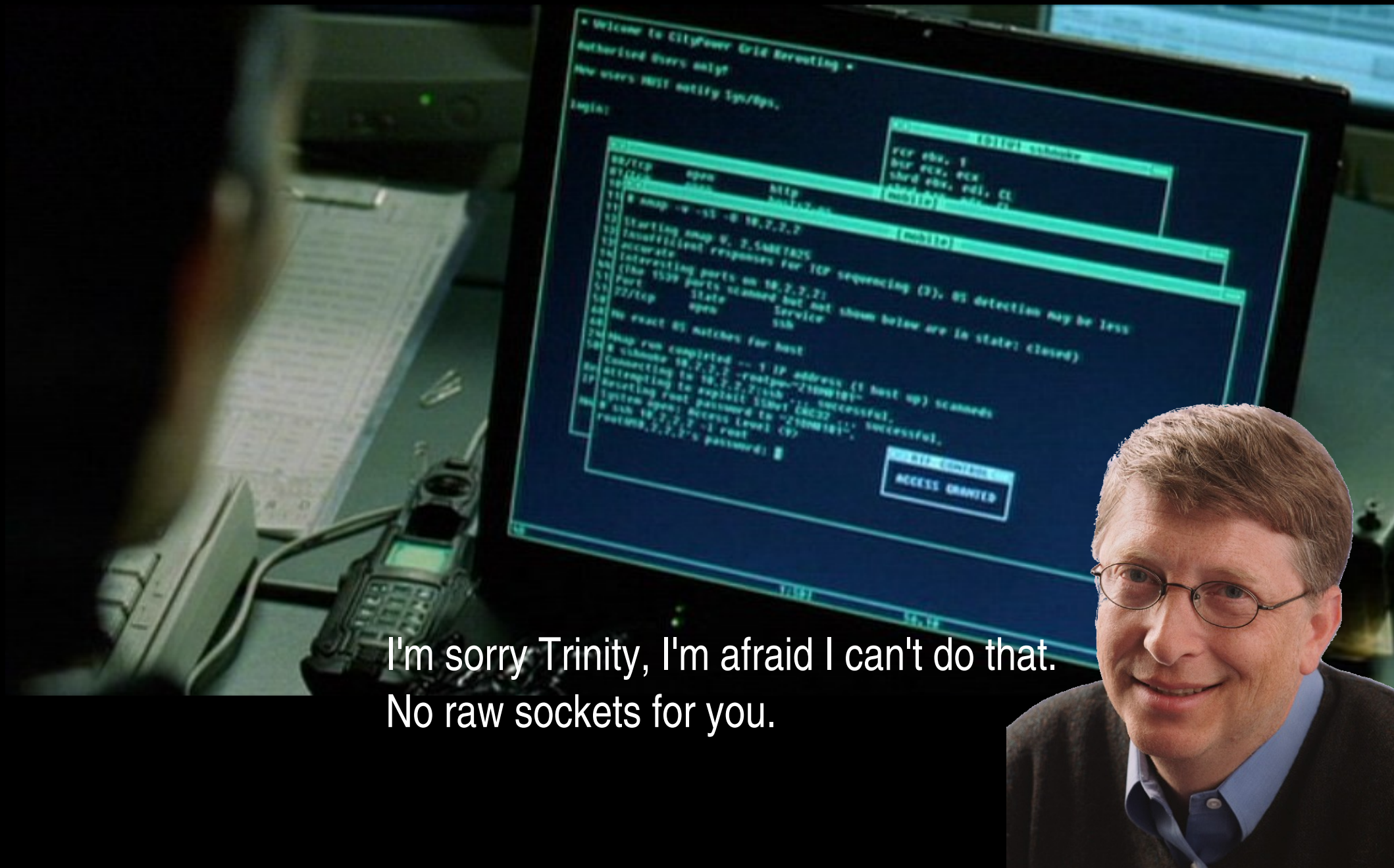
$> Hacken mit Linux

Benedikt 'Hunz' Heinz
<hunz @ hunz.org>

Weitere Erläuterungen zum Vortrag finden sich hier:
http://hunz.geekheim.de/?p=22

Code der im Vortrag verwendet wird findet sich unter
http://hunz.org/lit06/


I'm sorry Trinity, I'm afraid I can't do that.
No raw sockets for you.

Was haben Kant und Linus gemeinsam?

- Kant als wichtigster Denker der deutschen Aufklärung
- Was ist Aufklärung?
  - Kant: *„Aufklärung ist der Ausgang des Menschen aus seiner selbst verschuldeten Unmündigkeit.“*
- Was hat das mit Linux zu tun?
  - Kein „Blackbox“ Betriebssystem das dem Benutzer vorschreibt was er kann / darf und was nicht
  - Freies Denken <-> freier Quellcode
  - Nicht auf fertige, kommerziell verfügbare Programme warten
  - sondern selbst Hand an den Code legen, versuchen diesen zu verstehen und verändern
- Bedeutung von Linux im Informationszeitalter

Was kann man mit raw sockets machen?

- Beliebige Netzwerkpakete (Ethernet, Wireless LAN, etc.) selbst „zusammenbauen“ und versenden
- Beliebige Absenderadressen, nicht standardkonforme Pakete, eigene Protokolle im Userspace
  - Man in the middle Attacken
  - Denial of service
  - Beispiel: Wireless LAN
    - „gefälschte“ Abmeldung verschicken: beliebige Leute in beliebigen Netzen abmelden: 100 – 150 Zeilen C
    - kann man auch für Trafficshaping benutzen: Teilnehmer die zu viel übertragen automatisch vom Netz abmelden
    - Verschlüsselung aktiv angreifbar

Inhalt dieses Vortrags

- Dateisysteme als Benutzerschnittstelle
  - Tastaturbeleuchtung -> Discolight
  - was die Kripo kann können wir auch: gelöschte Dateien wiederfinden
  - bei Flash Spielen schummeln
  - warum man Videos unter Linux nicht kopierschützen kann
- Interessante Software zum Spielen
  - Spezialeffekte in Videos mit MPlayer
  - Netzwerkkram
    - kommerzielle Wireless LAN Hotspots kostenlos benutzen
    - mein eigener WWW-Bot in 5 Minuten

Wäre die Festplatte eine normale Textdatei...

- könnte man sie mit einer Textverarbeitung einfach öffnen, bearbeiten und die Änderungen abspeichern
- Die Kommandozeile als Textverarbeitung
  - echo schreibt in Dateien
  - cat zeigt Dateien an
  - pipes leiten Dateiinhalte um / weiter (z.B. filtern durch grep)
  - netcat überträgt Dateiinhalte über das Netzwerk
- Programmierersicht:
  - open / close
  - read / write
  - mmap: Änderungen müssen nicht extra gespeichert werden

Systeminformationen als Textdateien

- /proc
  - cpuinfo: Der Prozessor stellt sich vor
  - meminfo: Was passiert mit dem Hauptspeicher?
  - partitions: Ein Blick in die Partitionstabelle(n)
- /proc/acpi/
  - battery/: Wie gehts dem Akku?
  - thermal_zone/: Prozessortemperatur
- /proc/acpi/ibm/light: Tastaturbeleuchtung
  - Discolight in ca. 100 Zeilen C

Welche Hardware steckt im Rechner?

- /proc/bus
  - lspci
  - usbview
- Warum sind doch ein paar Extratools nötig?
  - Systemkern (Kernel) kleiner halten
    - Embedded Systeme
    - KISS: Keep it simple, stupid - mehr Code -> mehr potentielle Sicherheitslücken
  - Extratools sind einfacher/schneller zu aktualisieren

FUSE: Das Dateisystem im Userspace

- SSHfs: Ermöglicht das Einbinden der Dateisysteme fremder Rechner ohne Eingriff am Server
- GMailfs: GoogleMail Speicherplatz (2GB?) als Online-Festplatte benutzen
- Bluetooth / SiemensFS: Bluetooth Geräte ohne Extratreiber einbinden
- Datenbank-Filesysteme: Erweiterte Suchmöglichkeiten
- WikipediaFS, FlickrFS
- diverse CryptoFSe, CVS/WaybackFS
- Archive einbinden: tar, lzo, pack, wad
- Bittorrent, WebDAV, FTP, BlogFS

/sys und /dev

- /sys/ - Die Tür zum Maschinenraum
  - cpufreq: Welche Taktraten kann die CPU? Taktraten im laufenden Betrieb ändern
- /dev/ - nützliche Gerätedateien
  - null: Das schwarze Loch im Dateisystem um Daten „wegzuwerfen“
  - (u)random: Diese Dateien enthalten immer frische Zufallszahlen
    - Cryptographie
    - für „sicheres“ Löschen
  - zero: Enthält Nullen in unbegrenzter Menge
    - zum Überschreiben / Löschen
    - Benchmarking

Die Festplatte ist doch eine Datei!

Der USB-Stick auch! :-)

- Block-Geräte: Zugriff erfolgt beschleunigt in Blöcken
- sortiert nach Gerät und Partition
- Wahlweise ganzes Gerät oder einzelne Partition
- IDE Geräte (Festplatten, DVD/CD): z.B. /dev/hda1
- SCSI und USB Geräte: z.B. /dev/sda1
- Benutzung als „normale“ Datei möglich:
  - einfaches Durchsuchen der binären Informationen
  - Ermöglicht das Auffinden von gelöschten Daten sofern diese noch nicht überschrieben wurden: ca. 50 Zeilen C
  - Ermöglicht das Anlegen von 1:1 Kopien: dd
  - Erleichtert die Datenrettung: ddrescue

Virtuelle Festplatten

- Entfernte Platten Dateisystemunabhängig via Netzwerk nutzbar: Network Block Device
- Das loopback Interface
  - Nutzung einzelner Dateien als Festplatten / CDs / DVDs
  - transparente Ver-/Entschlüsselung für beliebige Dateisysteme und Auslagerungsspeicher: ca. 4 Zeilen Konfiguration
- device mapper

Keylogger mit einem Befehl: /dev/input

- Dateien für Eingabegeräte: Maus, Tastatur, Joystick
- Schreib- und lesbar: Eingaben können protokolliert und simuliert werden

  -> ermöglicht virtuelle Eingabegeräte
- Eingaben können einfach über Netzwerk übertragen werden
- Flash spiele: Bestimmte Eingabefolge in kurzer Zeit
  - Eingabefolge mitprotokollieren
  - Protokoll schneller wieder abspielen
  - unter 50 Zeilen C

Zugriff auf den Videospeicher: /dev/fb0

- Kann gelesen, geschrieben und gemappt werden
- Matrox Unterstützung am besten
- „Größe“ eines Pixels im Speicher abhängig von der Farbtiefe
- Aufbau: Ein Pixel nach dem Andern
- Beispiel mit Auflösung 4x3 Pixel:

  Im Speicher: Pixel1 Pixel2 Pixel3 Pixel4 Pixel5 Pixel6 Pixel7 ...

  Im Koordinatensystem: Pixel 1 bis 4: Zeile 1, 5 bis 8: Zeile 2, ...
- Jedes Pixel kann direkt geändert werden

Videospeicher(2): Videoeffekte

- Diese Anordnung der Pixel im Speicher ist Standard
- in Grafikbibliotheken
- im Videoplayer MPlayer
  - etwa 25 Bilder pro Sekunde
  - in jedem dieser Bilder kann jedes Pixel gelesen und geschrieben werden
  - so lässt sich das Video in Echtzeit beim Abspielen verändern
  - z.B. alle Bewegungen rot einfärben: 200 Zeilen C

Spaß im Netz

- HTTP Tunnel: Daten in HTTP verstecken
  - ermöglicht die Benutzung einer HTTP Verbindung zum Übertragen von non-HTTP Daten (z.B. Mail)
  - z.B. in Netzen mit Firewall
- OpenVPN ermöglicht
  - verschlüsseltes Surfen im Netz durch:
  - „Einwahl“ in ein Netz mit OpenVPN Server
  - Verbinden von mehreren Netzen
- HTTP Tunnel in Kombination mit OpenVPN ermöglicht vollen Internetzugang in Netzen in denen sonst nur HTTP freigeschaltet ist
- Web-Zugriffe automatisieren mit Perl

Was soll ich tun?

- Die Erläuterungen lesen ;-)
- C lernen
- Eine Skriptsprache (Perl oder Python) lernen
- Manuals lesen
- Sourcecode lesen
- Am System rumspielen
- Erkenntnisse und Code mit andern teilen
- Neue Hacker rekrutieren :-)