

# IT-Sicherheit

- Sicherheit vernetzter Systeme -

## Kapitel 1: Einleitung



## Kap. 1: Einleitung: Motivierendes Beispiel

1. Internet Worm
  - Historischer Rückblick
  - Funktionsweise
  - Lessons Learned
2. SQL Slammer Wurm
  - Historischer Rückblick
  - Funktionsweise
  - Lessons Learned
3. Vergleich von Internet Worm und Slammer



## Historischer Rückblick I: 1988 Internet Worm

- Chronologie der Vorfälle an der University of Utah:
  - Mittwoch 2. November 1988
    - 17:01:59: Test oder Start des Wurms
    - 17:04: Maschine an der Cornell University „befallen“
    - 20:49: Wurm infiziert VAX 8600 an der Univ. Utah (cs.utah.edu)
    - 21:09: Wurm versucht von VAX aus andere Maschinen zu infizieren
    - 21:21: Load (Anzahl der rechenbereiten Prozesse) von 5
    - 21:41: Load von 7
    - 22:01: Load von 16
    - 22:06: Es können keine Prozesse mehr gestartet werden, Benutzer können sich nicht mehr anmelden
    - 22:20: Systemadministrator terminiert den Wurm Prozess
    - 22:41: Der Wurm ist zurück; Load 27
    - 22:49: System shutdown, reboot
    - 23:21: Der Wurm ist zurück; Load 37



## Internet Wurm: Globale Sicht

- Mittwoch 2. Nov. 1988
  - 17:01:59: Wurm Test oder Start
  - 21:00: Stanford University; ca. 2500 Unix Maschinen infiziert
  - 21:30: MIT infiziert
  - 22:54: University of Maryland
  - 23:00: University of California, Berkeley
  - 24:00: SRI International
- Donnerstag, 3. Nov. 1998
  - 2:00: Lawrence Livermore National Laboratory
  - 2:28: E-mail Warnung; erreicht aber die meisten nicht vor Samst. 5. Nov. 5:00
    - Wurm infiziert SUN und VAX
    - Beinhaltet DES Tabelle
    - Nutzt `.rhosts` und `host.equiv`
    - Speichert `X*` Dateien in `/tmp`
  - 5:58: Bug fix posting aus Berkeley:
    - Sendmail's `debug` Kommando deaktivieren
    - C Compiler umbenennen
    - Linker umbenennen
  - 8:00: Berkely entdeckt `finger` Mechanismus
  - 10:30: TV Teams am MIT
    - Ca. 10 % Infektionsrate am MIT (2000 Maschinen)
  - 11:00: Titel-Story in den Nachrichten:
    - Mehr als 6000 hosts im Internet infiziert (10 %)



## Internet Wurm: „How it Works“

- Wie befällt er neue Maschinen?
  - sendmail Bug (seit langem bekannt)
  - finger Bug; Buffer Overflow (nur VAX werden befallen)
  - Remote execution (rsh, rexec)
- Welche Accounts werden angegr.
  - Offensichtliche Passwörter
    - Leeres Passwort
    - Benutzername
    - Benutzername+Benutzername
    - Infos aus GECOS-String
    - Nachname
    - Nachname rückwärts
  - Build-In Wörterbuch (432 Wörter)
  - /usr/dict/words (24'474 Wörter)
  - Trusted Host Beziehung (.rhosts)
- Welche hosts werden angegriffen?
  - Maschinen in /.rhosts und /etc/host.equiv
  - .forward Datei gebrochener Accounts
  - .rhosts Datei gebr. Accounts
  - Gateways aus der Routing-Tabelle
  - Endpunkte von Point to Point Verbindungen
  - Zufällig geratene Adressen
  - Nur Sun und VAX
- Was der Wurm NICHT tut:
  - Versuchen root access zu erhalten
  - Well-known Accounts angreifen
  - Daten zerstören
  - „Zeitbomben“ zurücklassen

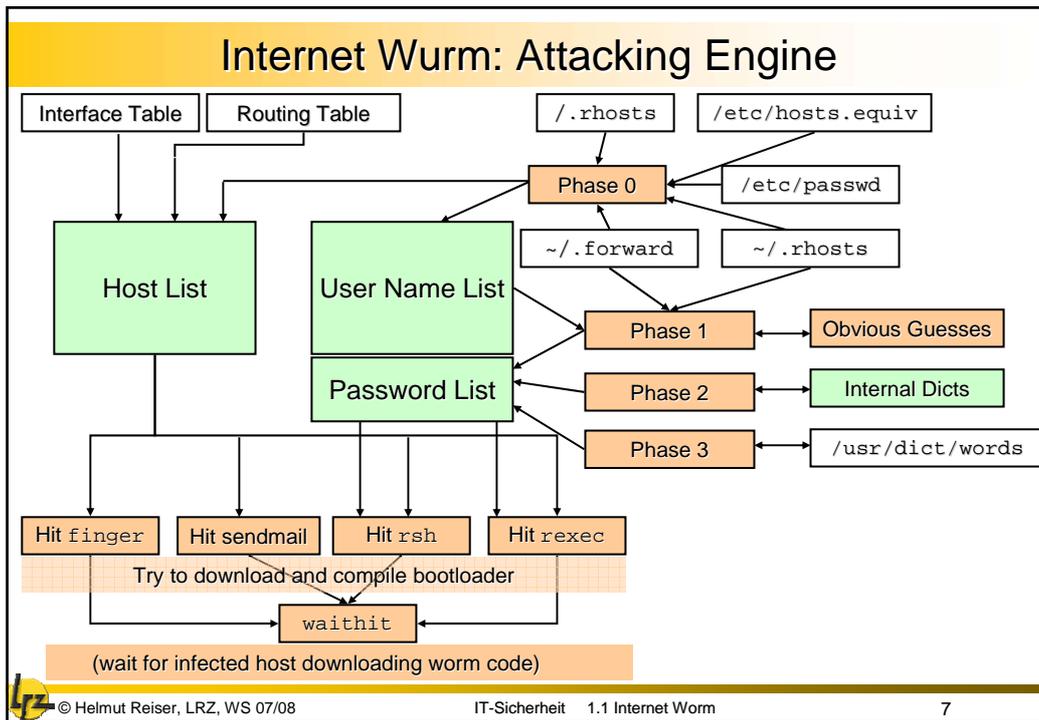


## Internet Wurm: Programm Struktur

### main Routine

```
argv[0] := "sh"; /* rename process */
Is there already a worm? /* faults here causes mass infection */
Initialize clock;
while (true) {
    cracksome(); /* attack accounts, try to find hosts */
    sleep(30); /* hide the worm */
    Listen for other worms /* faults here causes mass infection */
    create a new process, kill the old /* Camouflage */
    try to attack some machines;
    sleep(120); /* hide the worm */
    if (running > 12 hours)
        cleaning host List; /* reduce memory consumption */
    if (pleasequit && wordcheck > 10)
        exit
}
```



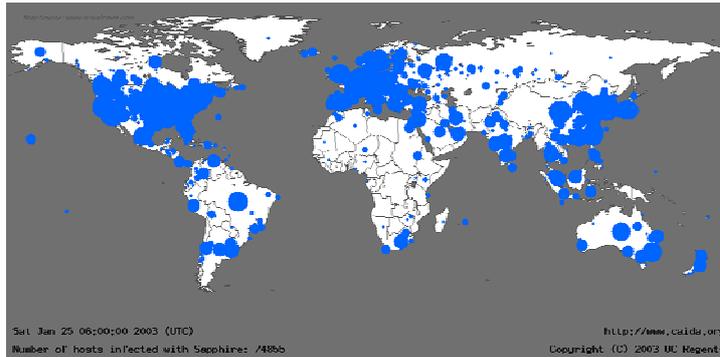


- ## Internet Wurm: Lessons Learned
- Verursacher und rechtliche Folgen
    - Robert T. Morris, 23, Cornell Student (Sohn des NSA Chief Scientist)
    - Suspendierung von der Cornell University
    - Verurteilt zu \$ 10.000 und 400 Stunden gemeinnütziger Arbeit
  - Lessons Learned
    - (lange) bekannte Bugs fixen
    - Starke Passwörter benutzen
    - Least privilege Prinzip (sowenig Rechte wie nötig), strenge Zugriffskontrolle
    - Logging und Auditing
    - Keine reflexartigen Reaktionen
    - Kontinuierliche Information von sich und anderen
    - „Zentrales“ Security Repository
      - CERT (Computer Emergency Response Team) wurde gegründet  
[www.cert.org](http://www.cert.org)
- © Helmut Reiser, LRZ, WS 07/08 IT-Sicherheit 1.1 Internet Worm 8

## Historischer Rückblick II: 2003 Slammer Wurm

### ■ Chronologie

- Samstag, 25. Januar 2003: Kurz vor 5:30 Uhr (UTC), d.h. 6:30 Uhr (MEZ) taucht der Wurm auf
- Verbreitung des Wurm um 6:00 Uhr (UTC):

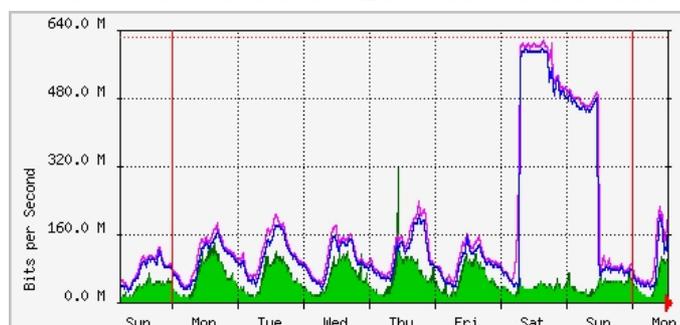


Quelle:  
MPSS 03

Kreisdurchmesser entspricht Anzahl infizierter Hosts (logarithmische Darstellung)

## SQL Slammer im Münchner Wissenschaftsnetz

- Münchner Wissenschaftsnetz (MWN), verbindet u.a. alle Standorte der Münchner Universitäten, der FH und der Bayerischen Akademie der Wissenschaften:  
Massive Störungen von Samstag 25.01.03 6:30 Uhr bis 26.01.03 11:30 Uhr
- Verkehrsstatistik am zentralen Router des MWN (1 Woche)



### ■ Legende

- Grün: eingehender Verkehr
- Blau: ausgehender Verkehr
- Dunkelgrün: Max. Peak im 5 Minuten Intervall (eingehend)
- Magenta: Max. Peak im 5 Minuten Intervall (ausgehend)

## Slammer Verbreitung und Folgen

- Schnellster Wurm in der Geschichte
  - 1. Minute: Verdopplung der Population alle 8,5 Sekunden ( $\pm 1$  s)
  - > 3 Minuten: etwas verringerte Verbreitungsrate; Netzbandbreite wird zum beschränkenden Faktor
  - 10 Minuten: ca. 90 % aller anfälligen Hosts sind infiziert
  
- Folgen:
  - Große Teile des Internets nicht mehr erreichbar
  - Steuerungssysteme für die Stromversorgung gestört
  - Funktionsstörungen bei Geldautomaten
  - Steuerrechner von zwei Atomkraftwerken in den USA betroffen
  - .....



## Slammer: Voraussetzungen

- SQL Server; Client Verbindungen über
  - NetBios (TCP Port 139/445)
  - Sockets (TCP Port 1433)
  - Monitor Port (UDP 1434) zur Ermittlung der Verbindungsart; Client schickt 0x02 an den Port; Server schickt Verbindungsinformationen
- Buffer Overflow Bug im SQL Server
  - Client setzt erstes Bit auf 0x04  
im Bsp. `\x04\x41\x41\x41\x41` (`\x41 = „A“`)
  - SQL Monitor nimmt Rest der Daten und öffnet damit Registry  
`HKLM\Software\Microsoft\Microsoft SQL Server\AAAA\MSSQLServer\CurrentVersion`
  - Über geeignet formatierte Daten kann hier ein Buffer Overflow herbeigeführt werden
- Problem:
  - SW von Drittanbietern beinhaltet SQL-Server
  - Dies ist nicht allgemein bekannt



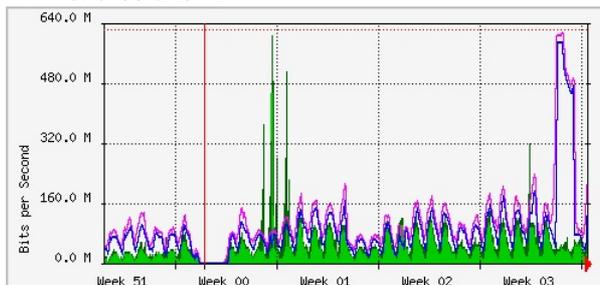
## Slammer: How it works

- Slammer passt in ein UDP Packet
  - 376 Byte groß, geschrieben in Assembler
  - Mit Header Informationen 404 Byte
- Slammer nutzt Buffer-Overflow an UDP Port 1434
- Nach Infektion:
  - „Raten“ zufälliger IP-Adressen
  - Angriff über UDP
- Keine Schadfunktionalität im eigentlichen Sinn
- Charakteristika:
  - UDP verbindungsloses Protokoll; wird nur durch Bandbreite beschränkt
  - Höchste beobachtete „Probing“-Rate: 26.000 Scans pro Sekunde
  - Aggressive Verbreitungsstrategie führt dazu, dass der Wurm mit anderen Würmern um Netzbandbreite konkurriert



## Slammer im MWN (Forts.)

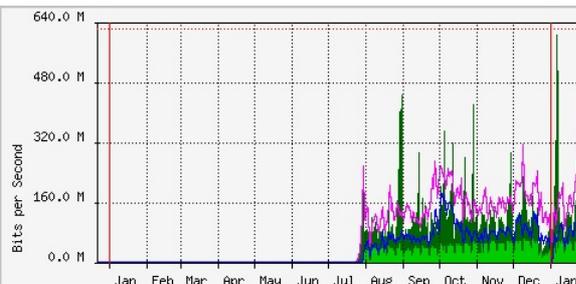
### ■ Monatsstatistik



### ■ Mind. 8 SQL-Server betroffen

### ■ Maßnahmen:

- Zugang zum MWN für diese Server gesperrt
- Port 1434 gesperrt



## Slammer: Lessons Learned

- Grundproblematik:
  - Nicht behobenen Bugs in Anwendungen (kein Einspielen von Patches)
- Bundling von Software; Anwender weiß u.U. nichts von Sicherheitsproblemen und notwendigen Patches
- Angriffe über UDP können zu extrem schneller Verbreitung führen
- Gegenmaßnahmen:
  - Filtern des entsprechenden Verkehrs (UDP Port 1434) über Firewall
  - Fehler und Schwächen beheben
  - Nicht notwendige Dienste abschalten



## Vergleich Internet Worm und ILOVEYOU

	<b>Internet Worm</b>	<b>Slammer</b>
Angegriffene Hosts/OS	SUN und VAX / UNIX	Microsoft Windows/SQL Server
Angriffsstrategie	Ziemlich komplex Nutzt eine Vielzahl von Bugs und fortschrittliche Strategien	Einfaches Assembler Programm nutzt Buffer Overflow
Schadfunktion	Verursacht große Load und viel Netzverkehr	Verursacht extremste Load und Netzverkehr
Verbreitung	~ 6.000 Systeme Ziemlich schnell	Extrem schnell 90 % aller verwundbaren Systeme nach 10 Minuten infiziert

