# Grenzwertbetrachtungen von Sicherheit

**Daniel Ettle**

dan@deam.org

05.06.2oo3

## Einstimmung

- Keine howto - keine technischen Details (fast fluctuations)
- Foerdern des Sicherheitsbewusstseins
- Vertrauenswuerdiger Umgang mit Daten Dritter
- Paranoia Bestaetigung

## Uebersicht

- Haerten von Betriebsystemen (15min.)
- social engineering (15min.)
- sicheres loeschen (23min)
- Die zeitliche Dimension der Sicherheit (42min)

## Haerten von Betriebsystemen

My Computer is my castle

## Begriffsklaerung

kommt von dem englischen begriff 'harden', etwas verstaerken/veredeln.
erstmals aufgetaucht 3000 v.Chr. im zusammenhang mit metallverarbeitung.

## default install von Betriebsystem

- aeltere Linux-Distributionen (SuSE, RedHat, ...) => vielzahl von diensten (finger, nfs, ...)
- windows?!? - IIS, IE, RPC, COM+, spooler, IrDa... ?!?? NT $->$ 2k $->$ XP
- positives Beispiel: BSD, OpenBSD startet nur ssh.

## problematisch

- hintertueren der Programmierer (matrix)
- Open Source code reviews vs. black box binary

## harden Windows 2K/XP

- IPD Pedestial (kernel haertung, einziges am markt)
- unnoetige dienste abstellen
- abstellen von automatischen updates
- ...

## harden Linux (example)

- Installieren von CD-ROM
- Lesen aller sicherheitshowto's fuer die zu inst. programme
- einspielen aller patches
- compilieren und installieren eines eigenen kernel
- zugriffsrechte auf das dateisystem einschraenken
- unnoetige dienste abschalten
- keine default config dateien verwenden
- test und audit von exploits
- log-server einrichten
- IDS installieren
- zusaetzliche sicherheitstools (siehe spaeter)

## harden OpenBSD (paranoia-mode)

- eigenen kernel bauen
- kein X installieren
- user anzahl minimieren
- tcp-wrapper installieren
- dateisysteme nur read only mounten (mount -ur)
- einschraenken der ausfuehrbaren dateien mit noexec, nosuid und nodev (mount -uw -o nodev,nosuid,noexec /var)
- motd veraendern und warnung ausgeben.
- auslagern von binaries auf CD-ROM

## sicherheitstools

*A*uditing Source

- flawfinder
- rats
- splint

*c*hecking Integrity

- tripwire, ...
- MS hat da auch was

*A*uditing System

- nessus, nmap, satan, ...
- lanscaner

of course: IDS, FW, proxys, ...

## fortlaufender prozess (one step beyond...)

- einspielen sicherheitsupdate
- auswerten von logfiles
- lesen von security lists
- befragen des CERT

Infos:

- http://www.geodsoft.com/howto/harden/
- http://defcon1.org/
- http://www.sans.org/
- http://www.bastille-linux.org/
- http://www.linux-sec.net

## social engeniering

Was ist das, warum wird so wenig darueber gesprochen und was koennen wir dagegen tun?

Definition: Ein aussenstehender Hacker nutzt psychologische Tricks, um Informationen von legitimen Benutzern eines Computer-Systems zu erlangen, welche ihm den Zugriff auf dieses System ermöglichen.

Erstmals 1991 beim CERT erwaehnt.

'Opfer' reden nicht darueber, Firmen sowieso nie.

social engineering kann jeden treffen!

## Beispiele

- Unterhaltung in der Kneipe nach der Arbeit mit Kollegen
- Benutzung eines Laptops im Zug
- offenhalten einer Tuer anstelle die Benutzung der Ausweiskarte
- routinemaessiges verhalten (alltagstrott)
- well known visuals (ja/nein/sind sie sicher? [weiter >>])
- instant messaging, emails

## Formen von social engineering

**computer based**

- klassischer trojaner

beispiele: re-enter password, shutdown-messages, ...

## Formen von social engineering

**human based**

- sammeln von hintergrundinformationen (oeffentlicher dienstplan, tuerbeschriftungen, website, ...)
- 'trashing', 'dumpster diving' - (nicht illegal, muell ist weder privat noch eigentum)

beispiel: interne Telephonbuecher, alte Dienstplaene, Kalender, Notizbuecher, Ausdrucke von heiklen Dokumenten, Urlaubsplaene, System Handbuecher, Ausdrucke von Benutzernamen und Paßwoertern, Ausdrucke von source codes, Disketten, Backup Kassetten, alte Hardware...

## Formen von social engineering

**Reverse Social Engineering**

- Social Engineer tritt als fiktive Autoritaetsperson auf
- sabotage des Netzwerks (ueberlast erzeugen, ...)
- behauptung er behebe das problem

## Vorkomnisse

- Vorsicht vor Pseudo-eBay (Update) [18.01.2002]

Heise News-Ticker
Ein Spammer gibt sich derzeit als Mitarbeiter des Online-Auktionshauses eBay aus und versucht so, an Kreditkartennummern und andere persönliche Daten heranzukommen. In seiner Massen-Mail bestätigt er den Kauf eines erfundenen Artikels und gibt als Stornierungsseite www.ebay.com.rr.nu/ an. Dort verlangt ein Web-Formular, über dem das eBay-Logo platziert ist, alte und neue Kreditkartendaten. Auf den zweiten Blick entlarvt sich die Seite aber schnell als Kostenlos-Webspace.

- CERT Incident Note IN-2002-03

(Social Engineering Attacks via IRC and Instant Messaging) You are infected with a virus that lets hackers get into your machine and read ur files, etc. I suggest you to download [malicious url] and clean ur infected machine. Otherwise you will be banned from [IRC network].

- DEAL MIT DER CIA [27.05.2003]

Der General der Republikanischen Garde, Maher Sufian al-Tikriti, wurde mit 25 Mio. Dollar von der CIA gekauft um die gegenwehr bei der uebernaehme Bagdad zu erleichtern.

## Know your enemy

- Der Forscher (wie funktioniert das)
- Der unzufriedene Mitarbeiter aka 'the MI6 problem'
- Der Spion
- Der Terrorist
- Der Dieb
- Der Hactivist (online streik, ...)
- Das Script Kiddie
- Hacker for Hire (hackgruppen oder einzelne personen, meistens osteuropaeische staaten)
- Der Wettbewerb (rivalisierende Firmen beschiessen sich gegenseitig mit exploits)

## Warum?

- Geld
- Vandalismus/Verrueckte
- Rache
- Arbeitsplatzsicherheit / milking clients
- Terroristen
- Spionage
- Krieg

'If you cant steal it, buy it!' (umkehrbar)

## vorsichtige Massnahmen

- investition in sicherheit (roi - money for noting?)
- faehige, kompetente Angestellte
- over-security
- Penetrations Tests
- M&M syndrom (aussen hart, innen weich)
- Policies
- incident handling

# Sicher loeschen

Digitale Forensik

Problem

- Speicher von wichtigen Daten schnell und/oder zuverlaessig loeschen
- loeschen ohne rekunstruibarkeit der Daten

Einsatz bei

- Hackerangriffen
- Gesetzesuebertretungen
- Beschaffung von Informationen

## Festplatten

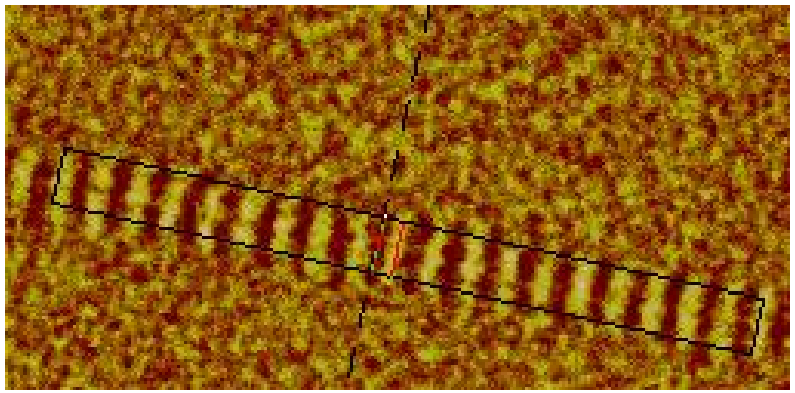
- kopf faehrt nicht exakt ueber die selbe spur (spurlage)
- hohe anzahl von fehlerkorrektursektoren bei modernen platten
- restmagnetismus in den scheiben
- software?!?

abhilfe bulk-erasor?!

# Bulk Erasor

- schaffen 2/3 von dem was man braucht
- tape/harddrive lesbar != konstruierbar

die ganze platte? ...nein ein kleines...



Laserabtastung der Oeberflaeche.

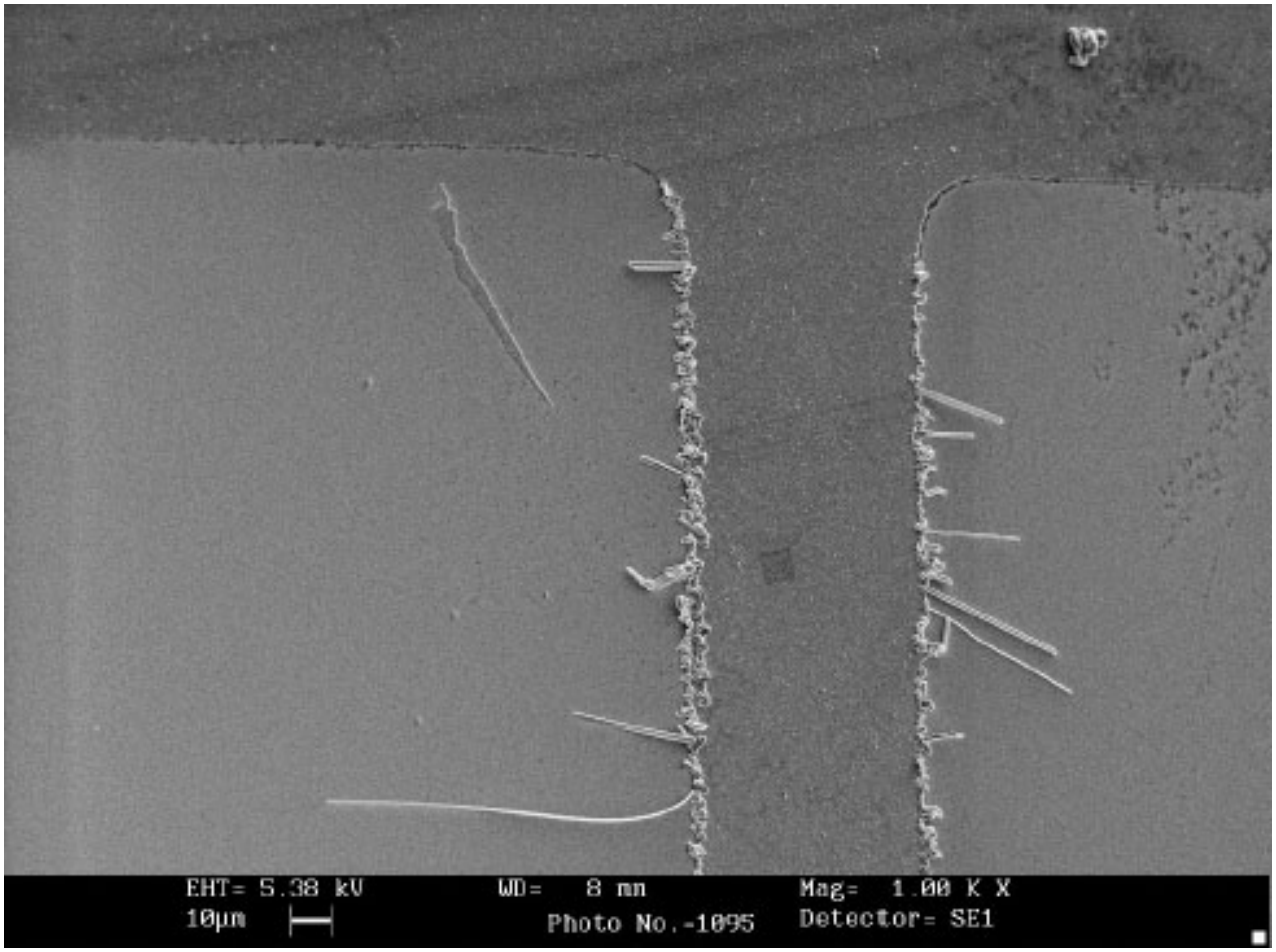
## RAM

Idee: bischen Computer, bischen RAM, bischen akku.
fluechtiger speicher, strom weg, alles weg?!

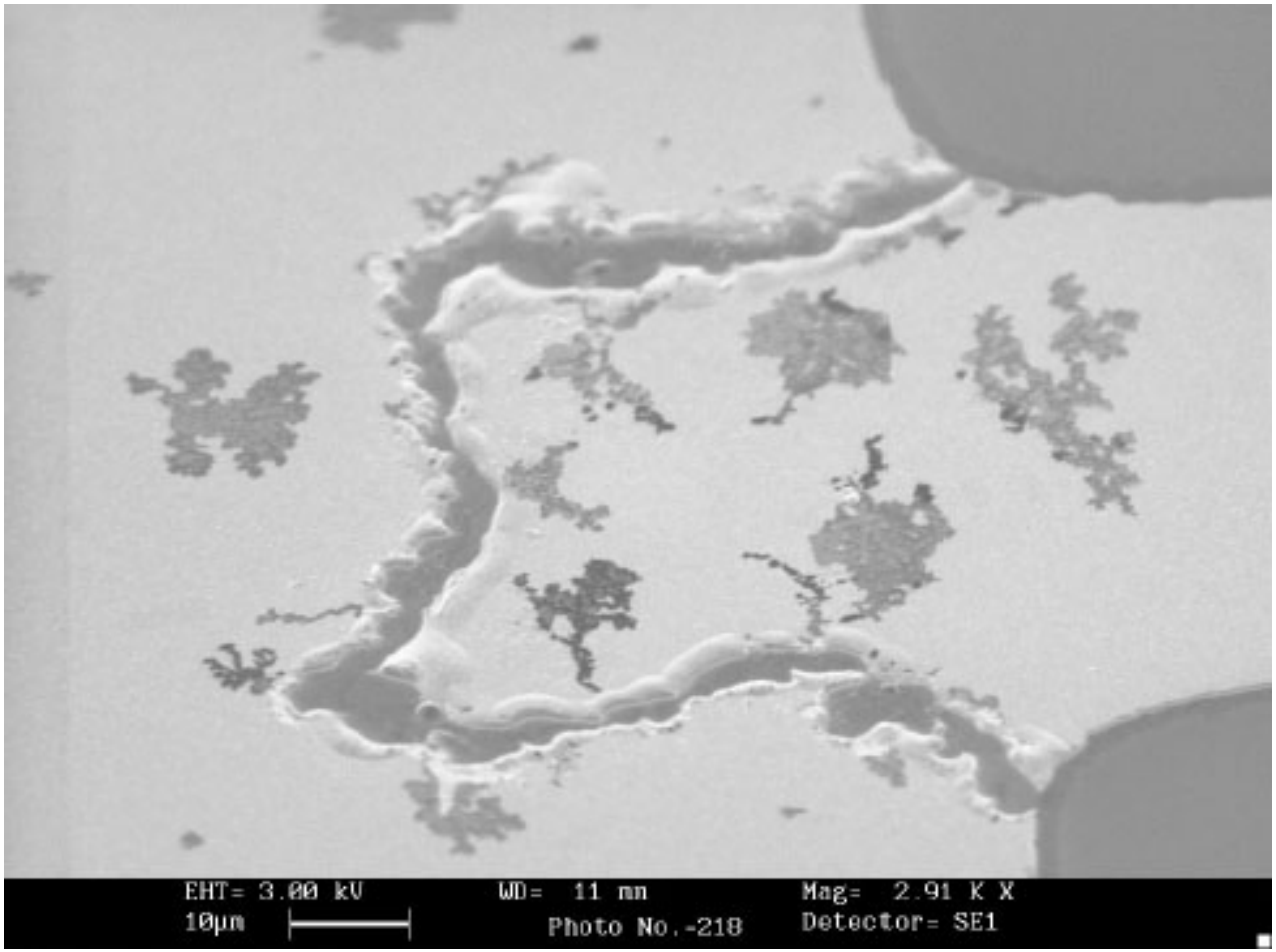
das boese:
IBM T.J.Watson Research Center

der boese:
Peter Gutman 'Data Remanence in Semiconductor Devices'

# Sicher loeschen



# Sicher loeschen



## bit-flipping

'...memory bit-flipping for burn in prevention...'
GPG release note 2002-07

aufenthaltswahrscheinlichkeit eines 0-bits gleich der eines 1-bits.
3-4 facher speicherbedarf

## Flash

vorteil:

- keine akkus
- schoen klein

nachteil

- 10 bis 23mal mit random ueberschreiben.
- burn in effekte
- interne speicherverwaltung?!?
- haufen bits

## CD-RW

siehe festplatten.

. . . randbereiche werden nicht erfasst

## Loesungsansatz

der gute:
Peter Gutman 'Secure Deletion of Data from Magnetic and Solid-State Memory'

'overwriting a drive 35 times with varying hexadecimal values may force the write head to vary magnetic effect on the iron oxide particles to such an extent as to remove the shadow data. Still, there is no guarantee that software solutions will effectively wipe out all this information because the process relies on the drive's controller, which is not suited for this purpose.'

## Beispiel pattern fuer loeschen einer festplatte

- Pass 1: Pattern is cryptographically secure random sequence. (Produced by the ISAAC algorithm)
- Pass 2: Pattern is all zeros (00h)
- Pass 3: Pattern is all ones (FFh)
- Pass 4: Pattern is all zeros (00h)
- Pass 5: Pattern is all ones (FFh)
- Pass 6: Pattern is all zeros (00h)
- Pass 7: Pattern is IBAS, with a header similar to the one used for level 1.

kann aber schon mal eine weile dauern.

**also. . .**

$=>$ existiert kein wiederbeschreibes geraet das frei von datenrueckstaenden ist.

einzige moeglichkeit $- >$ zerstoerung des mediums nach der keybenutzung.

## . . . was tun?

Agent Grandpa:

- aufessen des papiers

Agent 006++:

- Flash knirscht so zwischen den zaehnen.
- CD-RW ist giftig => Agent tot.
- Festplatten?

Loesungen?!?

## Terminator

- einschmelzen
- chemisches loesungsmittel
- anstaendig viel strom
- gute strahlungsquellen
- mikrowelle
- endlich oft ueberschreiben

Problem: flughafen, transportproblem, ...

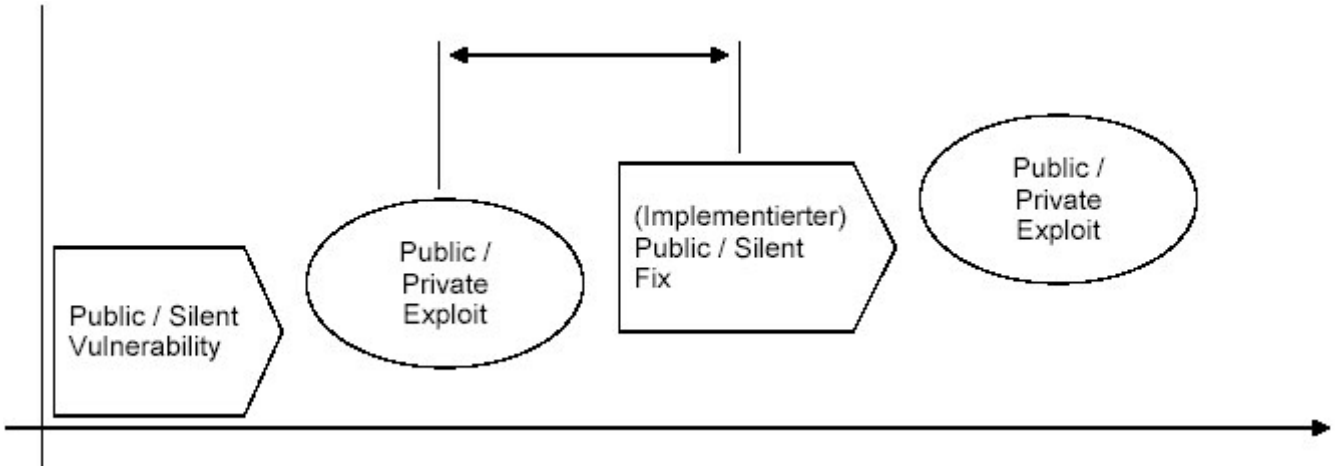
# zeitliche Dimenson der Sicherheit

Nichts ist fuer die Ewigkeit

# Was einem schnell leid tut.
# Trivitalitaetsebene 1:Minuten bis Wochen

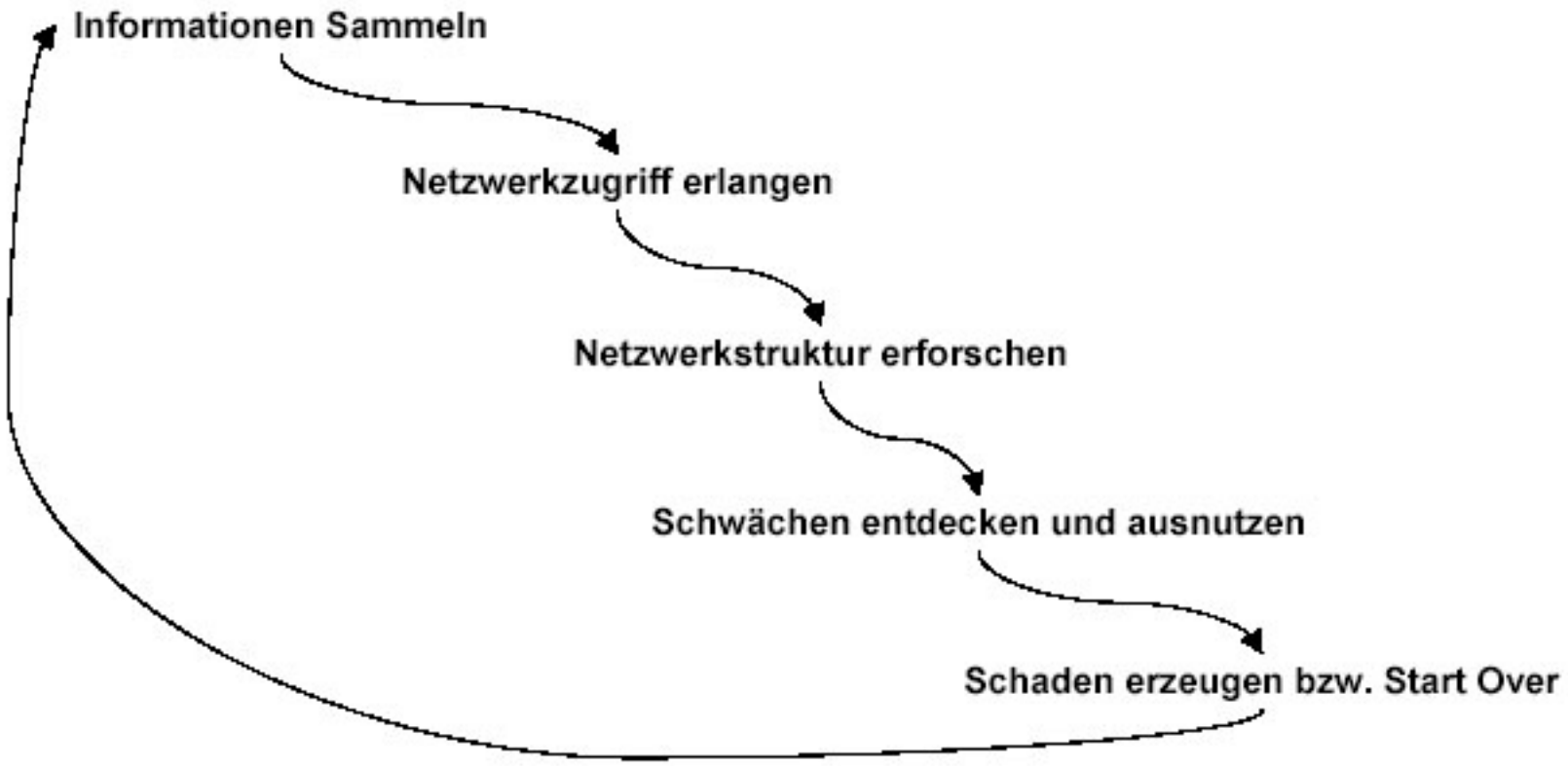
Vulnerability Window

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

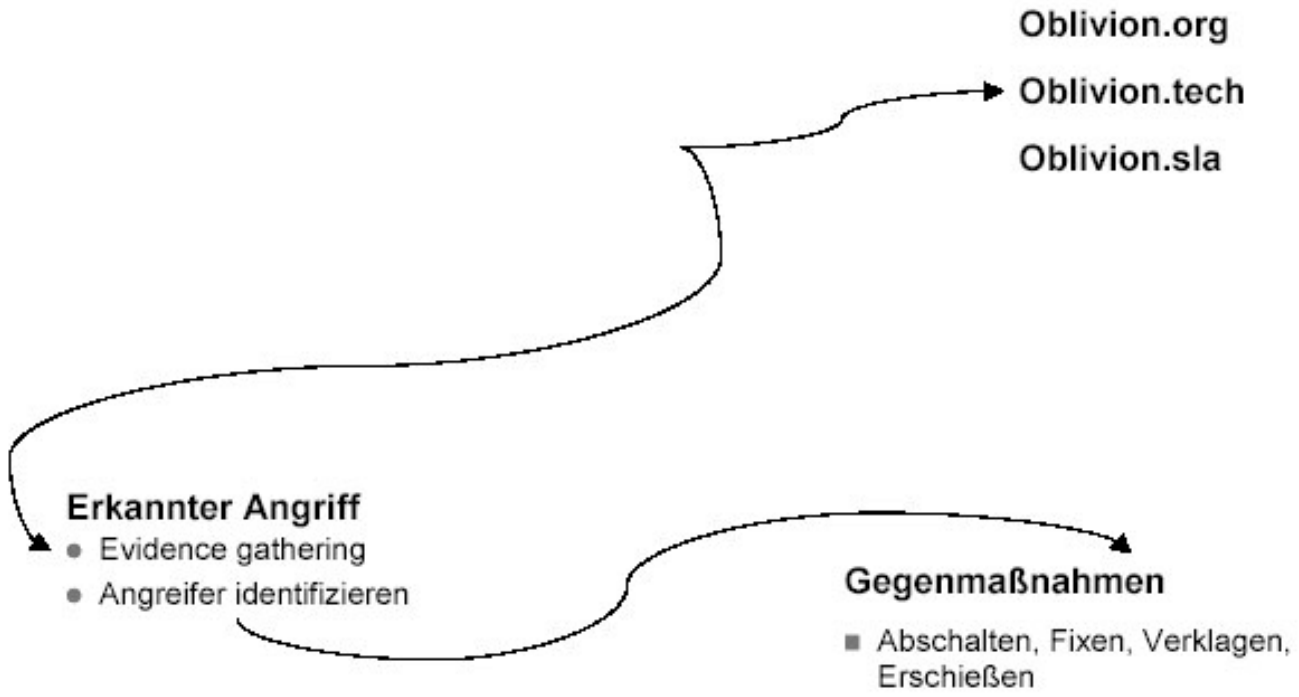



Fuer den Betrachter ohne 'Inside Sources' ist das Verwundbarkeitszeitfenster die Zeit zwischen veroeffentlichter Verwundbarkeit und (Implementation von) veroeffentlichtem Fix.

## Attack Window



## Attack DetectionWindow




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

## Qualitaet der Implementierung

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

- Auch in gut abgehangenem, viel benutztem Code koennen sich ueber Jahre hinweg
- finsterste Probleme verbergen
- Private Exploits sind weiterhin im Trend
- Durchschnittliche Dunkelperiode von Exploits ist schwer zu bestimmen
- Source Audits von motivierten und faehigen Leuten helfen, lesen allein reicht nicht. Computer unterstuetzte Source Audits sollten viel weiter verbreitet sein
- Auch Open Source Software benoetigt eine gewisse Reifephase bis sie halbwegs sicher ist

## 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 spaet ist...
## Trivitalitaetsebene 2: Jahre bis Jahrzente

Halbwertszeit der Schluessellaenge, der Implementation, ...
Was uns die Geschichte lehrt oder die Haltbarkeit von klassischer Kryptographie

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

Haltbarkeit haengt von Motivation, Faehigkeit und Ausstattung des Angreifers ab.

Beispiele:

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

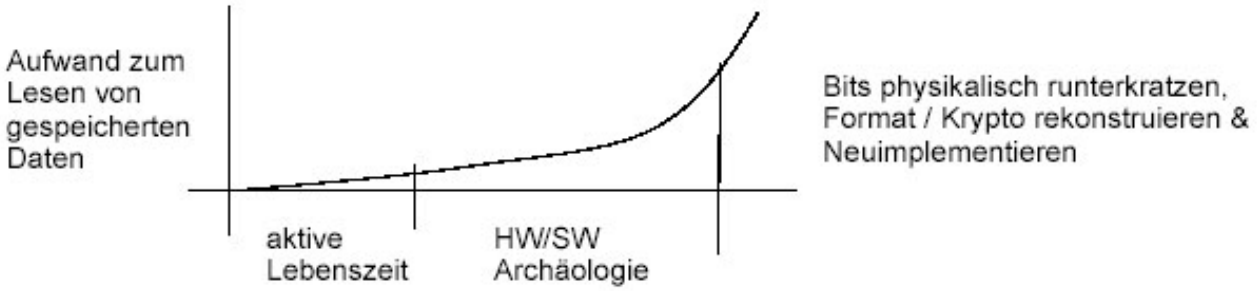
## Verfuegbarkeit - Lesbarkeit hat eine Halbwertszeit

Verfuegbarkeit von gespeicherten Informationen ist zeitlich begrenzt
Was uns die Geschichte lehrt

- Papier
- Lochkarten
- Mikrofiche
- Magnetbaender

Die Zeitspanne der Lesbarkeit sinkt mit steigender Informationsdichte

- Softwareverfuegbarkeit, Haltbarkeit des Speichermediums, Formatverfuegbarkeit (out of Band), HW Verfuegbarkeit, ...



## Taktische vs. strategische Sicherheit

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

- z.B. Militaer: Funkverschluesselung für eine Mission
- Aber: Kommunikationsfluss sollte dann besser keine strategischen Inhalte beinhalten bzw. Rueckschluesse auf strategische Inhalte und mehrfach Verwendete Methoden/Verfahren so wenig wie moeglich zulassen. ('Angriffsformation Omega Alpha')

Genaue Abschaetzung des Sicherheits-Spielraumes ist schwierig, d.h. die Faehigkeiten des Angreifers muessen immer deutlich überschaetzt werden. Und die Vorbereitung einer Aktion ist in diesem Sinne natuerlich Teil der Aktion (!).

Ziel strategische Sicherheit: Sicherheit 'fuer 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 Verschluesselung die letzte Barriere ist falls der Container dem Angreifer in die Haende fällt

- derzeit nur auf der Basis algorithmischer Crypto (Public Key oder Symmetrisch) sinnvoll zu realisieren
- Cryptocontainer enthalten oft wesentlich brisanteres Material als verschluesselte Nachrichten

## Realitaetsabgleich: Worst / Best Practices

**Transport Security**

- Firmen: Keine (fuer eMail), Defaultverfallsdaten, PGP, S/MIME, ...
- Industrie-Standardsoftware: Wennueberhauptkrypto: 56 Bit
- Hacker-'Standard': 2048/1024 DSA/DSS Key mit gpg bzw. PGP
- Militaer: Verschluesselung per Hand mit zwei unabhaengig generierten OTP, danach noch mal durch einen vertrauenswuerdigen symmetrischen Maschinen-Cipher mit langer Schluessellaenge.

## Realitaetsabgleich: Worst / Best Practices

**Storage Security**

- Firmen: keine oder MS-Dokumenten'sicherung'
- Industrie: 56bit Datenbank-Encryption
- Hacker-'Standard': PGP-Disk, AES-Volume, CFS oder sowas
- Militaer: 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 durchfuehren kann.
- RSA mit 512 Bit kann in 9-10 Wochen von 20 Twinkles geknackt werden
- Das war 1999 (!)

## Quantenkryptoanalyse - Das Ende ist nah!

Exotische Technologien wie Quantencomputing und Schneller-Als-Licht-Signalisierung lassen bisherige Sicherheitsabschaetzungen für Keylaengen, die auf Voraussagen von Rechengeschwindigkeiten basieren zumindest fragwuerdig erscheinen.

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

## Warum auch Deine verschluesselte 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' entschluesselt
  - Venona
  - Fish

## Warum auch Deine verschluesselte Mail auch noch in 30 Jahren gegen Dich verwendet werden wird

- weil es sich immer lohnt auf einen Operator-Fehler zu warten
- weil man Technologieentwicklung eben nicht vorhersagen kann (Disruptive Technologies)
  - Entschluesselungsmöglichkeiten (Technologie / Verfahren) sind von Moores Law abgekoppelt (!)

Deine verschluesselte Mail ist nicht so sicher wie Deine Passphrase oder Deine Schlüssellaenge, sondern so sicher wie der Empfaenger sie speichert oder quoted...

## endlich

Sicherheit hat immer eine Halbwertszeit

Taktische Sicherheit muss und kann auch nur das Ziel sein
Die wesentliche Frage ist jedoch: Ueber welchen Zeitraum sprechen wir?

- Verjaehrungsfrist(en)
- Persoenliche Lebensspanne
- Familienlebensspanne
- Organisationslebensspanne

## endlich

Geheimhaltung der Entschluesselungskapazitaeten verlaengert manchmal die Frist bis zum Akutwerden des Problems (Dienste lesen lieber weiter mit als jemanden anzuklagen und entschluesselte Nachrichten als Beweismaterial vorzulegen. Bsp. Enigma).

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

## . . . letztendlich

sicher ist nur das es sicher nicht sicher ist.