GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

CIDAS Campus-Institut Data Science

Einführung in Sicherheit und Privatheit

Anwendungsbeispiel: Unix Sicherheit

Prof. Dr.-Ing. Delphine Reinhardt

Universität Göttingen
Institut für Informatik und Campus-Institut Data Science
Computersicherheit und Privatheit (CSP)
Goldschmidtstr. 7
37077 Göttingen, Germany
Email: reinhardt@cs.uni-goettingen.de
www.csp.informatik.uni-goettingen.de

Inhaltsverzeichnis

UNIX Grundlagen

Grundlagen

- **Principals**
  - Nutzeridentitäten (`uids`) und Gruppenidentitäten (`gids`)
  - 16 bits (bis zu 32 bits möglich)
  - Können verschiedene Bedeutung für verschiedene Systeme haben
  - Gebraucht für lokale Zugriffskontrolle
- **User accounts**
  - Informationen über Principals sind in user accounts und home directories gespeichert
  - User accounts in `/etc/passwd` gespeichert

Quelle: D. Gollmann, Computer Security, 3rd Edition, Wiley, 2011

Grundlagen

- **Superuser/Root**
  - Immer UID=0
  - Benutzt vom Betriebsystem fürs Logins, Audit Logs, sowie Zugriffe auf Geräte und Eingang/Ausgang
  - Sicherheitsschutz ist ausgeschaltet
  - Kann die Rolle eines anderen Nutzers annehmen
  - ABER: kann die Passwörter der Nutzer dank der Verschlüsselung nicht auslesen
- **Gruppen**
  - Ein Nutzer kann in einer oder mehreren Gruppen sein
  - GID der primären Gruppe des Nutzers auch in `/etc/passwd` gespeichert
  - Vereinfachen die Zugriffskontrolle

Quelle: D. Gollmann, Computer Security, 3rd Edition, Wiley, 2011

Grundlagen

- **Subjekte**: Prozesse
  - Jeder Prozess ist mit einer Prozess ID (PID), einem reellen und einem wirksamen UID/GID assoziiert
  - Erstellung durch `exec` or `fork`

Quelle: D. Gollmann, Computer Security, 3rd Edition, Wiley, 2011

Grundlagen

- Systemdienste können von Prozessen über Systemaufrufe zum Zugriff auf Dateisysteme genutzt werden
  - Systemaufrufe sind z.B. `open`, `read`, `write`, `fork`, `exec`, `kill`
  - Jeder Systemaufruf führt zu einem Moduswechsel
  - Systemdienste werden dann mit privilegierten Berechtigungen ausgeführt

Für uns besonders interessant:

- Benutzerauthentifikation: Login-Programm `/bin/login` und Zugangskontrolle
- Dateisystem mit Zugriffskontrolle

Quelle: C. Eckert, IT-Sicherheit: Konzepte-Verfahren-Protokolle

Benutzerauthentifkation in UNIX

Logins und Passwörter

Grundlagen

- System startet im Root Modus
- Ist der Login erfolgreich, werden die UID und GID geändert und das Login Shell durchgeführt
- `/usr/adm/lastlog`: Liste der letzten Anmeldungen

Quelle: D. Gollmann, Computer Security, 3rd Edition, Wiley, 2011

Passworteinträge

- Passworteinträge in `/etc/passwd` gespeichert

Frage

Warum ist es ein Problem?

- `passwd` ist von jedermann lesbar
- Angreifer kann es kopieren und offline Wörterbuchattacke starten

Frage

Welche Lösung könnte adaptiert werden?

- Passworteinträge ausgelagert in nicht öffentliche Datei `/etc/shadow`
  - enthält verschlüsselte Passwörter (d.h. nicht mehr in `/etc/passwd`)
  - nur die Kennung `root` besitzt Lesezugriff

Quelle: M. Meier, IT-Sicherheit Vorlesung

Erstellung der Passworteinträge

- In `/etc/shadow` wird der Hash des Passworts gespeichert
- Früher mit **DES** verschlüsselt (dadurch war Entschlüsselung möglich)
- Heutzutage wird nur der Hash (mit Salt) gespeichert (zB SHA-512)

Zugriffskontrolle in UNIX

Zugriffskontrolle

Subjekte

- Benutzer
- Benutzergruppen
- Prozesse

Objekte

- Dateien
  - (externe) Geräte wie Monitor, Drucker, Modem sowie Massenspeicher wie Festplatten, CD-Rom-Laufwerke oder auch Arbeitsspeicher des Betriebssystemkerns (`/dev/kmem`) werden als Dateien modelliert
- Verzeichnisse

Quelle: M. Meier, IT-Sicherheit Vorlesung

Identifikation

- von Dateien
  - Baumartig strukturiertes Dateisystem
  - Benennung einer Datei oder eines Verzeichnisses durch Pfadnamen
- von Prozessen
  - pid - vom Betriebssystemkern vergebene eindeutige Prozessidentifikatoren
- von Benutzern und Gruppen
  - uid und guid – eindeutige User/Group Identifikatoren
  - $\mathrm{uid} = 0$ vorgesehen für **Superuser**
  - unterschieden wird zwischen **realen** und **effektiven** uid und guid
    - Zugriffsberechtigungen hängen von der effektiven uid und guid ab
    - reale uid und guid fest
    - effektive uid und guid kann sich dynamisch im Verlauf einer Sitzung ändern (vgl. setuid-Konzept)

Quelle: M. Meier, IT-Sicherheit Vorlesung

Identifikation

- "Namen sind irrelevant"
- uids werden in $/\mathrm{etc}/\mathrm{passwd}$ Benutzernamen zugeordnet
- guids werden in $/\mathrm{etc}/\mathrm{group}$ Gruppennamen zugeordnet
- Für Rechteüberprüfung ist **nicht** der Benutzername und **nicht** der Gruppenname relevant sondern nur die uid und die guid
- Benutzer mit $\mathrm{uid} = 123$ und Benutzername root ist keineswegs Superuser
- Superuser: $\mathrm{uid} = 0$

Quelle: M. Meier, IT-Sicherheit Vorlesung

Rechtevergabe

- (im Wesentlichen) verfügbare Zugriffsrechte
  - r: lesen
  - w: schreiben
  - x: ausführen

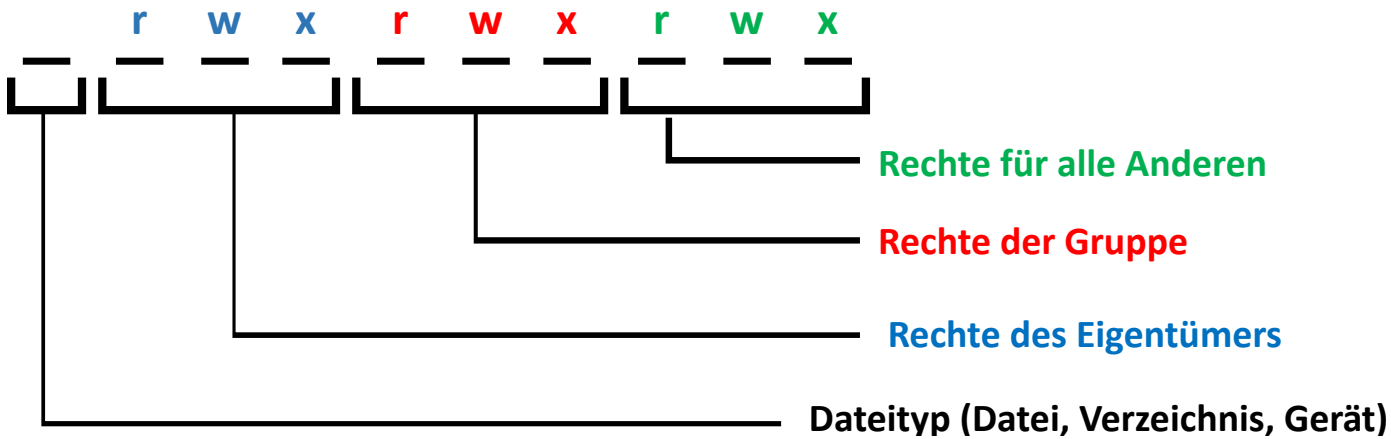
- nur der Eigentümer des Objekts *o* und Superuser können die Rechte an Objekt *o* ändern (also hinzufügen oder entfernen)

- Jedes Objekt besitzt eine Zugriffskontrollliste mit Rechten für drei Klassen von autorisierten Benutzern:
  - Objekteigentümer (Owner-Rechte)
  - Benutzer, die der Gruppe angehören, der das Objekt gehört (Gruppen-Rechte)
  - alle anderen Benutzer (World-Rechte)

Quelle: M. Meier, IT-Sicherheit Vorlesung

UNIX-Zugriffskontrollliste

- Allgemeine Struktur



- An Dateitypen werden unterschieden
  - – einfache Datei (regular file)
  - d Verzeichnis (directory)
  - l Verweis (link)
  - c zeichenorientiertes Gerät (z.B. Terminal, Drucker)
  - b blockorientiertes Gerät (z.B. Band)

Quelle: M. Meier, IT-Sicherheit Vorlesung

Zugriffsrechte Verändern mit CHMOD

- Rechte für Dateieigentümer (**U**ser) ändern
  chmod **u**+r datei
  chmod **u**-r datei

- Rechte für die **G**ruppe ändern, zu der Datei/Verzeichnis gehört
  chmod **g**+w datei

- Rechte für alle Anderen (**O**ther) ändern
  chmod **o**+x datei

Quelle: M. Meier, IT-Sicherheit Vorlesung

Rechte-Semantik für Verzeichnisse

- r – **Leserecht**: erlaubt, den Inhalt des Verzeichnisses aufzulisten (z.B. mit ls)
- w – **Schreib-Recht**: erlaubt das Hinzufügen oder Entfernen von Elementen zu bzw. aus dem Verzeichnis
- x – **Suchrecht**: erlaubt, das Verzeichnis als Teil eines Pfadnamens zu durchlaufen bzw. das Verzeichnis mit dem Kommando cd "zu betreten" und darin befindliche Dateien zu öffnen

- Beispiel: drwx r- - - - - -
  - Objekt ist ein ...
  - Eigentümer besitzt .....
  - Mitglieder der Gruppe besitzen ...
  - Alle anderen Benutzer besitzen ...

Quelle: M. Meier, IT-Sicherheit Vorlesung

Rechte-Semantik für Verzeichnisse

- r – **Leserecht**: erlaubt, den Inhalt des Verzeichnisses aufzulisten (z.B. mit ls)
- w – **Schreib-Recht**: erlaubt das Hinzufügen oder Entfernen von Elementen zu bzw. aus dem Verzeichnis
- x – **Suchrecht**: erlaubt, das Verzeichnis als Teil eines Pfadnamen zu durchlaufen bzw. das Verzeichnis mit dem Kommando cd "zu betreten" und darin befindliche Dateien zu öffnen

- Beispiel: drwx r- - - - - -
  - Objekt ist Verzeichnis
  - Eigentümer besitzt Lese-, Schreib- und Ausführungs-/Such-Recht
  - Mitglieder der Gruppe, zu der das Objekt gehört, haben das Recht das Verzeichnis zu lesen, also den Inhalt aufzulisten
  - Allen anderen Benutzern sind alle Zugriffe verwehrt.

Quelle: M. Meier, IT-Sicherheit Vorlesung

Such-Recht für Verzeichnisse

- Um auf eine Datei zugreifen zu können muss der zugreifende Nutzer für alle Verzeichnisse des Pfadnamens der Datei das Such-Recht besitzen
  - /home/reinhardt/teaching/EiCSP.pdf
  - Um auf EiCSP.pdf zugreifen zu können, müssen für Verzeichnisse home, reinhardt, teaching x-Rechte vorliegen

- Suchrecht ist **nicht** gleichzeitig mit Leserecht verknüpft, sodass allein mit x-Recht ein Auflisten des Verzeichnisinhalts **nicht** zulässig ist.
- Idee: Nur wer Dateinamen im Verzeichnis kennt, kann auf Datei zugreifen.

$\rightarrow$ Verstecken von Dateien, allerdings nur schwacher Schutz

---

Quelle: M. Meier, IT-Sicherheit Vorlesung

Schreib-Recht für Verzeichnis

- Mit `w-` und `x-`Recht für ein Verzeichnis ist das Recht zum Entfernen von Dateien aus dem Verzeichnis erteilt,
  unabhängig davon, wie die Rechte der Dateien gesetzt sind und
  unabhängig davon welche Benutzer Eigentümer der Dateien sind.

Quelle: M. Meier, IT-Sicherheit Vorlesung

Beispiel

- **Datei** f mit rw- r- - - - -
- Datei f befindet sich in **Verzeichnis** d, das zur gleichen Gruppe wie Datei f gehört, mit rwx -wx - - -

Kann ein Mitglied der Gruppe, zu der die Datei gehört, die Datei verändern?

- Alle Benutzer der Gruppe, zu der das Verzeichnis gehört, besitzen **Schreibrecht für Verzeichnis** d
- Schreibrecht für $d \rightarrow$ Löschrecht für Datei f
- Alle Benutzer der Gruppe besitzen kein **Schreibrecht für Datei** f
- $\Rightarrow$ Benutzer der Gruppe können Datei f entfernen und durch neue Datei ersetzen!
- Diese Zugriffe sind durch Rechte für Verzeichnis d erlaubt, widersprechen jedoch der intendierten Rechtevergabe für Datei f

Detailkenntnisse erforderlich um Inkonsistenz zu vermeiden!

Sonderrechte

Sticky-Bit

- Ursprünglich: Hinweis, dass Datei von der Speicherverwaltung nicht auf Festplatte ausgelagert werden darf.
- Heute: nur noch für **Verzeichnisse** relevant.
- Normaler Fall:
  Schreibberechtigung für Verzeichnis $\rightarrow$ Löschrecht für alle Datei (auch wenn Benutzer!= Dateieigentümer)
- Mit Sticky-Bit:
  Löschrecht $\rightarrow$ nur Eigentümer der Datei und Eigentümer des Verzeichnisses (und Superuser)
- Sinnvoll für Verzeichnisse wenn alle Benutzer Schreib- und Ausführungsrechte besitzen (z.B. /tmp)

$$\Downarrow$$

drwx rwx rw**t** 6 sys sys 577 Dec 2 11:15 tmp

Sonderrechte

SETUID/SUID

- SUID $\rightarrow$ temporäre Weitergabe von Rechten eines Benutzers
- Mit SUID-bit:

  Effektive User-ID $\rightarrow$ User-ID des Eigentümers der Datei
  - Der Prozess führt das Programm mit den Rechten des Eigentümers aus.
- Eigentümer oder Superuser setzen SUID-bit mit

  $$\text{chmod u+s datei}$$
- Voraussetzung: Ausführungsrechte für Gruppe oder Andere
- Nur für Dateien

  $$\Downarrow$$

  $$\text{- r-}\mathbf{s}\text{ r-}\mathbf{s}\text{ r-x 1 root sys ... /bin/passwd}$$

Quelle: M. Meier, IT-Sicherheit Vorlesung

Sonderrecht

SETGID/SGID

- Analog SUID
- Temporäre Weitergabe von Rechten **einer Gruppe**
- Mit SGID-bit:
  Effektive Group-ID $\rightarrow$ Group-ID der Eigentümergruppe der Datei
  - Der Prozess führt das Programm mit den Rechten der Eigentümergruppe aus.

Quelle: M. Meier, IT-Sicherheit Vorlesung

Risiken von SUID

- Beispiel 1: - rws r— rwx klaus stud ... beispielprogramm
  - Schreibbar und ausführbar für alle

  → Jeder kann Programm ändern!
  - Läuft aufgrund SUID mit den Rechten von Klaus

  → Jeder kann beliebige Aktionen mit den Rechten von Klaus ausführen!
- Beispiel 2: Der Superuser lässt seinen Rechner einen Moment unbeaufsichtigt:
  - Angreifer tut Folgendes:
    cp /bin/sh /tmp/endlich_root
    chmod o+x /tmp/endlich_root
    chmod +s /tmp/endlich_root

  → Kommandozeilen-Interpreter/Shell die mit Root-Rechten ausgeführt wird!
  - Angriff mittlerweile durch Check des Betriebssystems nicht mehr möglich

Quelle: M. Meier, IT-Sicherheit Vorlesung

Sonderrecht

SETGID/SGID für Verzeichnisse

- Benutzer erstellt eine neue Datei in einem Verzeichnis
  $\rightarrow$ Benutzer = Eigentümer der Datei
  $\rightarrow$ Datei gehört zur (primären) Gruppe des Benutzers

- Benutzer erstellt eine neue Datei in einem Verzeichnis für das SGID gesetzt ist
  $\rightarrow$ Datei gehört zu der Gruppe, zu der das Verzeichnis gehört.

- Wo benötigt man das?
  - Wenn Nutzer mehreren Gruppen zugehören und Dateien untereinander teilen

Quelle: M. Meier, IT-Sicherheit Vorlesung

Fazit

Fragenrunde

Frage

Gibt es noch Fragen?

Kontakt

Vincent Muller/Sébastien Haller

Prof. Dr.-Ing. Delphine Reinhardt
Computersicherheit und Privatheit
Universität Göttingen, Institut für Informatik
Goldschmidtstr. 7
37077 Göttingen, Germany
Email: reinhardt@cs.uni-goettingen.de
www.csp.informatik.uni-goettingen.de

Copyright Notice