# Die zeitliche Dimension der Sicherheit

## Nichts ist für die Ewigkeit

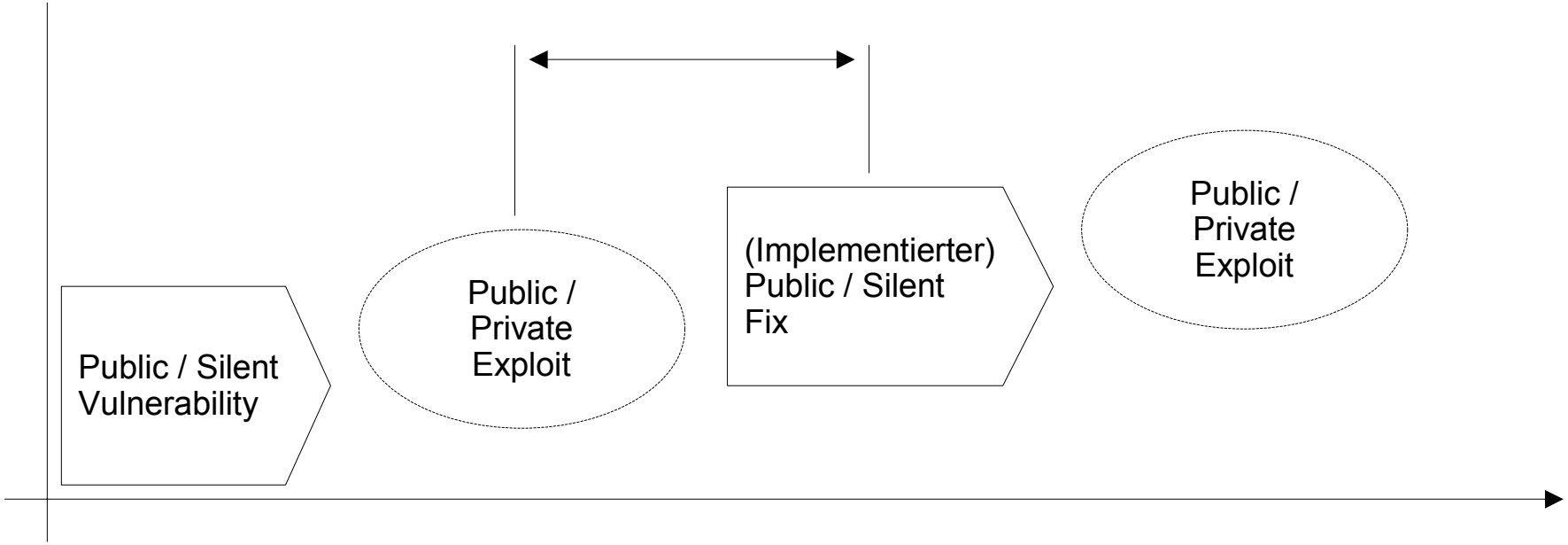
**19. Chaos Communication Congress**
**Berlin, 27. – 29. Dezember**


**Ron** **ron @ ccc.de**
**Frank Rieger** **frank @ ccc.de**

## *Was einem schnell leid tut…*
## *Trivialitätsebene 1: Minuten bis Wochen*

### Vulnerability Window

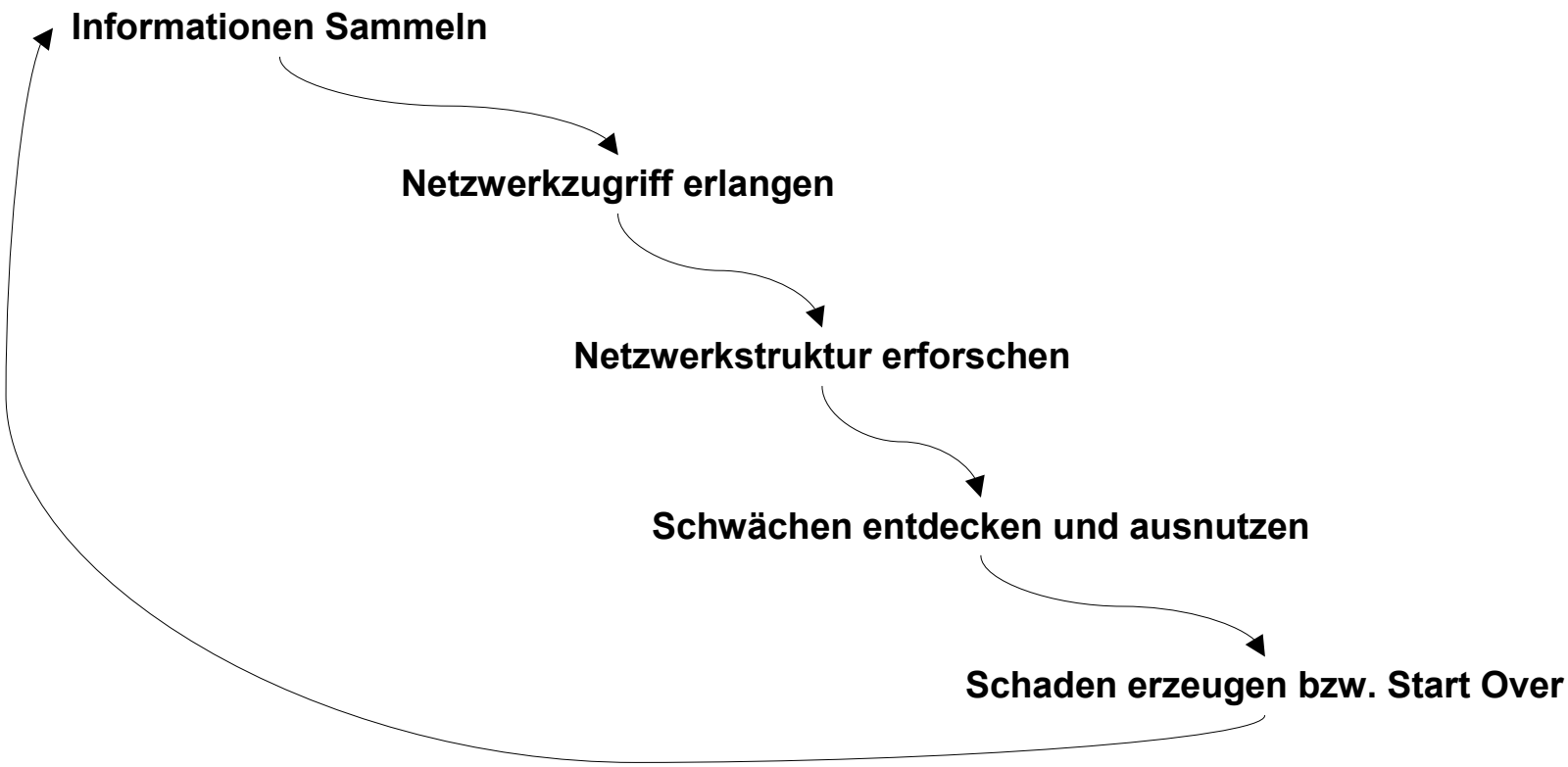
- Die Zeit zwischen der Verfügbarkeit des Exploit und (Implementierung des) Fix




**Für den Betrachter ohne "Inside Sources" ist das Verwundbarkeitszeitfenster die Zeit zwischen veröffentlichter Verwundbarkeit und (Implementation von) veröffentlichtem Fix.**

## *Was einem schnell leid tut…*
## *Trivialitätsebene 1: Minuten bis Wochen*

**Attack Window**



## *Was einem schnell leid tut…*
## *Trivialitätsebene 1: Minuten bis Wochen*

**Attack Detection Window**

**(technische) Unregelmäßigkeit**

- Sensibilität abhängig von den Sensoren, Platzierung / Ausrichtung der Sensoren (Exkurs: Inbound Traffic vs. Outbound Traffic), Auswertung der Sensoren (Thresholds usw. Beispiel Firewallrauschen)

**Oblivion.org**

**Oblivion.tech**

**Oblivion.sla**

**Erkannter Angriff**

- Evidence gathering
- Angreifer identifizieren

**Gegenmaßnahmen**

- Abschalten, Fixen, Verklagen, Erschießen

## *Exkurs: Qualität der Implementation (1/2)*

**Full Disclosure wo bist du geblieben?**

- Den Standpunkt von MS kennen wir ja…
- Silent Bugfixes jetzt auch in Open Source Software (siehe Exkurs)
- Angreifbarkeit der Source-Distributionssysteme
- Full Disclosure “Psychofalle”

**Who audits (reads) the Source anyway?**

- Sendmail
- SSH
- OpenSSL

## *Exkurs: Qualität der Implementation (2/2)*

**Auch in gut abgehangenem, viel benutztem Code können sich über Jahre hinweg finsterste Probleme verbergen**

**Private Exploits sind weiterhin im Trend**

**Durchschnittliche Dunkelperiode von Exploits ist schwer zu bestimmen**

**Source Audits von motivierten und fähigen Leuten helfen, lesen allein reicht nicht**

- Computer unterstützte Source Audits sollten viel weiter verbreitet sein

**Auch Open Source Software benötigt eine gewisse Reifephase bis sie halbwegs sicher ist**

## *Exkurs: Silent Bugfixes vs. “den Patch brauche ich nicht”*

**Silent Bugfixes sind an der Tagesordnung**

- Den Standpunkt von MS kennen wir ja…
- Silent Bugfixes jetzt auch in Open Source Software

**Damit kann man sich es nicht mehr leisten, einen Fix *nicht* zu implementieren.**

**Der Stress, den die Praktizierung von Silent Bugfixes bei paranoiden Sysadmins induziert, kann damit pathologische Ausmaße annehmen…**

## *Was einem leid tut, wenn es viel zu spät ist…*
## *Trivialitätsebene 2: Jahre bis Jahrzehnte*

**Halbwertszeit der Schlüssellänge, der Implementation, …**

**Was uns die Geschichte lehrt oder die Haltbarkeit von klassischer Kryptographie**

- Enigma
- DES
- RSA 512
- AES (?)

**Haltbarkeit hängt von Motivation, Fähigkeit und Ausstattung des Angreifers ab**

**Beispiele**

- MS Access
- MS PPTP
- /dev/random
- WLAN
- Bluetooth(?)

## *Exkurs Verfügbarkeit – Lesbarkeit hat eine Halbwertszeit*

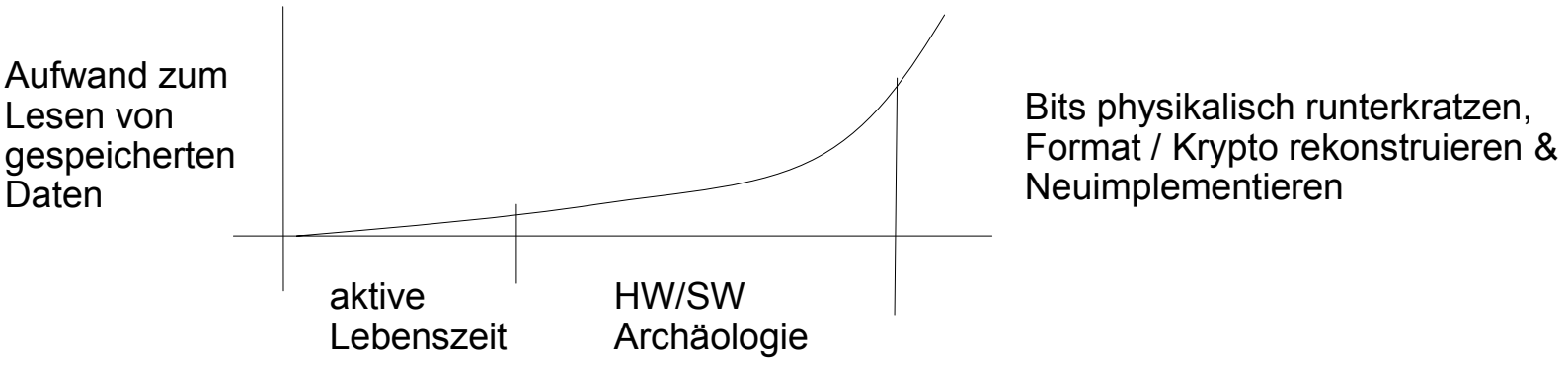
**Verfügbarkeit von gespeicherten Informationen ist zeitlich begrenzt**

**Was uns die Geschichte lehrt**

- Papier
- Lochkarten
- Mikrofiche
- Magnetbänder

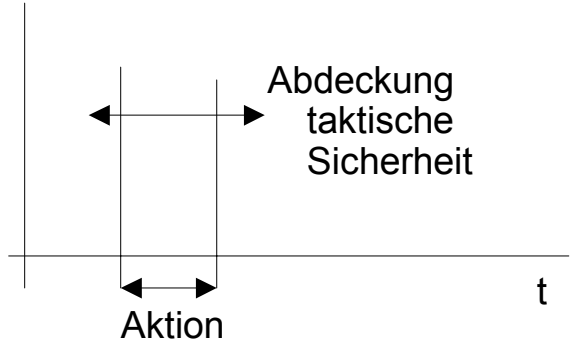
**Die Zeitspanne der Lesbarkeit sinkt mit steigender Informationsdichte**

- Softwareverfügbarkeit, Haltbarkeit des Speichermediums, Formatverfügbarkeit (out of Band), HW Verfügbarkeit,, …



## *Taktische vs. strategische Sicherheit*

**Ziel *taktischer Sicherheit*: Sicherung der (Kommunikations-)Inhalte für genau die Zeitspanne in der diese Inhalte für einen Angreifer von Nutzen sein können.**




- z.B. Militär: Funkverschlüsselung für eine Mission
- Aber: Kommunikationsfluss sollte dann besser keine strategischen Inhalte beinhalten bzw. Rückschlüsse auf strategische Inhalte und mehrfach Verwendete Methoden/Verfahren so wenig wie möglich zulassen. (“Angriffsformation Omega Alpha”)

**Genaue Abschätzung des Sicherheits-Spielraumes ist schwierig, d.h. die Fähigkeiten des Angreifers müssen immer deutlich überschätzt werden. Und die Vorbereitung einer Aktion ist in diesem Sinne natürlich Teil der Aktion (!).**

**Ziel *strategische Sicherheit:* Sicherheit “für immer”**

- Und damit, wie „totale Sicherheit”, nicht erreichbar

## *Transport vs. Storage Security*

**Transport Security geht immer davon aus das der Cyphertext in der Hand des Angreifers *ist***

- wird derzeit mit Public Key, Symmetrischen oder One Time Pad Verfahren realisiert
- muss i.d.R. weniger lange halten als Storage Security

**Storage Security geht immer davon aus das Verschlüsselung die letzte Barriere ist *falls* der Container dem Angreifer in die Hände fällt**

- derzeit nur auf der Basis algorithmischer Crypto (Public Key oder Symmetrisch) sinnvoll zu realisieren
- Cryptocontainer enthalten oft wesentlich brisanteres Material als verschlüsselte Nachrichten (z.B. Nachrichtensammlungen, strategisch bedeutsame Dokumente und Notizen, Planungsunterlagen, Finanzdaten etc.)

## *Realitätsabgleich: Worst / Best Practices*

### Transport Security

- Firmen: Keine (für eMail), Defaultverfallsdaten, PGP, S/MIME, …
- Industrie-Standardsoftware: Wennüberhauptkrypto: 56 Bit
- Hacker-”Standard”: 2048/1024 DSA/DSS Key mit gpg bzw. PGP
- Militär: Verschlüsselung per Hand mit zwei unabhängig generierten OTP, danach noch mal durch einen vertrauenswürdigen symmetrischen Maschinen-Cipher mit langer Schlüssellänge

### Storage Security

- Firmen: keine oder MS-Dokumenten”sicherung”
- Industrie: 56bit Datenbank-Encryption
- Hacker-”Standard”: PGP-Disk, AES-Volume, CFS o.ä.
- Militär: Bunker und bewaffnete Wachposten

## *Quantenkryptoanalyse – Das Ende ist nah!*

**768 Bit ought to be enough for everybody…**

**... until Twinkle came around and RSA512 was considered unsafe at any speed**

- Twinkle ist eine elektrooptische Maschine, die sehr sieb-basierte Faktorisierung durchführen kann.
- RSA mit 512 Bit kann in 9-10 Wochen von 20 Twinkles geknackt werden
- Das war 1999 (!)

**Exotische Technologien wie Quantencomputing und Schneller-Als-Licht-Signalisierung lassen bisherige Sicherheitsabschätzungen für Keylängen, die auf Voraussagen von Rechengeschwindigkeiten basieren zumindest fragwürdig erscheinen.**

- Verdächtiges Ausbleiben von Publikationen in den relevanten Gebieten in den letzten zwei Jahren
- Massive Investments in Forschung an derartigen Technologien

## *Exkurs: Warum auch Deine verschlüsselte Mail auch noch in 30 Jahren gegen Dich verwendet werden wird*

**Sie speichern alles**

- zur Trafficanalyse
- weil Nachrichten auch einen Wert haben, wenn man sie erst „später“ entschlüsselt
  - Venona
  - Fish
- weil es sich immer lohnt auf einen Operator-Fehler zu warten
- weil man Technologieentwicklung eben nicht vorhersagen kann (Disruptive Technologies)
  - Entschlüsselungsmöglichkeiten (Technologie / Verfahren) sind von Moores Law abgekoppelt (!)

***Deine* verschlüsselte Mail ist nicht so sicher wie *Deine Passphrase* oder *Deine Schlüssellänge*, sondern so sicher wie der *Empfänger* sie speichert oder quoted…**

## *Conclusio*

**Sicherheit hat immer eine Halbwertszeit**

**Taktische Sicherheit muss und kann auch nur das Ziel sein**

- Die wesentliche Frage ist jedoch: über welchen Zeitraum sprechen wir?
  - Verjährungsfrist(en)
  - Persönliche Lebensspanne
  - Familienlebensspanne
  - Organisationslebensspanne

**Geheimhaltung der Entschlüsselungskapazitäten verlängert manchmal die Frist bis zum Akutwerden des Problems (Dienste lesen lieber weiter mit als jemanden anzuklagen und entschlüsselte Nachrichten als Beweismaterial vorzulegen. Bsp. Enigma).**

**Aber: Generierung von Beweismaterial das die eigentliche Informationsquelle nicht kompromittiert ist ein altes, oft geübtes Spiel**

**... and beware of Communication Metadata**