NSA und die Kryptographie: Wie sicher ist sicher?

René 'Lynx' Pfeiffer

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

Linuxwochen Eisenstadt 2014
Fachhochschulstudiengänge Burgenland, Studienzentrum 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 – Warum?

- Kryptographie ist wichtige Sicherheitskomponente
- Klartext ist so 2000 . . . aber **vor Christus!**
- Klartext ist fahrlässig (schon lange)
- Verschlüsselung macht neugierig (seit über 1200 Jahren)
- Welche Algorithmen und Methoden greifen (jetzt) noch?

National Security Agency (NSA)

- NSA erfaßt *Signals Intelligence (SIGINT)* für USA
- *„The NSA is tasked with the global monitoring, collection, decoding, translation and analysis of information and data for foreign intelligence and counterintelligence purposes, including surveillance of targeted individuals on U.S. soil.“*
- Autorisiert für Geheimoperationen, Abhöranlagen & Sabotage

Zugriff auf Kommunikation

- Sinn und Zweck der Dienste
- Anstrengungen schon seit den 1990er Jahren
  - „Crypto Wars“ , Begründung Cypherpunk Bewegung
  - Exportverbote für Kryptographie
  - Clipper Chip für Key-Escrow
  - Skipjack Algorithmus der NSA
  - CALEA, Cyberspace Electronic Security Act, . . .
- Leaks von Snowden beleuchten Geheimdienstarbeit in punkto Kommunikationsüberwachung
  - Projekt Bullrun (NSA)
  - Projekt Edgehill (GCHQ)
  - plus weitere Werkzeuge (Hammerstein, Hammerchant, . . .)

Projekt Bullrun/Edgehill

- Reaktion auf Verbreitung der Kryptographie im Internet
- Kompensieren der fehlenden Schlüsselhinterlegung (*key escrow*)
- Sicherstellung des Zugriffs auf verschlüsselte Kommunikation
- Kombination von verschiedenen Methoden
- NSA startete Programm bereits $\approx$ 2000

Cyber Spy Gang

- NSA und GCHQ sind nicht alleine . . .
- „Five Eyes“ – NSA, GCHQ, CSE (CA), DSD (AU), GCSB (NZ)
- MI5, MI6, US DoJ, CIA, FBI, Homeland Security, DEA, IRS, . . .
- Partner in Dänemark, Deutschland (BND), Israel, Niederlande, Singapur, Schweden, Schweiz
- Programme in anderen Ländern zu erwarten/vorhanden
  - russischer FSB (SORM, Internetzugriff, . . .)
  - China (61398 部☒)
  - Iran
  - . . .

Um welche Daten geht es eigentlich?

- Metadaten
  - Quelle, Ziel (in jeder Form)
  - Protokoll
  - Zeitstempel, Dauer, Datengröße
  - Inhaltstyp
  - jegliche Orte (GPS, Ortsnamen, Domains, . . .)
  - . . .
- Inhaltsdaten
  - Gespräche, Korrespondenz
  - Dateien jeglicher Art
  - . . .

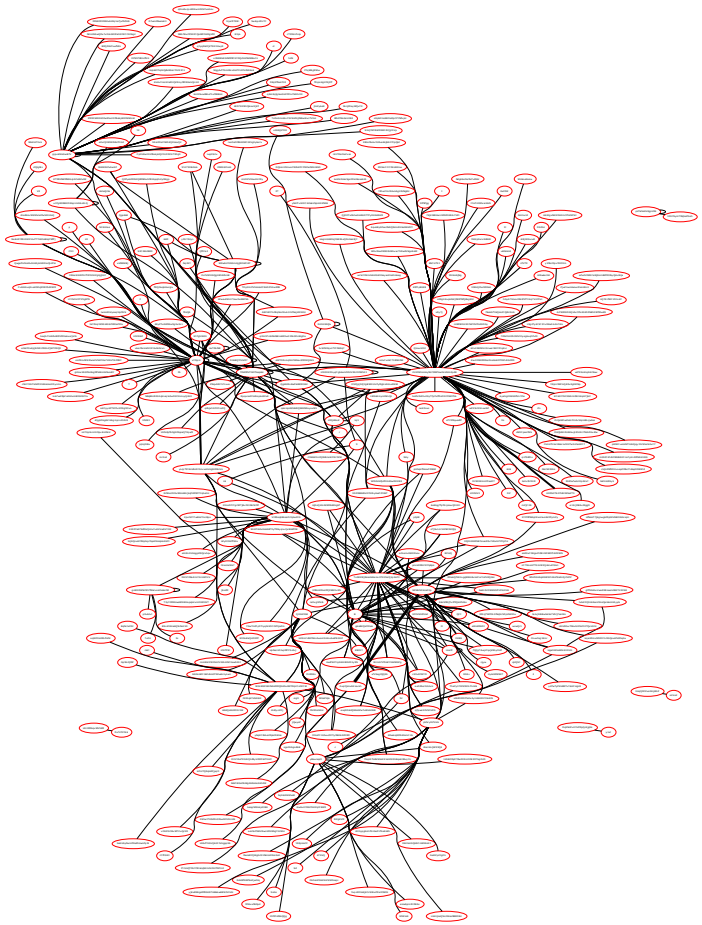
Fokus der Attacken von NSA/GCHQ

- Abfangen von Netzwerkverkehr
  - mit/ohne Hilfe von Internetanbietern
  - Kompromittieren von Infrastruktur (Router, Kabel, . . .)
- Kompromittieren von Endgeräten (Smartphones, PCs, . . .)
- Kompromittieren von Implementationen
  - Einbau schwacher / geschwächter Kryptographie
  - Schwächen von Standards
- Beschaffen von kryptographischen Schlüsseln
- *notfalls* „brute force“ Ausprobieren („Brechen“)

Abfangen von Netzwerkverkehr

- *Jedweder* Datenverkehr im Internet ist *Freiwild*!
- (Mehrfache) Verschlüsselung ist *best practice*!
- Metadaten schwierig bis nicht zu schützen
  - Spuren in Logfiles, IP Adressen, Mobilnetzwerken, . . .
  - Anonymisierungsnetzwerke
  - gebündelte Datentransmissionen (*Multiplexing*)
  - VPN Mesh Netzwerke (siehe z.B. *tinc*)
- technische Lösungen müssen mit *OpSec* ergänzt werden!

Metadaten aus Korrespondenz



Kompromittierte Systeme

- NSA kauft *Exploits* von Vupen, Raytheon, Lockheed Martin, Northrop, ...
- gutes Zeichen – Indiz für ungebrochene Mathematik
- effizientes Umgehen von Schutzmaßnahmen
  - Kopieren von (temporären) Schlüsseln
  - Mitschneiden direkt an der Quelle
- an alle Crypto Hipster: **Mobiltelefone sind nicht vertrauenswürdig!**
- Aufgrund Kosten nicht gegen breite Masse angewendet

Kompromittieren von Implementationen

- NSA Hintertür in Produkten der Crypto AG Ciphering Machines
- NSA Hintertür in IBM Lotus Notes
- NSA Hintertür in Microsoft Windows ADVAPI.DLL
- NSA „bittet“ Hersteller periodisch um „kleine Anpassungen“
- Zweck
  - Schwächung der Kryptographie
  - Manipulation von Zufallsquellen (Schlüsselgenerierung)
  - Einbau von Zweit-/Dritt-/. . .schlüsseln
  - . . .

Schwächen von Standards

- Fall: Dual_EC_DRBG – Dual Elliptic Curve Deterministic Random Bit Generator
- NIST publizierte diesen als Standard
  - Schwächen schon bei Standardisierung bekannt (2006)
  - Vermutung einer Hintertür (2007)
  - Firma RSA Security verwendet Dual_EC_DRBG zu Schlüsselgenerierung
- effiziente Reduktion des *key space* für „*brute force*“
- RSA Security warnt vor PRNG (2013, *nach* Snowden Leaks)

NSAs und RSAs Angriff auf TLS

- Dual EC PRNG + TLS = DualECTLS
- Dual EC PRNG vorhanden in
  - RSA BSAFE Share for C/C++
  - RSA BSAFE Share for Java
  - Microsoft® Secure Channel (SChannel; verwendet von IIS) *
  - OpenSSL FIPS Object Module
- Studie fand Dual EC Attacken auf TLS Transmissionen möglich
- RSA BSAFE Library hat *TLS Extended Random Modus*
  - auf Bitte von NSA eingefügt (2008)
  - Modus beschleunigt Attacken um Faktor 65.000
  - still deaktiviert von RSA Monate nach Snowden Leaks
- *klare* Anzeichen für gezielte Attacke auf TLS

Was noch gut funktioniert

- SSL/TLS, (Open)SSH, OpenVPN™ & IPsec in richtiger Konfiguration
- PGP/GPG, (Open)VPN & OTR auf unkompromittierten Systemen
- TOR (O-Ton NSA: „TOR stinks!“)
  - *Verwenden wie empfohlen!*
  - Hidden Services
- Implementationen in Freier Software & offenen Standards
  - Keine Black Boxes!
  - Keine proprietäre Software!
  - Keine „Open Source“ Software (wie z.B. TrueCrypt)!
  - Keine *binary blobs*!

SSL/TLS Probleme

- OpenSSL „Heartbleed “Bug (CVE-2014-0160)
  - TLS Heartbeat Extension fehlerhaft implementiert
  - TLS Heartbeat Extension nur für UDP sinnvoll
- Widerruf von Zertifikaten funktioniert nicht zuverlässig
  - Chrome Browser prüft Widerrufe nicht
  - Online Certificate Status Protocol (OCSP) nicht zuverlässig
  - Infrastruktur nicht für 100.000+ Widerrufe vorbereitet
- Verbessrungen sind in Planung
  - TLS v1.3
  - OCSP Stapling

Alternative Standards

- Advanced Encryption Standard – NIST, USA
  NIST muß mit der NSA zusammenarbeiten
- CRYPTREC (Cryptography Research and Evaluation Committees) – Japan
- NESSIE (New European Schemes for Signatures, Integrity and Encryption) – EU
- ECRYPT (European Network of Excellence in Cryptology) – EU
- Aber:
  - *Keine Algorithmen verwenden, die nicht untersucht wurden!*
  - *Keine Algorithmen selbst entwickeln!*

Perfect Forward Secrecy (PFS)

- Langzeitschlüssel $\longrightarrow$ Session Key für Transmission
- Idee: Session Key nicht kompromittiert, wenn Langzeitschlüssel kompromittiert
- Generierung Session Key erfordert **gute** Zufallszahlen
- Perfect Forward Secrecy wird unterstützt von
  - IPsec (optional)
  - OpenSSL & GnuTLS
  - OpenVPN™ (nur mit SSL/TLS Authentication!)
  - OTR
  - SSH
- Enigma kannte Session Keys, moderne ISPs und Start-Ups anscheinend nicht!

PFS mit OpenSSL konfigurieren

Für nginx:

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

ssl_prefer_server_ciphers on;

ssl_ciphers „EECDH+AESGCM EECDH+aRSA+AESGCM EECDH+SHA384 EECDH+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS“ ;

Für Apache (siehe auch Security/Server Side TLS):

SSLProtocol all -SSLv2 -SSLv3

SSLHonorCipherOrder on

SSLCipherSuite „EECDH+AESGCM EECDH+aRSA+AESGCM EECDH+SHA384 EECDH+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS“

Siehe Applied Crypto Hardening Projekt.

PFS richtig konsumieren

- Server *und* Client einigen sich auf Modus
- viele Browser akzeptieren unsichere/schwache Ciphers/Keys
  - Defaults oft nicht sicher genug
  - *Preferences* / *about:config* is your friend!
  - minimal SSLv2 und Schlüssel $<$128 Bit deaktivieren!
- TLSv1.1 und TLSv1.2 noch nicht überall vorhanden
- Updates können Konfiguration ändern!

Algorithmen

- Analyse von Algorithmen sehr schwer
- „verwendbar“
  - AES, ~~Blowfish~~, Twofish, Camellia, Serpent
  - SHA-2 Familie (SHA256, SHA384, SHA512, . . .), RIPEMD Familie, Tiger, Whirlpool
- vermeiden
  - Digital Signature Algorithm (DSA) in jeder Form
  - RC4 (gute Attacke 2013 publiziert)
  - DES (!)
  - 3DES (mit Key Option 2 & 3, und generell)
  - Hashalgorithmen MD5, SHA1, NIST Version von SHA-3 (Keccak)

Betriebsarten von Blockchiffren

- Blockchiffren lassen sich in Stromchiffren umwandeln
  - Cipher Block Chaining Mode (CBC Mode) sehr verbreitet
  - CBC in TLS v1.0 angreifbar (BEAST Attacke)
    behoben in TLS v1.1 und höher
- gute Alternativen sind
  - Counter Mode (CTR) – leicht(er) verfügbar
  - Galois/Counter Mode (GCM) – mit Vorbehalt
- *Randnotiz:* Attacken erforden (oft) hohe Bandbreite

Zusammenfassung

- Kryptographie funktioniert (noch).
- Implementationen { funktionieren, nicht, immer }.
- Private Schlüssel sind Kronjuwelen des 21. Jahrhunderts.
- Spionageskandal erzwingt Hinterfragen *sämtlicher* Infrastruktur.
- Es gibt keine Abkürzungen.
- Eigenes Know-How in Sachen Crypto unabdingbar für die Zukunft!
  $\longrightarrow$ Hausaufgabe: Vergleich elliptische Kurven mit RSA Algorithmen

Warnung!

NSA (Leaks) stilisiert:

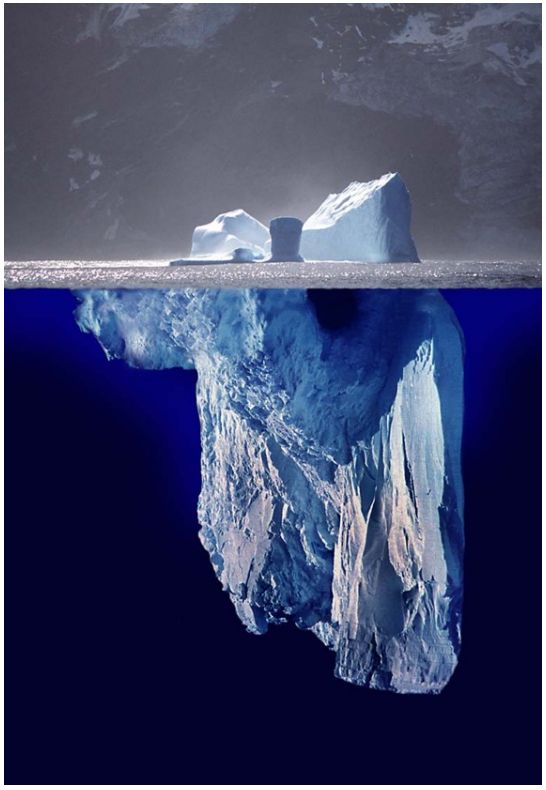


Bild von Uwe Kils.

Fragen?


WARNING
You are now entering a top secret lair.
The area is patrolled regularly by
psychotic henchmen, rabid guard dogs,
sadistic killer clowns, insane minions,
genetically engineered super soldiers,
flesh eating zombies, evil lackeys
heavily armed titanium-robots and
super villains in training.

Kontakt

Informationen über die DeepSec und DeepINTEL Konferenzen erhält man über die folgenden Wege:

- https://deepsec.net/ & https://deepintel.net/
- E-Mail (PGP/GPG) 0x8531093E6E4037AF oder 0xE1170EDE22860969
- Videos http://www.vimeo.net/deepsec
- Twitter https://twitter.com/deepsec
- RedPhone & TextSecure: +43.676.5626390

Obligatorische geheime Botschaft

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

jA0EAwMCWyPontkww4RgycEAk47QuXBPkckVahSHlwmuFj7EYT21eyz9g4gtzDMD
67vohgWPYFi6tdCkEXE9yp/SeR8JxrR73bmukrY9779lkZ+vEWFaFNd+CdlQio6Q
DSLdayq57pd/E7jdvbhfkLzGpu/oAVGTsMcLRWWjTvSlVL65PRPZNBu0Ce22uMhr
4mFiBJCmL8+vTUXuuDo09QDu5CIxVbBWnJ9JN6Y4tJuiBykFgo0gVftmFmjJqVX6
GMs4ygxCdcdv6JEczXd53D7glIlFTeYEEk44rNCZDnVRpnQ9ByT3mgJBfp9g6nSg
3Z8hXPw1A5z12m3v8elCCt8jFbtQ4HSKsRW41oOGJ1JG8bHH5dGtYOTbNxrdPNuM
df7G1Emzd/vFWE5wKW8xoCP4V9oWLBdGZsyQejq13A6z92neR8uqpyZs5EjhUdYA
U0tClauFgkhu23eenRALlUQFKfhbmmCqvrfyR6OsH4k2+5Rxm05wn9YitdD6uTx+
3CiSEQYDkzjWb0PFl8jrJRbmu/1OfjSpcrZFtywyzOD2MmUFWG5NP/q/32wDOc61
DD0IBF2cUsmpL1AspjOiRStJuLPz5Bqa5WzUXzFTKT8H3Q==
=F7ea
--END PGP MESSAGE--
```

Hinweis: GnuPG kann auch symmetrische Verschlüsselung.