Unix aus Datenschutzsicht

1. Rechtestruktur
   - Benutzerrollen
   - Dateisystem
2. Hauptprobleme und Abhilfevorschläge
   - Unkontrollierbarkeit des SuperUsers
   - fehlende Protokollierungsmöglichkeiten
   - Gefährdung durch mächtige Programme
3. lokale Sicherheit (-slücken)
   - beim Server
   - beim Client
4. Network File System (NFS)
   - Transmission Control Protocol / Internet Protocol (TCP/IP)
   - Remote Procedure Call (RPC)
   - Networt Information System (NIS)

Vorbemerkungen

- Ergänzungen & Beispiele sind willkommen.

- Unix ⇔ NetWare ⇔ WinNT:

  Unix ist kein Produkt,
  sondern eine Abstraktion über viele Produkte
  (Varianten & Versionen).

- diese Folien im W³:

  http://swt-www.informatik.uni-hamburg.de/~1gehrke

  als gzipptes Postscript

Rechtestruktur

- Benutzerrollen

  nur 2 Klassen von Benutzern vorgesehen:

  - User
  - SuperUser

  ⇒ keine SubAdministratoren

  ⇒ keine Unterscheidung zwischen Anwender und Entwickler

- Dateisystem

  Rechte pro Datei:

  read, write & execute

  jeweils für

  owner, group & world

  ⇒ Datei gehört nur zu einer Gruppe, zu dieser aber komplett.

  - User kann zu mehreren Gruppen gehören,
    davon ist aber immer nur eine aktiv.

Hauptprobleme (1)

Unkontrollierbarkeit des SuperUsers:

☹ SuperUser unterliegt auf Betriebssystemebene keinerlei Beschränkungen

⇒ voller Zugriff auf alle Dateien

⇒ keine aussagefähigen Protokolldateien möglich

☺ Abhilfe (-versuche):

- Verschlüsselung von Dateien
- Systemverwaltung über Menüsystem

  ⇐ Betriebssystem-Defizite auf Anwendungsebene kompensieren
- Betriebsystemzugang als SuperUser mit geteiltem Paßwort

  ⇒ 4-Augen-Prinzip

Hauptprobleme (2)

fehlende Protokollierungsmöglichkeiten:

❍ standardmäßig gar keine Protokollierung

⇒ erfolgter Mißbrauch im Nachhinein nicht erkennbar

☺ Abhilfe (-versuch):

- User-Befehle auf protokollierende Kommandos umleiten

  😐 Problem: SuperUser bleibt unkontrollierbar

  😐 Problem: Protokolle sind sensible Daten.

  ⇒ Protokolle asymetrisch verschlüsselt erstellen

Hauptprobleme (3)

Gefährdung durch mächtige Programme:

↳ SUID-Programm = Set-User-ID-Programm
Das Programm wird nicht mit den Rechten der aufrufenden Person, sondern mit denen des Besitzers / Erzeugers der Datei ausgeführt. Von besonderer Bedeutung sind vom SuperUser erzeugte SUID-Programme. Einige derartige Programme sind unverzichtbarer Bestandteil des Betriebssystems Unix.

❍ SUID-Programme müssen selber gewährleisten, daß ihr Anwender ihre erweiterten Rechte nicht ausgiebiger nutzt als (vom Administrator) beabsichtigt.

⇒ fehlerhafte / überlistbare Programme (schlimmstenfalls) als Möglichkeit zur Erschleichung des SuperUser-Status

☺ Abhilfe (-versuch):

- Watchdog-Programme suchen das Dateisystem regelmäßig nach SUID-Programmen ab

⇒ Administrator blockiert unbekannte SUID-Programme

lokale Sicherheit (-slücken) (1)

beim Server:

- ☹ nach dem Booten im Ein-Benutzer-Betrieb
- ☹ Ein-Benutzer-Betrieb stets im SuperUser-Modus

☺ Abhilfe (-versuche):

- "shutdown" nur für SuperUser ermöglichen
- nach dem Booten automatisch, ohne Eingriffsmöglichkeit in Mehr-Benutzer-Betrieb wechseln
  oder:
  Paßwort für Verbleib im Ein-Benutzer-Betrieb

lokale Sicherheit (-slücken) (2)

beim Client:

- In der Datei ".rhosts" im Login-Verzeichnis jedes Users kann jeder User Kombinationen aus User und Rechner festlegen, die beim Zugriff auf das lokale System keine Authentifizierung angeben müssen.

⇒ vom Client aus Sicherheit des Servers ausgehebelt

- betroffene Kommandos:
  rlogin, rsh, rcp

☺ Abhilfe (-versuche):

- Trennung von Login- und Home-Verzeichnis:
  Login-Verzeichnis in "/etc/passwd" festgelegt
  Home-Verzeichnis durch $HOME-Variable bestimmt
- Usern das Schreibrecht auf Login-Verzeichnis nehmen, aber für das Home-Verzeichnis einräumen

- weitere Sicherheitslücken bei rlogin, rsh, rcp:

  - unverschlüsselte Übertragung von Benutzerkennung und Paßwort

  ⇒ Abhörgefahr

Network File System (NFS) (1)

Begriffsklärung: Was tut NSF ?

- Auf dem lokalen System werden Dateien zur Verfügung gestellt, als wären sie dort gespeichert. Sie sind aber nicht lokal gespeichert und werden auch nicht im herkömmlichen Sinne übertragen.
- auch Ausführung von Kommandos auf anderen Rechnern möglich - mit dem Befehl "on".

😐 Problem:
keine Verschlüsselung von Nutzdaten während des Transportes möglich

- Transmission Control Protocol / Internet Protocol (TCP/IP)
  - liegt (u.a.) NFS zugrunde:
    IP: OSI-Schicht 3
    TCP: OSI-Schicht 4
    NFS: OSI-Schicht 7

    (liegt auch rlogin, rsh, rcp zugrunde)

  - Authentifizierung von Netzwerkadressen nicht möglich

  ⇒ User können in die Identität eines anderen Systems schlüpfen.

  ☺ Abhilfe (-versuch):
  Firewall-System (⇨ 2. heutiger Vortrag)

Network File System (NFS) (2)

❍ Remote Procedure Call (RPC)

- wird von NFS zur Identifizierung und Authentifizierung benutzt
- erst ab NFS 4.0 verschlüsselte, authentifzierte, Replay-geschützte Übertragung möglich

⇒ Sonst: Abhörgefahr

❍ Networt Information System (NIS)

- zur netzweiten Abstimmung von Paßwort- und Gruppendateien

⇒ erleichtert den Einsatz von NSF

😐 Problem:
ermöglicht überall im Netz

- ⊙ Einsatz von Paßwort-Rateprogrammen
- ⊙ Beobachtung von Benutzeraktivitäten

unterstützt

- ⊙ Suche nach Angriffspunkten im Netz
- ⊙ gezieltes Fälschen von Netzadressen