Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Mantenir un fitxer obert per molt de temps pot ser un problema de seguretat

Per **Joan Miquel**, ***Joanmi*** (http://www.mallorcaweb.net/joanmiquel)

Creado el 13/04/2013 16:16 modificado el 13/04/2013 16:16

> Ahir paginava (ja no es pot dir "fullejar") el capítol del Manual de Debian que parla sobre la seguretat on es fan vàries reflexions sobre les màquines compromeses i com (intentar) recuperar el màxim d'informació d'una intrussió per així poder evitar que es repeteixi i m'en vaig adonar d'una forma molt senzilla en que es pot falsejar qualsevol log en temps real sense deixar-hi gairebé el menor rastre.
>
> **DISCLAIMER:** No soc cap expert en seguretat. Possiblement sigui un problema arxiconegut i totes les distribucions ja portin sistemes de protecció per evitar-ho. Però no n'he sabut trobar referències i, igualment m'ha parescut interessant reflexionar-hi. Segurament no soc l'únic al món que no hi havia caigut... ;-)

En concret, a l'esmentat capítol s'hi comentava que els intrussos poden arribar a reemplaçar binaris d'aplicacions (o fins i tot el propi kernel) per versions modificades que ocultin informació relativa a l'atac, esborrar fitxers de log, etc...

Tot i que és difícil que no deixin cap rastre perquè si, per exemple, ens canvien el 'ps' per ocultar processos maliciosos, és molt difícil que si fem un 'ps --version' aquesta coincideixi exactament amb la que consta a la BD del dpkg.

Esborrar un log és més barroer però evita (o dificulta) que poguem fer servir a posteriori aquesta informació per seguir-li la pista a l'atacant. I sempre es pot sobreescriure amb un altre, o modificar-lo eliminant només el que no volen que vegem.

Si be "en temps real" seria a priori difícil doncs per poder eliminar aquesta informació primer caldria esperar a que s'hagi escrit i, a més, és molt probable que l'aplicació tengui un bloqueig exclussiu per escriptura sobre el fitxer.

En canvi hi ha una forma molt "tonta" de fer-ho en temps real aprofitant la forma en que gestionen els directoris els sistemes de fitxers basats en inodes (excepte engendres FAT-like, crec que tots):

El procés és senzill: Només cal obrir en mode de lectura el fitxer que volem suplantar, esborrar-lo, crear-ne un de nou amb el mateix nom i anar llegint del primer el que va escriguent l'aplicació i escrivint-ho al segon però filtrant el que no ens interessa que passi.

NOTA: Evidentment, això no és pròpiament un forat de seguretat. Per poder fer-ho primer cal aprofitar una autèntica vulnerabilitat per poder executar descarregar i executar un procés amb els privilegis suficients per fer-ho. Però sí és una tàctica que, arribat aquest punt, un atacant podria fer servir per esborrar el seu rastre i evitar ser detectat i/o localitzat un cop l'hagin descobert.

Llegir del fitxer esborrat??

Sí: Perquè, de fet, no l'hem esborrat, sino eliminat el seu nom de directori. Però els noms de directori no representen els fitxers. Son els inodes els que ho fan i els noms de directori només son simples punters a aquests.

Nosaltres no podem esborrar fitxers. Qui ho fa en realitat és el sistema operatiu quan se n'adona que a un fitxer ja no li queden noms de directori apuntant a ell. Això és el que fa possible els "enllaços durs" a fitxers que no son més que entrades de directori que apunten al mateix inode. Per això podem esborrar qualsevol dels dos indistintament sense

perdre l'altre (ni alliberar l'espaï que ocupen).

Això és possible gràcies al "comptador d'enllaços" que porta cada inode i que s'incrementa i decrementa, respectivament, cada cop que afegim o eliminam un nou enllaç dur (nom de fitxer en algún directori) al fitxer.

El que passa és que aquest comptador d'enllaços no es fa servir només per això: Què passaria si esborrassim un fitxer que encara està sent utilitzat per algún procés? Doncs res: no podem esborrar-lo perquè cada cop que un procés obre un fitxer, el sistema operatiu incrementa el seu comptador d'enllaços i no el torna a decrementar fins que es tanca.

Per tant, suposant que el nom de directori que hem eliminat (perquè és l'únic que podem eliminar), fos el darrer que li quedava a l'inode, el sistema operatiu no esborrarà el fitxer perquè el seu comptador d'enllaços no ha arribat a 0.

Això també significa que, tot i que el procés encara pot escriure al fitxer, nosaltres (tret que coneguem el número d'inode i actuem a temps) ja mai podrem veure el que hi ha escrit perquè el sistema operatiu esborrarà el fitxer tan bon punt el procés el tanqui. El somni de tot intrús...

Demostració:

Per veure millor el que dic, vos pos dos senzills scripts en python:

- myApp.py: Que simula una aplicació que escriu logs a un fitxer.
- myIntruder.py: Que reemplaça aquest fitxer per un altre sincronitzat en temps real però filtrant el que no li interessa que vegem.

La forma de provar-ho és senzilla:

1. Si mirau el codi del primer (myApp.py), veureu que escriu, periòdicament, dos logs al fitxer myApp.log: un amb la paraula 'malicious' i un altre sense.
2. Si l'executau i feu un 'less +F myApp.log' des d'un altre terminal veureu que, efectivament, s'escriuen els dos logs.
3. Si executau myIntruder.py sense aturar el primer i tornau a fer un 'less +F myApp.log' des d'un altre terminal (no val el primer si no l'aturau abans perquè ja tenia obert el fitxer original), veureu que els logs amb la paraula 'malicious' han desaparegut i que, a més, continuen entrant nous logs des de myApp.py (si l'aturau veureu que deixaràn de fer-ho), però només els que no duen la paraula 'malicious'.

myApp.py:

```
#!/usr/bin/python

from time import sleep
from os import fsync

def main ():
    log = open('myApp.log', 'a')
    while (1):
        log.write("Connection from good client.n")
        log.write("Connection from malicious client.n")
        log.flush();
        fsync(log);
        sleep (1);

if __name__ == "__main__":
    main()
```

myIntruder.py:

```
#!/usr/bin/python

from time import sleep
from os import unlink
from os import fsync
```

```
from re import search

def main ():
    fname = 'myApp.log'
    ilog = open(fname, 'r')
    unlink(fname)
    olog = open(fname, 'w')
    while (1):
        row = ilog.readline()
        if len(row) and not search('malicious', row):
            olog.write(row)
            olog.flush();
            fsync(olog);
        sleep(1);

if __name__ == "__main__":
    main()
```

Solucions:

A nivell d'aplicació:

Una forma rudimentaria d'evitar que ens pugin fer una jugada com aquesta és tancar i tornar obrir el fitxer de log de forma periòdica en breus intervals de temps. D'aquesta manera, encara que no evitariem que ens poguessin falsejar l'històric, sí que "aturariem" el procés perquè el procés intrús continuaria escoltant el mateix fitxer mentre nosaltres escriuriem dirèctament al nou.

Una mica més refinat seria llegir el número d'inode del nostre fitxer al principi del procés i periòdicament verificar que l'entrada de directori corresponent continua apunant al mateix inode. D'aquesta manera podriem detectar la intrussió i actuar en conseqüència (avisar l'administrador, reemplaçar de nou l'entrada de directori perquè torni a apuntar al nostre inode enganant així el procés intrús, matar-lo si tenim privilegis suficients -tot i que així podríem alertar l'atacant que ha estat descobert-, etc...).

A nivell de sistema operatiu:

Es podria implementar algún sistema de "vigilància" que verificàs periòdicament la correspondència entre els noms dels fitxers de log i els seus inodes. Tot i que això implicaria tenir en compte, per a cada serveï, els possibles canvis de nom legítims debuts a processos d'arxivat, etc...

E-mail del autor: joanmi _ARROBA_ bulma.net