



## Das OpenAntivirus Projekt

<http://www.openantivirus.org/>  
Kurt Huwig <kurt@iku-netz.de>



<http://www.iku-netz.de/>



## Aufbau

- Motivation
- Bestandteile
- Status / Technik / Demonstration
- Zukunft



<http://www.iku-netz.de/>



## Warum?

- Linux Viren werden zunehmen
    - ◆ XML Virus befällt Mozilla 0.9.8
  - Linux Server werden immer beliebter
    - Virenschutz für Windows Systeme erwünscht
  - Hoher wirtschaftlicher Schaden durch Viren
    - > 3 Milliarden Schaden alleine durch CodeRed und Nimda (Quelle: Computer Economics)
- Viren sollten so gut es geht bekämpft werden



<http://www.iku-netz.de/>



## Das Problem

- hohe Lizenzkosten kommerzieller Virenschanner
  - 10-20 pro Nutzer und Jahr
- viele können sich das nicht leisten
  - Universitäten, Schulen
  - kleine Unternehmen



<http://www.iku-netz.de/>



## Die Folge

- Es werden Systeme infiziert mit
  - guter Internetanbindung
  - oftmals schlechter Wartung
  - wenig Sicherheitsbewusstsein
- ➔ rasante Ausbreitung von Viren



<http://www.iku-netz.de/>



## Können wir es schaffen?

- z.Zt. existieren > 60.000 Viren
- sehr wenige Viren sind wirklich im Umlauf
- 2001: < 10 unterschiedliche Viren auf einem Mailserver gefunden
- pro Monat etwa 5000 neue Viren
  - ◆ davon treten etwa 5 „in the wild“ auf
- Erkennen dieser Viren ist machbar



<http://www.iku-netz.de/>



## Freie Lizenz

- GNU Public License (GPL)
- Copyright von Dritten muss auf das Projekt übertragen werden



<http://www.iku-netz.de/>



## Kommerzielle Lizenz

- Lizenzen für Closed Source Projekte können gekauft werden
- Einnahmen dienen der Weiterentwicklung
- ➔ Virenschutz in vielen Produkten
- ➔ wer nicht offen sein will, muss das Projekt finanziell unterstützen



<http://www.iku-netz.de/>



## Die Bestandteile

- PatternFinder (Java)
  - Virensignaturen bestimmen
- ScannerDaemon (Java)
  - Virensan-Dienst
- VirusHammer (Java)
  - eigenständiger Virens Scanner mit GUI
- Samba-vscan (C)
  - Zugriffsscan für Samba
- Squid-vscan (C)
  - ◆ Zugriffsscan für Squid
- Linux-Kernel Modul (C)
  - ◆ Zugriffsscan



<http://www.iku-netz.de/>



## Projektstatus

- etwa 1.800 Viren werden erkannt
- seit 6 Monaten auf einem Mailserver aktiv
  - ◆ Integration in AMaViS
  - ◆ parallel mit einem kommerziellen Scanner
  - ◆ alle aktuellen Viren werden erkannt
  - ➔ Virenschutz für Mailserver ist praktikabel
- VirusHammer ist benutzbar
- Squid-vscan ist z.Zt. Proof-of-concept
  - ◆ noch nicht für Produktiveinsatz geeignet
- Samba-vscan ist benutzbar
  - ◆ läuft auf einigen Servern
- Kernel Modul ist benutzbar



<http://www.iku-netz.de/>



## Warum Java?

- plattformunabhängig
  - ◆ einziger Scanner für AS/400 PPC
- keine Buffer Overflows
- schnell



<http://www.iku-netz.de/>



## PatternFinder: das Problem

- Virens Scanner suchen nach charakteristischen Bestandteilen von Viren (Signatures)
  - ◆ sind nicht frei verfügbar
  - ◆ Erstellung erfordert Analyse des Virus



<http://www.iku-netz.de/>



## PatternFinder: die Lösung

- Voraussetzung
  - ◆ Virus
  - ◆ (kommerzieller) Virens Scanner, der ihn erkennt
- Virus wird systematisch überschrieben
  - ◆ Virens Scanner wird gefragt, ob der Virus noch vorhanden ist
    - ◆ wenn ja: weiter überschreiben
    - ◆ wenn nein: diese Stelle nicht überschreiben



<http://www.iku-netz.de/>



## PatternFinder: das Ergebnis

- Gerüst des Virus, das der Virens Scanner erkennt
- vereinfachte Analyse dieses Gerüsts
- evtl. direkte Verwendung im Scanner
- ~ 1.800 Signaturen sind bekannt



<http://www.iku-netz.de/>



## PatternFinder: Beschleunigung

- Starten eines Virens Scanners braucht Zeit
- mehrere Kopien des Virus werden unabhängig voneinander gelöscht
- alle werden bei einem Start geprüft

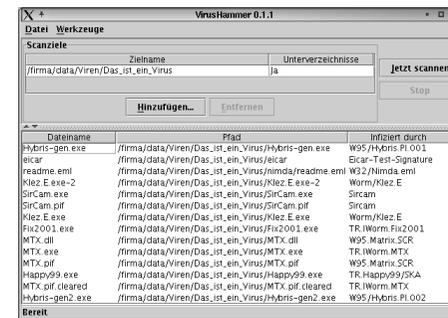


<http://www.iku-netz.de/>



## VirusHammer

- einfache Verwendung



<http://www.iku-netz.de/>



## VirusHammer

- internationalisiert
- bereits lokalisiert auf
  - ◆ Englisch, Deutsch, Französisch, Italienisch, Spanisch
- JavaWebStart: Scannen ohne Installation
- geplant
  - ◆ Integration von PatternFinder
  - ◆ besserer Fortschrittsanzeiger
  - ◆ Angabe von DOS-Laufwerksbuchstaben
  - ◆ Berichten von nicht erkannten Viren



<http://www.iku-netz.de/>



## Linux Kernel Modul

- Anbindung durch Dazuko  
(<http://www.dazuko.org/>)
  - ◆ Datei-Zugriffskontrolle per Userspace Anwendung
  - ◆ Kernel Modul benachrichtigt
  - ◆ Userspace Anwendung gibt frei oder verweigert
- triviale Anbindung an ScannerDaemon (etwa 1h Aufwand)
- nutzbar für
  - ◆ Fileserver (z.B. Samba)
- bedingt nutzbar für
  - ◆ Proxy
  - ◆ Mailserver



<http://www.iku-netz.de/>



## Squid-vscan

- verwendet Squid-Filter von Olaf Titz
  - ◆ <http://sites.inka.de/sites/bigred/devel/squid-filter.html>
- benötigt laufenden ScannerDaemon
- Problem: Meldung eines Virus
  - ◆ HTTP-Header sind bereits versendet
  - ◆ Übertragung lässt sich nur abbrechen
  - ◆ Umleitung auf eine Fehlerseite ist nicht möglich
- Andere Scanner lassen sich einbauen



<http://www.iku-netz.de/>



## Samba-vscan

- Projekt von Rainer Link
- Realisierung über Samba Virtual Filesystem
  - ◆ Patch für Samba 2.2.0 - 2.2.3a
  - ◆ Samba 3.0 benötigt keinen Patch
- Scannen aller Dateien vor dem Lesen
  - ◆ Zugriff wird bei Infektion verweigert
- Unterstützt mehrere Scanner über C-API
  - ◆ OAV ScannerDaemon
  - ◆ Sophos
  - ◆ Trend Micro
  - ◆ Kaspersky
  - ◆ Symantec



<http://www.iku-netz.de/>



## ScannerDaemon

- Starten des Scanners braucht Zeit
  - ◆ Duron 800MHz: 4s
- als Dienst entfällt das Starten
- `echo "SCAN /home/kurt" | netcat localhost 8127`
- Antwort:
  - ◆ OK
  - ◆ FOUND: <Virusname>
- Integration in
  - ◆ AmaViS
  - ◆ MIMEDefang
  - ◆ OdeiaVir
  - ◆ Samba/Squid-vscan/Kernel-Modul



<http://www.iku-netz.de/>



## ScannerDaemon: Aufbau

- Aufteilung in
  - ◆ Filter
  - ◆ Scanner
  - ◆ Finder
- Filter
  - ◆ Vorverarbeitung von Dateien
  - ◆ Extrahierung von Programmdateien (ZIP, UPX, ...)
- Scanner
  - ◆ Weiterleitung der Daten an die Finder
- Finder
  - ◆ Suche nach Viren



<http://www.iku-netz.de/>



## ScannerDaemon: Finder

- erhält Puffer mit
  - ◆ Präfix
  - ◆ Datenblock
  - ◆ Postfix
- Finder kann auf mindestens +/- 4kB zugreifen



<http://www.iku-netz.de/>



## Scan-Technik: Stringsuche

- viele (binäre) Strings müssen erkannt werden
  - auch für polymorphe oder verschlüsselte Viren
- Aho/Corasick arbeitet unabhängig von der Zahl der zu suchenden Strings
  - ◆ <http://www-sr.informatik.uni-tuebingen.de/~buehler/AC/AC.html>



<http://www.iku-netz.de/>



## Aho/Corasick Stringsuche

- erstellt einen Baum zur Erkennung
- besonderes Merkmal: Failure Function
  - ◆ bei Nichterkennung wird in einen anderen Teil des Baumes gesprungen
  - ◆ jedes Zeichen muss nur einmal in  $O(1)$  verarbeitet werden
- Beispiel
  - ◆ Suchstrings: baumhaus, auffahrt
  - ◆ Text: bauffahrt
  - ◆ Erkennung: b - a - u - f (passt nicht)
  - ◆ Sprung nach: a - u - f



<http://www.iku-netz.de/>



## Modifizierter Aho/Corasick

- Knoten im Baum verbrauchen viel Speicher
- unwahrscheinlich, dass viele Pattern lange identische Präfixe haben
- maximale Tiefe wurde auf 4 begrenzt
- danach lineare Suche



<http://www.iku-netz.de/>



## Aho/Corasick mit Tiefe 3

- Stand 06.03.2002:
  - ◆ 2.072 Knoten (inkl. Wurzel)
  - ◆ 200 Knoten mit Tiefe 1
    - ◆ Trefferwahrscheinlichkeit:  $200/256 = 78\%$
  - ◆ 855 Knoten mit Tiefe 2
    - ◆ Trefferwahrscheinlichkeit:  $855/65.536 = 1,3\%$
  - ◆ 1.016 Knoten mit Tiefe 3
    - ◆ Trefferwahrscheinlichkeit:  $1.016/16.777.216 = 0,006\%$
    - ◆ 1kB: 6%, 32kB: 86%
  - ◆ Praxis (Windows-EXE): Ein Treffer alle 74 Byte
- Duron 800MHz: 10MB/Sekunde



<http://www.iku-netz.de/>



## Aho/Corasick mit Tiefe 4

- 3178 Nodes (+53%)
- 1 Treffer alle 153 Byte
- Duron 800MHz: 12MB/Sekunde (+20%)



<http://www.iku-netz.de/>



## Aho/Corasick: Fazit

- ausreichend schnell für Internetverbindungen
  - ◆ E-Mail
  - ◆ Proxy
- Programmcode ist noch nicht optimiert
- optimierte Simulation ergab > 100MB/Sekunde



<http://www.iku-netz.de/>



## Zukunft

- Viren von den Erschaffern
  - ◆ eigenes Virenanalyseteam
  - ◆ Sourcer
  - ◆ Virenreportpolitik
- Scanner
  - ◆ Heuristiken
  - ◆ MS-Office Dateien
  - ◆ signiertes Dateiformat
  - ◆ Scannen von mehr Archiven (tar.gz, RAR...)



<http://www.iku-netz.de/>



## Zukunft

- Squid-vscan
  - ◆ Squid v2.[456]
  - ◆ Integration weiterer Virens Scanner
- VirusHammer
  - ◆ mehr Sprachen
  - ◆ bessere Fortschrittsanzeige
  - ◆ Laufwerksbuchstaben auswählbar
- Samba-vscan
  - ◆ Scan beim Schreiben
  - ◆ Quarantäne
  - ◆ Auto-clean



<http://www.iku-netz.de/>



## Zukunft

- Beta Tester
- Wohin mit Viren?
  - ◆ Mail an <kurt@iku-netz.de>



<http://www.iku-netz.de/>



## Demonstration

- VirusHammer
  - ◆ Start per Browser
  - ◆ Scannen von Verzeichnissen
- ScannerDaemon
  - ◆ Scannen per  
„echo SCAN <filename> | netcat localhost 8127“
- Squid-vscan
  - ◆ Zugriffsverweigerung auf Viren im WWW
- Kernel Modul
  - ◆ Zugriffsverweigerung bei „normalen“ Zugriffen
- Samba-vscan
  - ◆ Zugriffsverweigerung beim Clientzugriff



<http://www.iku-netz.de/>



## Fragen?

□ <http://www.openantivirus.org/>  
□ <http://www.iku-netz.de/>  
□ Kurt Huwig <kurt@iku-netz.de>



<http://www.iku-netz.de/>



# Das OpenAntivirus Projekt

`http://www.openantivirus.org/  
Kurt Huwig <kurt@iku-netz.de>`





# Aufbau

- Motivation
- Bestandteile
- Status / Technik / Demonstration
- Zukunft



# Warum?

- Linux Viren werden zunehmen
    - ◆ XML Virus befällt Mozilla 0.9.8
  - Linux Server werden immer beliebter
    - Virenschutz für Windows Systeme erwünscht
  - Hoher wirtschaftlicher Schaden durch Viren
    - > 3 Milliarden Schaden alleine durch CodeRed und Nimda (Quelle: Computer Economics)
- ➔ Viren sollten so gut es geht bekämpft werden



# Das Problem

- hohe Lizenzkosten kommerzieller Virens Scanner
  - 10-20 € pro Nutzer und Jahr
- viele können sich das nicht leisten
  - Universitäten, Schulen
  - kleine Unternehmen



# Die Folge

- Es werden Systeme infiziert mit
  - guter Internetanbindung
  - oftmals schlechter Wartung
  - wenig Sicherheitsbewusstsein
- ➔ rasante Ausbreitung von Viren



# Können wir es schaffen?

- z.Zt. existieren  $> 60.000$  Viren
- sehr wenige Viren sind wirklich im Umlauf
- 2001:  $< 10$  unterschiedliche Viren auf einem Mailserver gefunden
- pro Monat etwa 5000 neue Viren
  - ◆ davon treten etwa 5 „in the wild“ auf
- Erkennen dieser Viren ist machbar



# Freie Lizenz

- GNU Public License (GPL)
- Copyright von Dritten muss auf das Projekt übertragen werden



# Kommerzielle Lizenz

- Lizenzen für Closed Source Projekte können gekauft werden
- Einnahmen dienen der Weiterentwicklung
  - ➔ Virenschutz in vielen Produkten
  - ➔ wer nicht offen sein will, muss das Projekt finanziell unterstützen



# Die Bestandteile

- PatternFinder (Java)
  - Virensignaturen bestimmen
- ScannerDaemon (Java)
  - Virensan-Dienst
- VirusHammer (Java)
  - eigenständiger Virens Scanner mit GUI
- Samba-vscan (C)
  - Zugriffsscan für Samba
- Squid-vscan (C)
  - ◆ Zugriffsscan für Squid
- Linux-Kernel Modul (C)
  - ◆ Zugriffsscan



# Projektstatus

- etwa 1.800 Viren werden erkannt
- seit 6 Monaten auf einem Mailserver aktiv
  - ◆ Integration in AMaViS
  - ◆ parallel mit einem kommerziellen Scanner
  - ◆ alle aktuellen Viren werden erkannt
- Virenschutz für Mailserver ist praktikabel
- VirusHammer ist benutzbar
- Squid-vscaan ist z.Zt. Proof-of-concept
  - ◆ noch nicht für Produktiveinsatz geeignet
- Samba-vscaan ist benutzbar
  - ◆ läuft auf einigen Servern
- Kernel Modul ist benutzbar



# Warum Java?

- plattformunabhängig
  - ◆ einziger Scanner für AS/400 PPC
- keine Buffer Overflows
- schnell



# PatternFinder: das Problem

- Virens Scanner suchen nach charakteristischen Bestandteilen von Viren (Signaturen)
  - ◆ sind nicht frei verfügbar
  - ◆ Erstellung erfordert Analyse des Virus



# PatternFinder: die Lösung

- Voraussetzung
  - ◆ Virus
  - ◆ (kommerzieller) Virens Scanner, der ihn erkennt
- Virus wird systematisch überschrieben
  - ◆ Virens Scanner wird gefragt, ob der Virus noch vorhanden ist
    - ◆ wenn ja: weiter überschreiben
    - ◆ wenn nein: diese Stelle nicht überschreiben



# PatternFinder: das Ergebnis

- Gerüst des Virus, das der Virens Scanner erkennt
- vereinfachte Analyse dieses Gerüsts
- evtl. direkte Verwendung im Scanner
- ~ 1.800 Signaturen sind bekannt



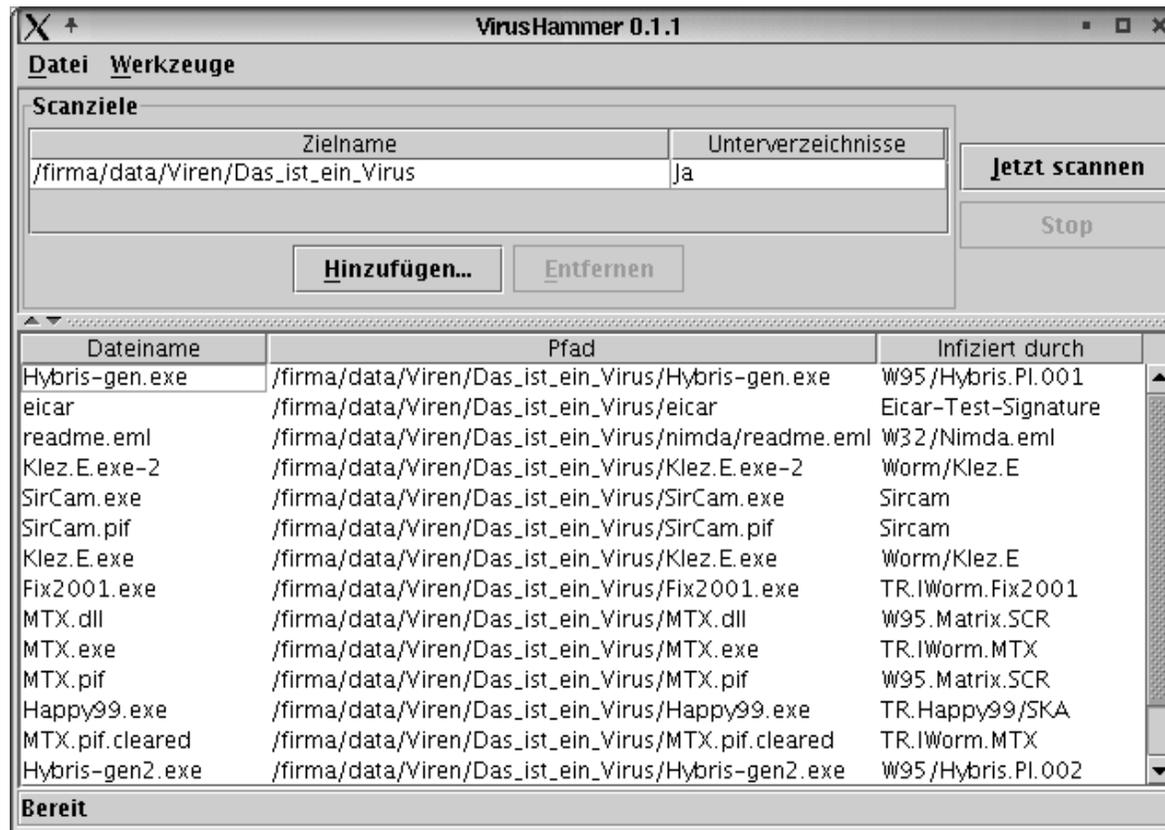
# PatternFinder: Beschleunigung

- Starten eines Virens scanners braucht Zeit
- mehrere Kopien des Virus werden unabhängig voneinander gelöscht
- alle werden bei einem Start geprüft



# VirusHammer

- einfache Verwendung





# VirusHammer

- internationalisiert
- bereits lokalisiert auf
  - ◆ Englisch, Deutsch, Französisch, Italienisch, Spanisch
- JavaWebStart: Scannen ohne Installation
- geplant
  - ◆ Integration von PatternFinder
  - ◆ besserer Fortschrittsanzeiger
  - ◆ Angabe von DOS-Laufwerksbuchstaben
  - ◆ Berichten von nicht erkannten Viren



# Linux Kernel Modul

- Anbindung durch Dazuko  
(<http://www.dazuko.org/>)
  - ◆ Datei-Zugriffskontrolle per Userspace Anwendung
  - ◆ Kernel Modul benachrichtigt
  - ◆ Userspace Anwendung gibt frei oder verweigert
- triviale Anbindung an ScannerDaemon (etwa 1h Aufwand)
- nutzbar für
  - ◆ Fileserver (z.B. Samba)
- bedingt nutzbar für
  - ◆ Proxy
  - ◆ Mailserver



# Squid-vscan

- verwendet Squid-Filter von Olaf Titz
  - ◆ <http://sites.inka.de/sites/bigred/devel/squid-filter.html>
- benötigt laufenden ScannerDaemon
- Problem: Meldung eines Virus
  - ◆ HTTP-Header sind bereits versendet
  - ◆ Übertragung lässt sich nur abbrechen
  - ◆ Umleitung auf eine Fehlerseite ist nicht möglich
- Andere Scanner lassen sich einbauen



# Samba-vscan

- Projekt von Rainer Link
- Realisierung über Samba Virtual Filesystem
  - ◆ Patch für Samba 2.2.0 - 2.2.3a
  - ◆ Samba 3.0 benötigt keinen Patch
- Scannen aller Dateien vor dem Lesen
  - ◆ Zugriff wird bei Infektion verweigert
- Unterstützt mehrere Scanner über C-API
  - ◆ OAV ScannerDaemon
  - ◆ Sophos
  - ◆ Trend Micro
  - ◆ Kaspersky
  - ◆ Symantec



# ScannerDaemon

- Starten des Scanners braucht Zeit
  - ◆ Duron 800MHz: 4s
- als Dienst entfällt das Starten
- `echo "SCAN /home/kurt" | netcat localhost 8127`
- Antwort:
  - ◆ OK
  - ◆ FOUND: <Virusname>
- Integration in
  - ◆ AmaViS
  - ◆ MIMEDefang
  - ◆ OdeiaVir
  - ◆ Samba/Squid-vscan/Kernel-Modul



# ScannerDaemon: Aufbau

- Aufteilung in
  - ◆ Filter
  - ◆ Scanner
  - ◆ Finder
- Filter
  - ◆ Vorverarbeitung von Dateien
  - ◆ Extrahierung von Programmdateien (ZIP, UPX, ...)
- Scanner
  - ◆ Weiterleitung der Daten an die Finder
- Finder
  - ◆ Suche nach Viren



# ScannerDaemon: Finder

- erhält Puffer mit
  - ◆ Präfix
  - ◆ Datenblock
  - ◆ Postfix
- Finder kann auf mindestens +/- 4kB zugreifen



# Scan-Technik: Stringsuche

- viele (binäre) Strings müssen erkannt werden
  - auch für polymorphe oder verschlüsselte Viren
- Aho/Corasick arbeitet unabhängig von der Zahl der zu suchenden Strings
  - ◆ <http://www-sr.informatik.uni-tuebingen.de/~buehler/AC/AC.html>



# Aho/Corasick Stringsuche

- erstellt einen Baum zur Erkennung
- besonderes Merkmal: Failure Function
  - ◆ bei Nichterkennung wird in einen anderen Teil des Baumes gesprungen
  - ◆ jedes Zeichen muss nur einmal in  $O(1)$  verarbeitet werden
- Beispiel
  - ◆ Suchstrings: baumhaus, auffahrt
  - ◆ Text: bauffahrt
  - ◆ Erkennung: b - a - u - f (passt nicht)
  - ◆ Sprung nach: a - u - f



# Modifizierter Aho/Corasick

- Knoten im Baum verbrauchen viel Speicher
- unwahrscheinlich, dass viele Pattern lange identische Präfixe haben
- maximale Tiefe wurde auf 4 begrenzt
- danach lineare Suche



# Aho/Corasick mit Tiefe 3

- Stand 06.03.2002:
  - ◆ 2.072 Knoten (inkl. Wurzel)
  - ◆ 200 Knoten mit Tiefe 1
    - ◆ Trefferwahrscheinlichkeit:  $200/256 = 78\%$
  - ◆ 855 Knoten mit Tiefe 2
    - ◆ Trefferwahrscheinlichkeit:  $855/65.536 = 1,3\%$
  - ◆ 1.016 Knoten mit Tiefe 3
    - ◆ Trefferwahrscheinlichkeit:  $1.016/16.777.216 = 0,006\%$
    - ◆ 1kB: 6%, 32kB: 86%
  - ◆ Praxis (Windows-EXE): Ein Treffer alle 74 Byte
- Duron 800MHz: 10MB/Sekunde



# Aho/Corasick mit Tiefe 4

- 3178 Nodes (+53%)
- 1 Treffer alle 153 Byte
- Duron 800MHz: 12MB/Sekunde (+20%)



# Aho/Corasick: Fazit

- ausreichend schnell für Internetverbindungen
  - ◆ E-Mail
  - ◆ Proxy
- Programmcode ist noch nicht optimiert
- optimierte Simulation ergab  $> 100\text{MB/Sekunde}$



# Zukunft

- Viren von den Erschaffern
  - ◆ eigenes Virenanalyseteam
  - ◆ Sourcer
  - ◆ Virenreportpolitik
- Scanner
  - ◆ Heuristiken
  - ◆ MS-Office Dateien
  - ◆ signiertes Dateiformat
  - ◆ Scannen von mehr Archiven (tar.gz, RAR...)



# Zukunft

- Squid-vscan
  - ◆ Squid v2.[456]
  - ◆ Integration weiterer Virens Scanner
- VirusHammer
  - ◆ mehr Sprachen
  - ◆ bessere Fortschrittsanzeige
  - ◆ Laufwerksbuchstaben auswählbar
- Samba-vscan
  - ◆ Scan beim Schreiben
  - ◆ Quarantäne
  - ◆ Auto-clean



# Zukunft

- Beta Tester
- Wohin mit Viren?
  - ◆ Mail an <kurt@iku-netz.de>



# Demonstration

- VirusHammer
  - ◆ Start per Browser
  - ◆ Scannen von Verzeichnissen
- ScannerDaemon
  - ◆ Scannen per

```
„echo SCAN <filename> | netcat localhost 8127“
```
- Squid-vscan
  - ◆ Zugriffsverweigerung auf Viren im WWW
- Kernel Modul
  - ◆ Zugriffsverweigerung bei „normalen“ Zugriffen
- Samba-vscan
  - ◆ Zugriffsverweigerung beim Clientzugriff



# Fragen?

- <http://www.openantivirus.org/>
- <http://www.iku-netz.de/>
- Kurt Huwig <kurt@iku-netz.de>

