründete die Core Infrastructure Initiative (CII)
- regelmäßige Code Audits nötig
- Fokus auf Sicherheit statt Performance

Zusammenfassung

- CVE-2014-1060 OpenSSL Heartbleed Bug war sehr lehrreich
- Bibliotheken brechen ganze Klassen von Implementationen
- Wenige wissen wie es hinter den Kulissen im Code ausschaut
- FOSS ein klarer Vorteil
  Wieviele Heartbleed-artige Bugs in proprietärer Software noch schlummern, ist unbekannt . . .
- konkurrierende Implementationen bei Infrastruktur überlebenswichtig

Über die DeepSec

Die DeepSec GmbH veranstaltet seit 2007 jährlich im November die „DeepSec In-Depth Security Conference" in Wien. Die DeepSec bringt als neutrale Plattform die Sicherheitsexperten aus allen Bereichen zum Gedanken- und Erfahrungsaustausch zusammen. Dort erhalten IT- und Security-Unternehmen, Anwender, Behördenvertreter, Forscher und die Hacker-Community in über 30 Vorträgen und Workshops die Chance, sich über die aktuellen und zukünftigen Sicherheitsthemen auszutauschen. Die Konferenz möchte insbesondere dem verbreiteten Vorurteil entgegen wirken, dass Hacker zwangsläufig Kriminelle sind.

Kontakt

- Web `https://deepsec.net/` & `https://deepintel.net/`
- E-Mail `rpfeiffer@deepsec.net` V `deepsec@deepsec.net`
- Videos `http://www.vimeo.net/deepsec`
- Twitter `https://twitter.com/deepsec`
- E-Mail (PGP/GPG) `0x8531093E6E4037AF` oder `0xE1170EDE22860969`
- RedPhone & TextSecure: +43.676.5626390
- Threema: `76WHDZTR`

Obligatorische geheime Botschaft

```
---BEGIN PGP MESSAGE---
Version: GnuPG v1.4.12 (GNU/Linux)

jA0EDQMCJVGNuoLcZjZg0ukBRImOaz/HUA5x1Ao3g3SNX1PU6MtzbYtMQnj3hP68
5Rxxx6oxrtQXJf60X+oCxT566d862pxRISFJAstzznGnlzp3/xAOGrJxzvw01akR
fn166+qyUkFvWzYWM+iHlgxKTmpO3tD82sqCL/g/1/uGeQ9WS03qPVs34pSXW5lu
81Hu1AjLcQdj1rDleCxsJYRbR70hbX+P2031MGzeyVEMHk8xtoPaZGlwc2gDZj6a
eiec7IMmblAtWxZVCw5oMQq5QALrGJMvCnvNrINFEPe6wy99KoxnCmWrvS5X1xIk
NunFFvRkrL8794YevyN/v0ks5QsWOXqwMXFm0AqHemQSbvGXKlCOnf4qQpnKCXVW
eMFqtGULKLp6SMSzHt7vUR53SOJrC39O8th/DWeYLqzSUMVHyk3c5NgAWbPQr89c
kLJUI470JoqI1ACRKyPo58GrG+8ipW9bnRD1wZC+Sf8/BQuktHPpIO0n89nkrXdN
abxCMSHhXmrKEN1pCG18n5crW0CV4j9KLu2M6CDS2UnW+gYVXUqqTzkVclX2AQbH
pcizOOZRPwkjvjTzqea1HO6bZnRzZ0tgIrux6ZO1WWB74ru4VgmcSMuQEqkWPEts
MSo9YClWxS3FN/B34fHPlRzcZ0qBTah0FTSNr6oNc/IJsbhSmhD1DffWFkhttPDg
NiHC/NaAnjdYVdy9c5f5oQ9lgEvbjBRyjnwo4/53tpYThp8=
=IbYI
---END PGP MESSAGE---
```

Hinweis: GnuPG kann auch symmetrische Verschlüsselung.

DEEPSEC