az eredeti `sys_execve` meghívása után visszatérhetünk. Ha azonban találunk ilyet, akkor kiszámítjuk a programhoz tartozó indexelést, majd ezt összehasonlítjuk az indexelt adatbázissal. Ha itt is van találat, akkor meghívjuk az eredeti `sys_execve`-t és visszatérünk. Ha nincs találat, akkor a próbálkozásról naplóbejegyzést készítünk, és hibajelzéssel visszatérünk.

2. Elfogjuk a `sys_delete_module` hívást. Ha a modul nevével hívtak bennünket, akkor hibajelzéssel térünk vissza. A modult nem engedjük törölni.
3. Elfogjuk a `sys_create_module` hívást és hibajelzéssel térünk vissza. Az új modulok beillesztését megtiltjuk, hiszen nem szeretnénk, hogy egy rossz szándékú programozó modulja elfogja az 1. lépésben említett `sys_execve` hívást.
4. Elfogjuk a `sys_open` hívást, így megelőzhetjük azt, hogy az indexelt adatbázist vagy a naplófájlt bármilyen program illetéktelenül megnyissa írásra.
5. A `sys_unlink` hívás elfogásával megakadályozzuk az idexelt adatbázis vagy a naplófájl törlését.

Tartsuk szem előtt, hogy a fenti módszer nem jelent teljes körű védelmet; de első nekifutásra nem is rossz. Egy rossz szándékú felhasználó például módosíthatja a `/dev/kmem` magszimbólumait, vagy közvetlen eszközeléréssel írhat a lemezre, s így az open megkerülésével is írhat az indexelt adatbázisba. Vagy, mivel ez a példa egy betölthető modul, a behatoló egyszerűen letilthatja a modul rendszerindításkor történő betöltését a `/etc/rc.d` fájlok módosításával. Mindezek mellett még számos más rendszerhívással módosítható vagy törölhető az indexelt adatbázis vagy a naplófájl.

Ami a legfontosabb: legyünk tisztában azzal, hogy a betölthető modulokat a behatoló saját terveinek végrehajtására is felhasználhatja. Például a `sys_execve` függvényhívás elfogásával trójai falovat, a read és write rendszerhívások elfogásával pedig billentyűzetfigyelő eljárást építhet be a rendszerbe. Behatolás esetén a betölthető modulok rugalmassága és hatékonysága tehát komoly veszélyforrást jelent. A Kapcsolódó címek részben felsorolt honlapokon további példaprogramokat is találhatunk a témával kapcsolatban.

Gustavo Rodriguez-Rivera (grr@cs.purdue.edu) a Purdue Egyetem vendégprofesszora és a Geodesic Systems programmérnöke. Érdeklődési körébe az operációs rendszerek, a hálózat- és memóriakezelés tartozik.

Nitesh Dhanjani (dhanjani@dhanjani.com) a Purdue Egyetem végzős hallgatója. Operációs rendszerekkel, hálózatokkal és a biztonsággal foglalkozik. Több cég, így például az Ernst & Young LLP, számára végzett már biztonsági felméréseket, szabadidejében pedig tanácsadást is vállal.

Kapcsolódó címek

Betölthető magmodulok Linux alatt *(1. kép)*
➲ http://packetstorm.securify.com/groups/thc/LKM_HACKING.html

Linux Kernel Module Programming Guide (Ori Pomerantz)
➲ http://howto.tucows.com/LDP/LDP/lkmpg/

Nitesh Dhanjani munkái *(2. kép)*
➲ http://www.dhanjani.com/programming/linuxexechash/

Várakozás elkerülése minden 21. újraindításnál

Mikor a rendszer az újraindításkor befűzi a lemezrészeket, általában húsz alkalommal kimarad az fsck futtatása, a huszonegyedik újraindításkor viszont az összes fájlrendszert leellenőrzi.
Ez a hosszas várakozás bizony bosszantó lehet. Honnan lehet tudni, hogy hányadik befűzésnél tartasz egy bizonyos fájlrendszer esetében? Írd be:

```
# dumpe2fs /dev/hda7 | grep
          ➥'[mM]ount count'
dumpe2fs 1.19, 13-Jul-2000 for EXT2 FS
0.5b, 95/08/09
Mount count:                7
Maximum mount count:        20
```

Látható, hogy a /dev/hda7 az utolsó fsck óta hét alkalommal lett befűzve, és az fsck húsz alkalommal ugorja át az ellenőrzést.
Ha az összes fájlrendszerednek azonos a befűzés-számlálója (mount count), akkor a rendszer valamennyit egyszerre fogja ellenőrizni.
Ezen könnyű segíteni:

```
# umount /dev/hda6
# tune2fs -C 9 /dev/hda6
tune2fs 1.19, 13-Jul-2000 for EXT2 FS
0.5b, 95/08/09
Setting current mount count to 9
# mount /dev/hda6
```

Így a befűzés-számlálót 9-re állíthatod.
Figyelem! A tune2fs programot kizárólag befűzetlen fájlrendszeren futassuk. A tune2fs akkor is végrehajtja feladatát, ha a fájlrendszer használatban van, de gyanítom, hogy ez veszélyes, szóval tőled függ, óvatos leszel-e.
Tételezzük fel, hogy négy fájlrendszered van. Állítsd be a befűzés-számlálókat a következőképpen: 1,6,11,16. Ezáltal az ellenőrzés egyenletesen oszlik meg közöttük.
A fentiek végrehajtásával a várható újraindítási idő azonos marad, csak az indítási idő egyenetlensége csökkent. Az, hogy kedvelni fogod-e ezt a módszert, attól függ, mennyire vagy türelmetlen, de az eredeti megoldásnál kétségkívül jobb.
Ha igazi rögeszmés vagy és szereted, ha sokszor fut az fsck, vagy ha több mint 20 fájlrendszered van, a legnagyobb befűzési számot is megváltoztathatod a `tune2fs -c N` végrehajtásával. Az `N=-1` érték kikapcsolja az ellenőrzést. Jó, ha tudod azt is, hogy a `tune2fs -i 2` jelentése: „kétnaponta ellenőrizz”. Ez akkor jöhet jól, ha például hordozható számítógépet használsz.