Script **generated by TTT**

Title: Grundlagen_Betriebssysteme (03.02.2012)

Date: Fri Feb 03 08:30:43 CET 2012

Duration: 92:18 min

Pages: 32

Beispiel: Pufferüberlauf

Durch einen künstlich herbeigeführten Pufferüberlauf kann ein Angreifer die Ausführung seines eigenen Programms veranlassen und oft auch noch die Systemadministrator-Berechtigung (root) erlangen.

Hintergrund

Veränderung der Rücksprungadresse

Gegenmaßnahmen

Unterscheidung

sichere Programmierung.

Maßnahmen zur Übersetzungszeit.

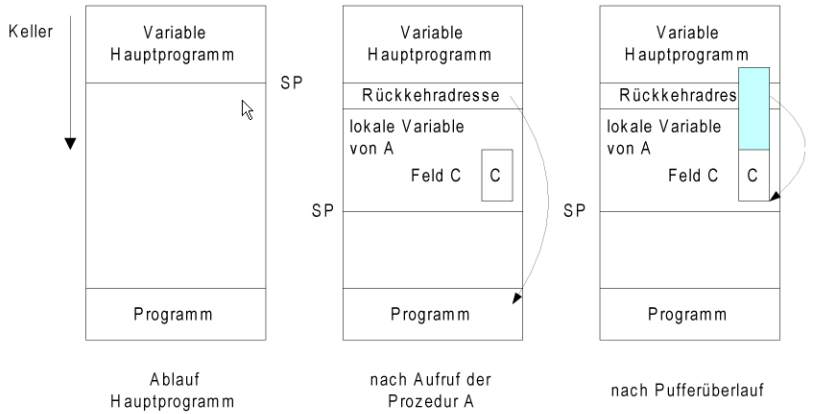
Maßnahmen zur Laufzeit.

Die meisten C-Compiler und Laufzeitsysteme überprüfen nicht die Einhaltung der Feldgrenzen.

```
int i;
char c[256];
i = 12000;
c[i] = 0;
```

String Library in C

Veränderung der Rücksprungadresse



Falls das attackierte Programm mit root-Berechtigung (setuid root in Unix) abläuft, läuft das aufgerufene Programm im Puffer auch mit root-Berechtigung.

Angreifer kann seiner aufrufenden Shell root-Berechtigung verleihen.

```
void echo() {
    char buf[4]; /* sehr klein */
    gets(buf);
    puts(buf);
}
int main() {
    printf("Type a string:");
    echo();
    return 0;
}
```

unix> bufdemo

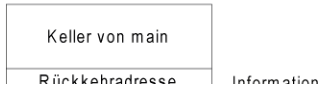
Type a string: ***123***

123

unix> bufdemo

Type a string: ***12345***

Segmentation Fault



Implementierung der Unix Funktion `gets` (get string from stdin)

keine Möglichkeit zur Spezifikation der Anzahl der zu lesenden Zeichen

```
char *gets(char *dest) {
    int c = getc();
    char *p = dest;
    while (c != EOF && c != '\n') {
        *p++ = c; c = getc();
    }
    *p = '\0';
    return dest;
}
```

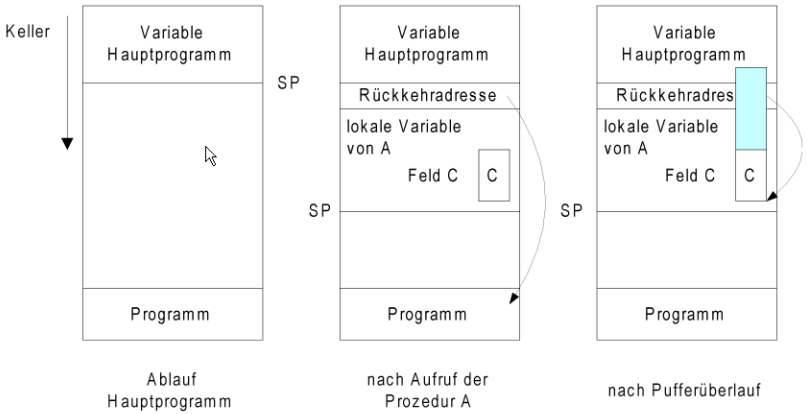
ähnliche Probleme auch bei anderen Unix Funktionen

strcpy: kopiert einen String beliebiger Länge

scanf, fscanf, sscanf: mit %s Konvertierungsspezifikation.

Angreifbarer Buffer Code

Veränderung der Rücksprungadresse



Falls das attackierte Programm mit root-Berechtigung (setuid root in Unix) abläuft, läuft das aufgerufene Programm im Puffer auch mit root-Berechtigung.

Angreifer kann seiner aufrufenden Shell root-Berechtigung verleihen.

Beispiel: Pufferüberlauf

Durch einen künstlich herbeigeführten Pufferüberlauf kann ein Angreifer die Ausführung seines eigenen Programms veranlassen und oft auch noch die Systemadministrator-Berechtigung (root) erlangen.

Hintergrund

Veränderung der Rücksprungadresse

Gegenmaßnahmen

Unterscheidung

sichere Programmierung.

Maßnahmen zur Übersetzungszeit.

Maßnahmen zur Laufzeit.

Motivation

Was versteht man unter Sicherheit im Bezug auf Rechensysteme?

Jemand : Unterscheidung von Personen und Gruppen von Personen

davon abhalten : durch technische und organisatorische Maßnahmen

einige : Begrenzung durch unser Vorstellungsvermögen

unerwünschte Dinge zu tun :

1) nicht autorisiert Daten lesen (Geheimhaltung, Vertraulichkeit),

2) nicht autorisiert Daten schreiben (Integrität),

3) unter "falscher Flagge" arbeiten (Authentizität),

4) nicht autorisiert Ressourcen verbrauchen (Verfügbarkeit),

usw.

zu tun .

Unterscheidung zwischen Angriffen von

innen .

außen .

Beispiel: Login-Attrappe

Beispiel: Virus

Beispiel: Pufferüberlauf

Sicherheit in Rechensystemen

Fragestellungen

Dieser Abschnitt behandelt die Sicherheitsproblematik in zentralen Rechensystemen. Dazu werden verschiedene Schutzmechanismen auf Betriebssystemebene vorgestellt.

Zugriffsschutz in Rechensystemen.

Schutzmatrix, insbesondere Zugriffskontrolllisten und Capability-Listen.

Mobiler Code.

Schutzmechanismen

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

Anforderungen

Für einen Schutzmechanismus gelten die folgenden Anforderungen

alle Objekte eines Systems müssen eindeutig und fälschungssicher identifiziert werden.

externer Benutzer eines Systems muss eindeutig und fälschungssicher identifiziert werden ⇒ Authentifizierung.

Zugriff auf Objekte sollte nur über zugehörige Objektverwaltung geschehen.

Zugriff auf Objekte nur, wenn Zugreifer die nötige Rechte hat.

Rechte müssen fälschungssicher gespeichert werden; Weitergabe von Rechten darf nur kontrolliert erfolgen.

Prinzip der minimalen Rechte.

grundlegenden Schutzmechanismen sollen ohne großen Aufwand überprüft werden können.

Schutzmechanismen

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

Schutzmechanismen

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

Anforderungen

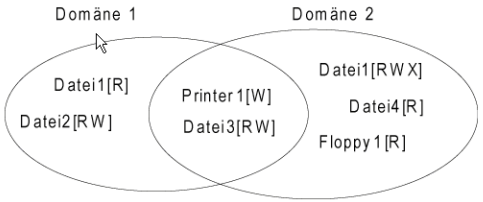
Ebenen des Zugriffschutzes

Schutzmatrix

Authentifizierung

Generated by Targeteam

Definition: Eine **Schutzdomäne** ist eine Menge von (Objekt, Rechte) Paaren.



R = read, W = write, X = execute

Verknüpfung eines Prozesses mit einer Schutzdomäne.

zu jedem Zeitpunkt wird ein Prozess in einer Schutzdomäne ausgeführt.

Beispiel Unix: bei Ausführung eines Systemaufrufs wechselt der Prozess vom Benutzermodus in den Systemmodus ("kernel mode") ⇒ entspricht einem Wechsel der Schutzdomäne.

Das Paar (Prozess P, Schutzdomäne D) wird als ***Subjekt*** bezeichnet.

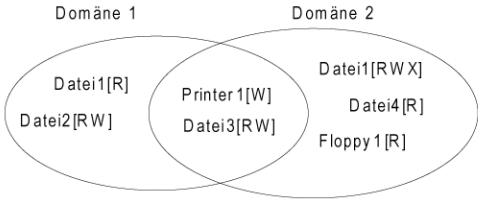
Der Zugriffswunsch eines Subjektes S auf ein Objekt o ist definiert als (D, o, a), wobei D die Schutzdomäne und a die Zugriffsart ist.

Konzeptuell verwendet ein Betriebssystem eine Matrix-Datenstruktur, um die Zuordnung Objekt-Schutzdomäne zu verfolgen.

	Objekt					
	Datei1	Datei2	Datei3	Datei4	Printer1	Floppy1
Domäne 1	read	read write	read write		write	
2	read write execute		read write	read	write	read

Schutzdomänen

Definition: Eine **Schutzdomäne** ist eine Menge von (Objekt, Rechte) Paaren.



R = read, W = write, X = execute

Verknüpfung eines Prozesses mit einer Schutzdomäne.

zu jedem Zeitpunkt wird ein Prozess in einer Schutzdomäne ausgeführt.

Beispiel Unix: bei Ausführung eines Systemaufrufs wechselt der Prozess vom Benutzermodus in den Systemmodus ("kernel mode") ⇒ entspricht einem Wechsel der Schutzdomäne.

Das Paar (Prozess P, Schutzdomäne D) wird als ***Subjekt*** bezeichnet.

Der Zugriffswunsch eines Subjektes S auf ein Objekt o ist definiert als (D, o, a), wobei D die Schutzdomäne und a die Zugriffsart ist.

Matrix-Datenstruktur

Schutzmechanismen

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

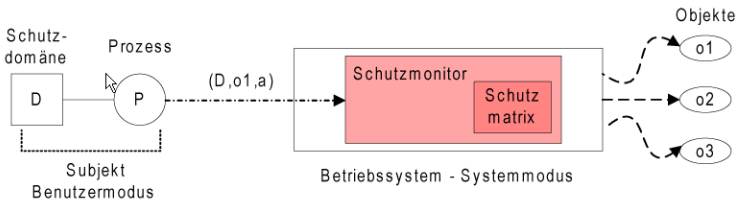
Anforderungen

Ebenen des Zugriffschutzes

Schutzmatrix

Authentifizierung

Jeder Zugriff (D, o, a) eines Subjektes S wird mit Hilfe eines Schutzmonitors überprüft.



der Schutzmonitor ist vertrauenswürdig.

Subjekte können in keinem Fall auf Objekte unter Umgehung des Schutzmonitors zugreifen.

neue Prozesse müssen sich gegenüber dem Schutzmonitor authentifizieren.

Das Konzept der Schutzmatrix wurde von B. Lampson eingeführt. Es verknüpft Schutzdomänen mit den zu schützenden Objekten.

Schutzdomänen

Schutzmonitor

Schutzmatrix ist typischerweise sehr groß und dünn besetzt ⇒ eine direkte Implementierung ist deshalb nicht sinnvoll.

Zugriffskontrollliste

Capability-Liste

Zusammenfassung: Zugriffskontrolllisten und Capability-Listen haben in gewisser Weise komplementäre Eigenschaften

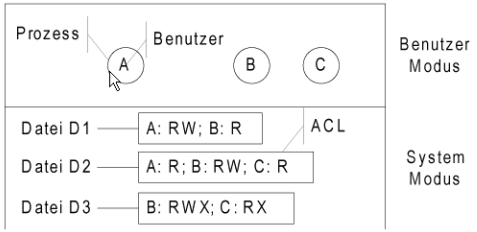
ACLs erlauben das selektive Zurücknehmen von Rechten.

Capabilities können weitergegeben werden.

Zugriffskontrolllisten ("Access Control List", ACL) realisieren die ***spaltenweise*** Speicherung der Schutzmatrix.

jedes Objekt o besitzt seine Zugriffskontrollliste.

Element einer Zugriffskontrollliste (ACL-Element) besteht aus Paar (Prozess, Zugriffsarten).

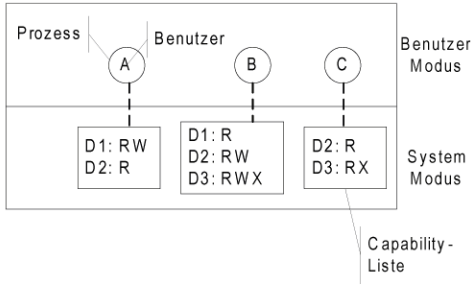


Capability-Liste

Capability-Listen ("Zugriffsausweislisten") realisieren die ***zeilenweise*** Speicherung der Schutzmatrix.

jeder Prozess besitzt eine Menge von Capabilities, die die erlaubten Zugriffe auf Objekte repräsentieren.

Element einer Capability-Liste besteht aus Paar (Objekt, Zugriffsarten).



Capabilities müssen geschützt werden, um Modifikationen durch den Prozess selbst zu verhindern. Alternativen sind

Speicherung im geschützten Bereich des Betriebssystems.

Capabilities sind zwar im Benutzermodus dem Prozess zugeordnet; sie sind jedoch verschlüsselt.

Capabilities können zeitlich begrenzt werden.

Schutzmatrix

Das Konzept der Schutzmatrix wurde von B. Lampson eingeführt. Es verknüpft Schutzdomänen mit den zu schützenden Objekten.

Schutzdomänen

Schutzmonitor

Schutzmatrix ist typischerweise sehr groß und dünn besetzt ⇒ eine direkte Implementierung ist deshalb nicht sinnvoll.

Zugriffskontrollliste

Capability-Liste

Zusammenfassung: Zugriffskontrolllisten und Capability-Listen haben in gewisser Weise komplementäre Eigenschaften

ACLs erlauben das selektive Zurücknehmen von Rechten.

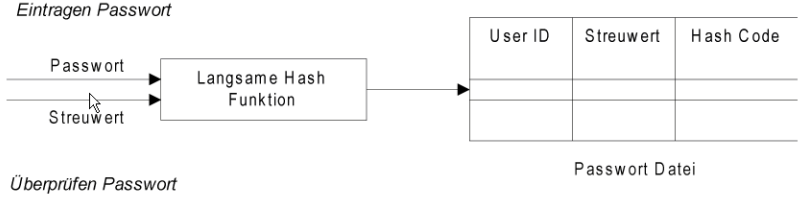
Capabilities können weitergegeben werden.

Authentifizierung

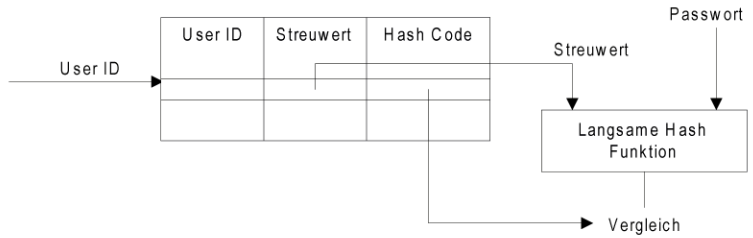
Authentifizierung eines Nutzers erfolgt meist über Login-Name und dem zugehörigen Passwort.

Generierung eines Hash Code aus dem Passwort und einem Streuwert fester Länge.

Eintragen Passwort



Überprüfen Passwort



Ziele des Streuwerts

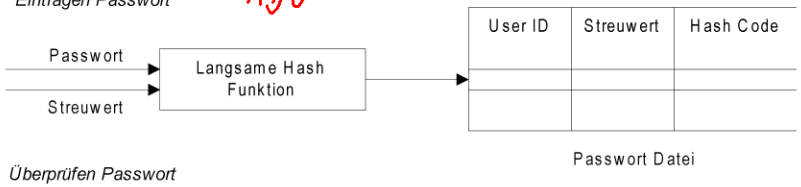
Duplikate von Passwörter sollen in der Passwort Datei nicht erkennbar sein.

Erhöht den Aufwand für offline Attacken auf die Passwort Datei.

Authentifizierung

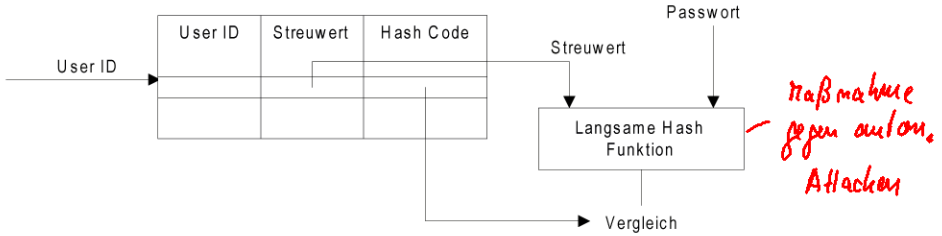
Generierung eines Hash Code aus dem Passwort und einem Streuwert fester Länge.

Eintragen Passwort



Passwort Datei

Überprüfen Passwort



Ziele des Streuwerts

Duplikate von Passwörter sollen in der Passwort Datei nicht erkennbar sein.

Erhöht den Aufwand für offline Attacken auf die Passwort Datei.

nicht erkennbar, ob eine Person dasselbe Passwort auf 2 oder mehreren Systemen nutzt.

Schutzmechanismen

Schutz von gespeicherter Information vor Diebstahl, unerwünschter Manipulation und Verletzung der Vertraulichkeit ist ein zentrales Anliegen in allen Mehrbenutzersystemen.

Ausführung des Applets wird auf einen bestimmten virtuellen Adressbereich beschränkt.

Für eine Sandbox sind die high-order Bits aller Adressen gleich, d.h.

angenommen für einen 32 Bit Adressraum werden 256 Sandboxes auf 16 MByte Grenzen eingerichtet

⇒ für alle Adressen innerhalb einer Sandbox sind die oberen 8 Bits identisch.

jedes Applet erhält zwei Sandboxes: eine Code-Sandbox, eine Daten-Sandbox.

nach dem Laden wird Applet-Code überprüft, ob er Befehle enthält, die ein Verlassen der Sandbox verursachen.

ein Applet, das die Sandbox-Grenzen verletzt, wird zurückgewiesen.

Das Internet führt zunehmend zu einer Verbreitung von mobilem Code. Beispiele sind

Web Seiten mit Applets

Postscript Dateien

mobile Software-Agenten (z.B. in Ecommerce Anwendungen).

Ausführung von heruntergeladenem Code birgt Risiken in sich. Methoden (anhand von Applets), um mit dieser Problematik umzugehen:

Es werden nur Applets von ***vertrauenswürdigen Quellen*** geladen und ausgeführt. Der Applet-Code wird mit einer ***digitalen Unterschrift*** versehen, um zu garantieren, dass der Code der vertrauenswürdigen Quelle nicht verändert wurde.

digitale Unterschrift basiert auf public-key Verfahren.

Erzeugung der Unterschrift durch vertrauenswürdige Quelle

Hashfunktion erzeugt von Applet-Code eine 128/160 bit Zahl.

erzeugte Hashzahl wird mit privatem Schlüssel der Quelle verschlüsselt.

digitale Unterschrift wird mit Applet-Code verschickt.

Überprüfung der Unterschrift

Browser führt auf Applet-Code Hashfunktion aus und berechnet selbst Hashzahl.

Browser entschlüsselt Unterschrift mit öffentlichem Schlüssel der vertauenswürdige Quelle.

berechnete Hashzahl und Hashzahl in Unterschrift müssen übereinstimmen.

Mobiler Code

Das Internet führt zunehmend zu einer Verbreitung von mobilem Code. Beispiele sind

Web Seiten mit Applets

Postscript Dateien

mobile Software-Agenten (z.B. in Ecommerce Anwendungen).

Ausführung von heruntergeladenem Code birgt Risiken in sich. Methoden (anhand von Applets), um mit dieser Problematik umzugehen:

Sandboxing

Interpretation

Signed Code

Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

Einführung

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

Hauptaspekte

Probleme

Schnittstellenentwurf

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

Leitlinien für den Entwurf

Paradigmen der Systemaufrufschnittstelle

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

Algorithmischer Ansatz

Ereignis-basierter Ansatz

Datenparadigma

Systemaufrufschnittstelle

Weitere Implementierungsaspekte

Trends beim Entwurf von Betriebssystemen

Man kann die folgenden 4 Hauptaspekte unterscheiden

Definierte Abstraktion:

z.B. Thread, Prozess, Datei

Bereitstellen einfacher Operationen:

Operationen zum Manipulieren der zur Abstraktion gehörigen Datenstrukturen.

Ansprechbar über Systemaufrufe.

Sicherstellen der Abgrenzung:

Eingrenzung von Fehlern durch Abgrenzung von Prozessen.

Verwalten der Hardware.

Hardware hat sich gemäß des Moore'schen Gesetzes kontinuierlich verbessert; Betriebssysteme haben zwar mehr Funktionalität, jedoch bzgl. Verfügbarkeit sind sie teilweise schlechter als die alten Systeme.

- Betriebssysteme sind sehr komplex.
- Betriebssysteme müssen mit Parallelität umgehen.
- Betriebssysteme müssen mit feindlichen Nutzern umgehen.
- Nutzer möchten Daten mit anderen Nutzern gemeinsam nutzen.
- Betriebssysteme existieren eine lange Zeit.
- Entwickler wissen oft nicht vorab, wie ihre Systeme genutzt werden.
- Betriebssysteme sind portabel für unterschiedliche Hardware entworfen.
- Kompatibilität mit älteren Betriebssystem Versionen.