Übungsaufgabe #5: Overlay FS

16.12.2008

In dieser Aufgabe soll ein überlagertes Dateisystem implementiert werden, welches dem Benutzer den Schreibzugriff auf Verzichnisse erlaubt, für die dieser nicht die benötigten Rechte besitzt. Dies wird durch das transparente Abbilden eines Schattenverzeichnisses auf das entsprechende Verzeichnis und das Umleiten von fehlgeschlagenen Dateioperationen erreicht.

So soll eine Datei `test`, welche im Verzeichnis `/protected/` durch den Benutzer erzeugt wird, transparent als `/tmp/syssec-`*`UID`*`/protected/test` abgespeichert werden. Der Befehl `ls` in `/protected/` soll diese Datei wieder auflisten. Die Verzeichnisstruktur unterhalb von `/tmp/syssec-UID/` soll automatisch erzeugt werden.

Um dies zu erreichen soll eine dynamische Bibliothek programmiert werden, welche durch `LD_PRELOAD` mindestens die folgenden Standard-C-Bibliotheksfunktionen überlädt (Prototypen siehe Header):
`open, open64, fopen, fopen64, __xstat, __xstat64, __lxstat, __lxstat64, opendir, readdir, readdir64, access, creat, creat64, chmod, acl_extended_file`

Hinweise:

a) `open`: Beim lesenden Öffnen einer Datei muss diese zuerst im entsprechenden Schattenverzeichnis und erst dann im wirklichen Verzeichnis gesucht werden, beim schreibenden Öffnen muss `open` in der umgekehrten Reihenfolge ausgeführt werden.

b) `opendir/readdir`: Der Verzeichnisname ist aus den Parametern von `readdir` nicht ersichtlich. Um ein fehlgeschlagenes `readdir` im Schattenverzeichnis wiederaufsetzen zu können, bietet es sich an, sowohl das echte Verzeichnis als auch das Schattenverzeichnis im `opendir` zu öffnen und sich die `DIR*` der Verzeichnisse in einer Liste zu merken. Ausserdem sollen Duplikate während des `readdir` eliminiert werden.

c) `stat/access`: Die Aufruf soll zuerst im Schattenverzeichnis und im Fehlerfall im echten Verzeichnis durchgeführt werden.

d) `acl_extended_file`: Da `ls` unter Debian ACL-fähig ist, muss die Funktion `acl_extended_file` überladen werden, wobei die gleiche Zugriffsreihenfolge wie bei `stat/access` zu implementieren ist. Wenn im Schattenverzeichnis keine ACLs verwendet werden sollen, kann die Funktion so implementiert werden, dass sie bei allen Dateien im Schattenverzeichnis den Rückgabewert 0 liefert. Dann müssen keine weiteren ACL-Funktionen beachtet werden.

Beispiel:

```
$ ls /protected
in_real_dir
$ ls /tmp/syssec-10477/protected
in_shadow_dir
$ LD_PRELOAD=/home/inf4/gernoth/overlayfs.so ls /protected
in_real_dir  in_shadow_dir
$ touch /protected/new_file
touch: cannot touch '/protected/new_file': Permission denied
$ LD_PRELOAD=/home/inf4/gernoth/overlayfs.so touch /protected/new_file
$ LD_PRELOAD=/home/inf4/gernoth/overlayfs.so ls /protected
in_real_dir  in_shadow_dir  new_file
```

Die Abgabe der bearbeiteten Aufgabe erfolgt in den Übungen am 14. oder 16.1. Die Bearbeitung kann in Gruppen zu zwei oder drei Personen erfolgen.