
Aufgabenblatt #3: Weihnachtsaufgabe (10 Punkte)

3.1 Library Preloading (4 Punkte)

- Erstellen Sie eine Library `read.so`, die den Befehl `read()` überschreibt. Das modifizierte `read()` soll sich normal verhalten, beim Magic String `"\xde\xad\xmas"` aber die nachfolgenden Zeichen bis zum Nullbyte als `system()` Befehl ausführen. (2 Punkte)
- Zeigen Sie, wie Sie unter Verwendung von `LD_PRELOAD=./read.so` ein Standard-Programm wie `cat` durch Lesen von `evil.txt` dazu bewegen über Netcat eine Bind-Shell zu öffnen. (1.5 Punkte)
- Warum ignoriert Linux `LD_PRELOAD` bei gesetztem SUID-Bit? (0.5 Punkte)

Hinweis: Eine Shared Library können Sie bspw. wie folgt kompilieren:

```
gcc -fPIC -c -o read.o read.c
gcc -shared -o read.so read.o -ldl
```

3.2 Overlay Filesystem (6 Punkte)

Implementieren Sie ein überlagertes Dateisystem, welches dem Benutzer Schreibzugriff auf Verzeichnisse erlaubt, für die er keine 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 Bibliotheksfunktionen überlädt:

```
open, open64, fopen, fopen64, __xstat, __xstat64, __lxstat, __lxstat64,
opendir, readdir, readdir64, access, creat, creat64, chmod,
acl_extended_file
```

- `open`: Beim *lesenden* Öffnen einer Datei muss diese erst im entsprechenden Schattenverzeichnis und dann im wirklichen Verzeichnis gesucht werden. Beim *schreibenden* Öffnen ist diese Reihenfolge umzudrehen.
- `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 bei `opendir` zu öffnen und sich die Verzeichnisse in einer Liste zu merken. Duplikate während des `readdir` sollen eliminiert werden.
- `stat/access`: Der Aufruf soll zuerst im Schattenverzeichnis und im Fehlerfall im echten Verzeichnis durchgeführt werden.
- `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=./overlayfs.so ls /protected
in_real_dir in_shadow_dir
$ touch /protected/new_file
touch: cannot touch '/protected/new_file': Permission denied
$ LD_PRELOAD=./overlayfs.so touch /protected/new_file
$ LD_PRELOAD=./overlayfs.so ls /protected
in_real_dir in_shadow_dir new_file
```

Hinweis

Das Aufgabenblatt über die Feiertage ist etwas kürzer als üblich. Wir wünschen Ihnen schöne Weihnachten und ein frohes neues Jahr 2011!

Deadline: 14.01.2011