Den Schlapphüten die Ohren verstopfen

Transportverschlüsselung für alle

Alexander Schreiber <als@thangorodrim.ch>

http://www.thangorodrim.ch/

Chemnitzer Linux-Tage 2014, 2014-03-15

Privacy - like eating and breathing - is one of life's basic requirements.

– *Katherine Neville*

Inhalt

Über den Autor

- beschäftigt sich seit fast 20 Jahren mit Linux
- tätig als Systemingenieur bei Google Switzerland
- schätzt seine Privatsphäre

Um was geht es?

Globaler Angriff auf die Privatspähre

- globale, anlassunabhängige Überwachung des Datenverkehrs
- Anzapfung von Unterseekabeln
- Anzapfung firmeninterner WAN-Verbindungen
- Kompromittierung von Routern
- Zusammenarbeit verschiedener Geheimdienste zur Umgehung nationaler Überwachungsbeschränkungen
- Ziel: Überwachung **allen** Datenverkehrs im Rahmen des technisch machbaren (mit großen Budget)
- Wie die Privatsphähre bewahren?

Ziele des Vortrags

- Klärung der Angriffsszenarios:
  - Schleppnetzüberwachung
  - Datenabgriff bei Datenübertragung im Klartext
- Maßnahmen, dem Schleppnetz zu entgehen
- Wiederherstellung der Vertraulichkeit über offene Kommunikationspfade
- Reduzierung der lesbaren Datenspuren
- Transportverschlüsselung: Schutz von Daten in Bewegung

Worum geht es nicht

- Datei-/Festplattenverschlüsselung: Schutz ruhender Daten
- Nutzdatenverschlüsselung oberhalb der Netzwerkprotokolle (z.B. PGP, S/MIME)
- Schutz gegen gezielte Überwachung/Datenabgriff durch Regierungsorgane im Rahmen der gesetzlichen Möglichkeiten
- Schutz gegen gezielte Überwachung/Angriffe durch andere hochkalibrige Angreifer inklusive Angriffe ausserhalb der Netzwerkinfrastruktur
- Schutz gegen Trafficanalyse ("Metadaten")

Warum Transportverschlüsselung

- Netzverkehr ist überwiegend Klartext: HTTP, SMTP, ...
- $\longrightarrow$ von Interessierten mitlesbar & modifizierbar
- Abgriff vertraulicher Daten, Übernahme von Sessions, ...
- Problem für Privatsphäre und Sicherheit
- Beispiel: Einschleusung von Werbung in HTTP durch ISPs
- Einschleusung von Malware
- Transportverschlüsselung schützt!
- SSL/TLS

SSL/TLS

- SSL: Secure Sockets Layer, Vorläufer von TLS
- TLS: Transport Layer Security
- hybrides System: symmetrische & public key Verschlüsselung im Einsatz
- public key: Authentisierung, Schlüsselaustausch
- symmetrisch: Verschlüsselung der Kommunikation
- Authentisierung via X.509 Zertifikate (Server, seltener Client)
- Schlüsseltausch: Diffie-Hellman (DH)
- Verschlüsselung der Kommunikation: AES, Blowfish, ...

Zertifikate & die chain of trust

- Zertifikate: “Ausweis” – mit wem spreche ich?
- Certification Authority (CA): (idealerweise) vertrauenswürdiger “Aussteller”
- chain of trust: Zertifikat → (intermediate CA) → root CA
- auf public key Kryptographie basierend
- root CA muß vertrauenswürdig sein, aber: Fehlausstellungen, Hacks, . . .
- vorinstallierte root CA Zertifikate
- CA Zertifikate nachinstallieren (Firmennetz, eigene CA)

Kritische Details

- einfach nur Transportverschlüsselung reicht nicht
- unsichere Algorithmen, nachträgliche Entschlüsselung
- Lösungen:
  - perfect forward secrecy ((EC)DHE): ephemeral DH
  - sichere Kryptoalgorithem mit langen Schlüsseln (AES)
  - sichere Hashalgorithmen (SHA-2 Familie)
  - schwache Algorithmen (RC4, DES, ...) vermeiden
- man-in-the-middle: Zertifikate prüfen
- System CA store gegen Einschleusung von Zertifikaten sichern

Verschlüsselung auf Protokollebene - Übersicht

- zwei Varianten:
  - komplettes Protokol in SSL/TLS einpacken, separat ansprechen (andere Ports)
  - existierende Verbindung auf SSL/TLS umschalten (STARTTLS), muß von Clientsoftware unterstützt werden
- WWW: https (SSL/TLS, siehe RFC 2818), separater Port
- SMTP: entweder SMTP+STARTTLS auf 25/tcp oder SSL/TLS SMTP auf separaten Ports
- POP3 und IMAP: SSL/TLS Varianten auf eigenen Ports
- `openssl s_client -connect google.ch:443 -status`

HTTP/IMAP/POP3/...

- Standard: Server nicht authentisiert, Session im Klartext
- TLS: Server authentisiert mit Zertifikat, Session verschlüsselt
- HTTPS, IMAPS, POP3S, ...
- von allen wichtigen Servern unterstützt
- HTTP-only Server: SSL reverse Proxies (Squid, Varnish, ...)
- stunnel (z.B. uucp-over-stunnel-over-IP)
- wichtig: Kryptooptionen korrekt setzen! (DHE, AES, ...)
- für SSL-fähige Server: nur Zertifikat + Schlüssel nötig

Transportverschlüsselung bei EMail

- komplettes Protokoll (andere Ports) oder STARTTLS
- wer schickt Mail mit Transportverschlüsselung?
  - Postfix Logs: `"setting up TLS connection"` suchen
  - EMail-Client: Header lesen, in `Received:` Headern nach `"using TLS"` suchen
  - Beispiel EBay:

    ```
    using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits
    ```

  - → perfect forward secrecy, sichere Krypto
  - Google, EBay, Debian ML, Yahoo, AOL, Microsoft, Freshmeat, Sourceforge, Linbit, digitec, . . . in variabler Sicherheit

Private Netze - OpenVPN

- OpenVPN: Open Source Virtual Private Network software
- Aufbau verschlüsselter VPNs über das Internet
- Anwendungsbeispiele:
  - mehrere Standorte sicher verbinden
  - Aussendienstler
  - Sicherung von WLAN-Links
  - Heimnetz und externer Server
  - Zugriff auf das Heimnetz von unterwegs
  - sicherer Netzzugang in unsicheren Netzen

OpenVPN: Übersicht

- eigenes Protokoll
- Authentisierung via shared secret/Zertifikate/Passwort
- setzt auf OpenSSL und SSLv3/TLSv1 auf,
- sehr flexibel, IPv6 Unterstützung
- Userspace Anwendung
- *wesentlich* einfacher als IPSec zu konfigurieren
- breite Plattformunterstützung: Linux, *BSD, MacOS X, Windows, Android, iOS, ...

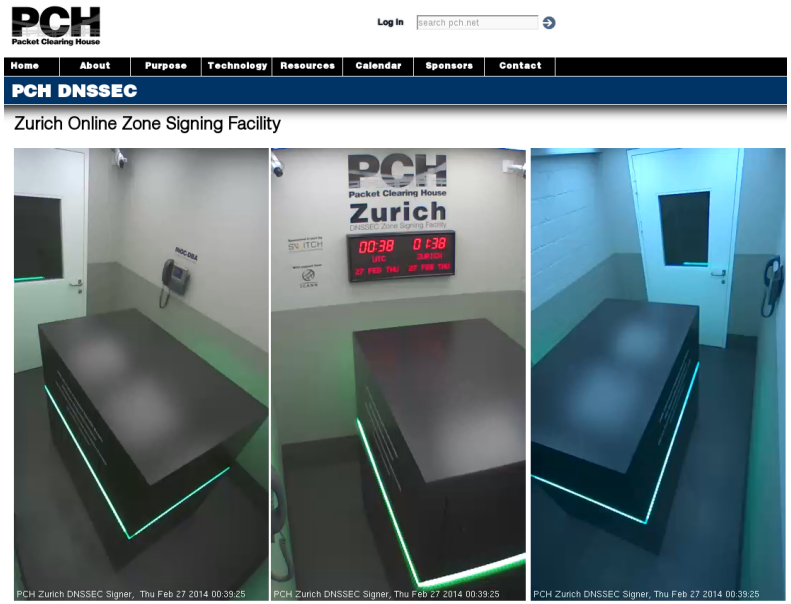
minimale OpenVPN client.conf

```
client                                  # client mode
dev tun                                 # tun device
proto udp                               # udp encapsulation
remote vpn-server.example.com 1234      # VPN server
ca keys/ca.crt                          # CA cert
cert keys/client.crt                    # client cert
key keys/client.key                     # client secret key
```

Private CA für zuhause

- CA für den Eigenbedarf:
  - Serverzertifikate: HTTPS, IMAPS, OpenVPN, ...
  - Clientzertifikate: OpenVPN
- eigene CA: Kontrolle & Sicherheit
- relativ einfach mit fertigen Werkzeugen
- Empfehlung: easy-rsa von OpenVPN

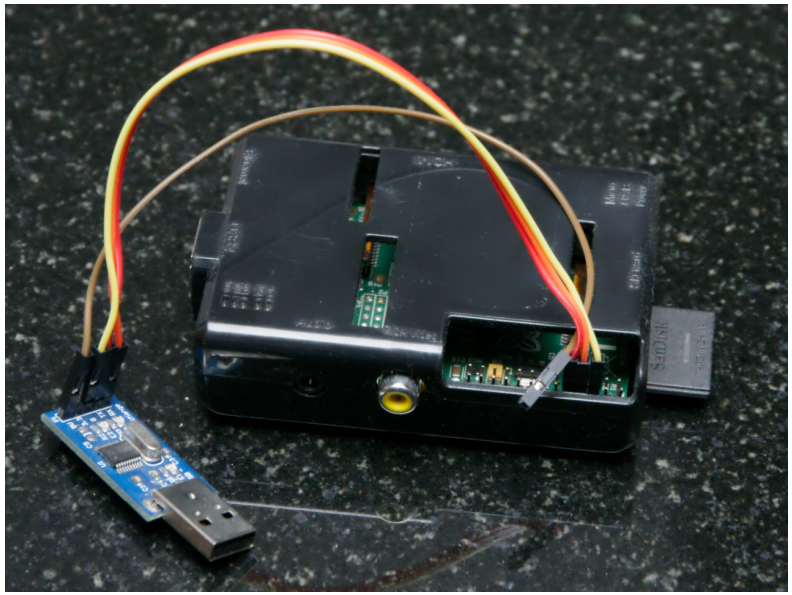
Der ganz aufwendige Ansatz ...



https://www.pch.net/dnssec/zurich.php

- SCIF in einem Rechenzentrum
- GSA Klasse 5 Panzerschrank (10 min Widerstand) im SCIF
- Computer im Panzerschrank
- Hardware-Krypto-Modul in Computer
- Schlüssel im Hardware-Krypto-Modul

. . . und der ganz einfache Ansatz.



- Raspberry Pi
- mit serieller Konsole (kein Netzwerk!)
- mit SD-Karte
- USB-Stick zum Datentransfer

easy-rsa

- CA Setup:
  - easy-rsa Verzeichnis nach Arbeitsverzeichnis kopieren
  - vars Datei editieren (Schlüssellänge, Namen, ...)
  - `source ./vars`
  - `./clean-all`
  - `./build-ca`
- DH Parameter erzeugen:
  - `./build-dh`
- Server-Key erzeugen:
  - `./build-key-server gate.vpn.example.com`
- Client-Key erzeugen:
  - `./build-key client.vpn.example.com`

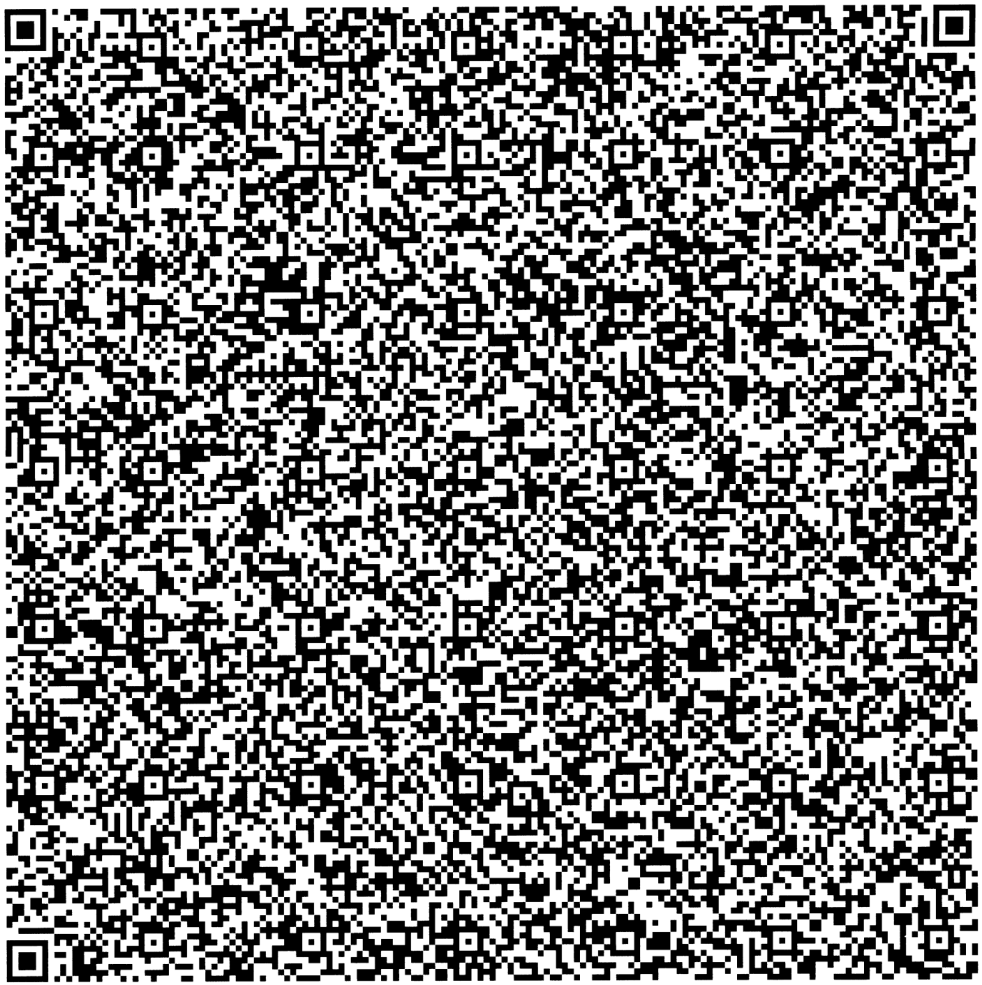
easy-rsa ./vars example

```
export KEY_SIZE=2048
export CA_EXPIRE=3650
export KEY_EXPIRE=3650
export KEY_COUNTRY="CH"
export KEY_PROVINCE="ZH"
export KEY_CITY="Zurich"
export KEY_ORG="CLT2014"
export KEY_EMAIL="admin@example.org"
export KEY_CN=changeme
export KEY_NAME=changeme
export KEY_OU="CLT2014 example"
```

CA Backup

- Zertifikate per default 10 Jahre gültig
- Verlust der CA wegen Bitfäule . . . nicht gut
- Flash (SD-Card, USB-Stick) kein Langzeitmedium
- Backup notwendig!
- Langzeitmedium: Papier
- gzip -9, base64, QR-Code, ausdrucken
- QR-Code decodieren (z.B. Barcode Scanner von ZX Crossing)

ca.crt Backup auf Papier



Fragen?

Fragen?