Security through obscurity

a review of FreeBSD's lesser known security capabilities

David Malone <dwmalone@freebsd.org>

2006-03-23

Overview

- Not covering: jail, ACLs.
- Older: file flags, secure levels.
- Newer: MAC (seeotheruid, BSDextended, portacl).
- Newer: GEOM (GBDE).

BSD File Flags

arch Archived (app flag).

opaque Directory is opaque in union.

nodump Dump level > 0 skips.

u/sappnd Append-only.

u/schg Immutable.

u/sunlnk Undeletable.

1 bit extended attributes.

Using file flags

- Set with `chflags`
- List with `ls -lo`

```
dwmalone@hostname% ls -ldo . normal undeleteable
drwxr-xr-x  19 dwmalone  wheel  -      3584 May 14 09:03 .
-rw-r--r--   1 lmalone   wheel  -         0 May 14 09:03 normal
-rw-r--r--   1 lmalone   wheel  uunlnk    0 May 14 09:03 undeleteable
dwmalone@hostname% rm normal undeleteable
override rw-r--r--  lmalone/wheel for normal? y
override rw-r--r--  lmalone/wheel uunlnk for undeleteable? y
rm: undeleteable: Operation not permitted

root@hostname# rm undeleteable
rm: undeleteable: Operation not permitted
root@hostname# chflags nouunlnk undeleteable
root@hostname# rm undeleteable
```

Stop hardlink tricks (cf `security.bsd.hardlink_check_uid`)

BSD Secure Levels

- Flags protect accidents(u), malicious(o).
- Secure level protect malicious root.
- Level can be raised, but not lowered while running.
- Increasing restrictions on root $\Rightarrow$ increasing problems for administrators.

If secure level > 0 then you can't:

- Access hardware (`/dev/mem`, `/dev/pci`, ...)
- Load or unload kernel modules,
- Change system level file flags,
- Run a debugger on init,
- Manually reseed `/dev/random`.

If secure level > 1 then you can't:

- Open disks for writing,
- Change firewall rules,
- Run the clock faster $> \times \approx 2$ or $< \times 0$.

If secure level > 2 then you can't:

- Change secondary firewall features,
- Change certain sysctls.

Enable at boot:

```
kern_securelevel_enable="YES"
kern_securelevel="2"
```

Pros/Cons:

- Could secure boot process,
- Often not quite what you want,
- Overmounting an issue.

Lowering Secure Level

```
root@hostname# sysctl kern.securelevel=2
kern.securelevel: -1 -> 2
root@hostname# KDB: enter: manual escape to debugger
[thread pid 13 tid 100001 ]
Stopped at      kdb_enter+0x2f: nop
db> write securelevel 0
securelevel                0x2    =          0
db> continue
root@hostname# sysctl kern.securelevel
kern.securelevel: 0
```

MAC Framework

- Ask module(s) if operation permitted.
- Often uses labels and extattrs.
- Can implement Biba, MLS, SELinux, ...
- Can also implement more simple policies.
- `option MAC`
- Some modules loadable, some not.

MAC Checks

bpfdesc_receive cred_relabel cred_visible ifnet_relabel ifnet_transmit
inpcb_deliver sysv_msgmsq sysv_msgrcv sysv_msgrmid sysv_msqget
sysv_msqsnd sysv_msqrcv sysv_msqctl sysv_semctl sysv_semget sysv_semop
sysv_shmat sysv_shmctl sysv_shmdt sysv_shmget kenv_dump kenv_get
kenv_set kenv_unset kld_load kld_stat kld_unload mount_stat pipe_ioctl
pipe_poll pipe_read pipe_relabel pipe_stat pipe_write posix_sem_destroy
posix_sem_getvalue posix_sem_open posix_sem_post posix_sem_unlink
posix_sem_wait proc_debug proc_sched proc_setuid proc_seteuid
proc_setgid proc_setegid proc_setgroups proc_setreuid proc_setregid
proc_setresuid proc_setresgid proc_signal proc_wait socket_accept
socket_bind socket_connect socket_create socket_deliver socket_listen
socket_poll socket_receive socket_relabel socket_send socket_stat
socket_visible sysarch_ioperm system_acct system_nfsd system_reboot
system_settime system_swapon system_swapoff system_sysctl vnode_access
vnode_chdir vnode_chroot vnode_create vnode_delete vnode_deleteacl
vnode_deleteextattr vnode_exec vnode_getacl vnode_getextattr
vnode_link vnode_listextattr vnode_lookup vnode_mmap vnode_mmap_downgrade
vnode_mprotect vnode_open vnode_poll vnode_read vnode_readdir
vnode_readlink vnode_relabel vnode_rename_from vnode_rename_to
vnode_revoke vnode_setacl vnode_setextattr vnode_setflags vnode_setmode
vnode_setowner vnode_setutimes vnode_stat vnode_write

mac_seeotheruids

Restrict seeing other proc/sockets to group wheel.

```
% ps -aux | wc
      75     863    5809
# kldload mac_seeotheruids
# sysctl security.mac.seeotheruids.enabled=1
security.mac.seeotheruids.enabled: 0 -> 1
# sysctl security.mac.seeotheruids.specificgid=0
security.mac.seeotheruids.specificgid: 0 -> 0
# sysctl security.mac.seeotheruids.specificgid_enabled=1
security.mac.seeotheruids.specificgid_enabled: 0 -> 1
% ps -aux | wc
       6      69     440
```

mac_portacl

Say who can bind to what port.

```
# kldload mac_portacl
# sysctl security.mac.portacl.rules=uid:80:tcp:80,uid:80:tcp:443
security.mac.portacl.rules:  -> uid:80:tcp:80,uid:80:tcp:443
```

Need to relax normal rules.

```
# sysctl net.inet.ip.portrange.reservedlow=0
net.inet.ip.portrange.reservedlow: 0 -> 0
# sysctl net.inet.ip.portrange.reservedhigh=0
net.inet.ip.portrange.reservedhigh: 1023 -> 0
```

Apache Example

```
Listen 0.0.0.0:80
LockFile /var/log/www/accept.lock
PidFile /var/log/www/httpd.pid
ErrorLog /var/log/www/httpd-error.log
CustomLog /var/log/www/httpd-access.log combined
```

BSDextended

- Standard process credentials.
- Standard file ownership.
- Allow more complex rules than u/g/o match.
- Rules are global, like firewall not file ACL.
- Can be used for sandboxing.

Rules

- Specify the subject (process by uid/gid).
- Specift the object (a file by uid/gid).
- Say what's permitted (rwxsan).

```
# kldload mac_bsdextended
# ugidfw add subject uid pproxy object uid pproxy mode srx
# ugidfw add subject uid pproxy object not uid pproxy mode n
```

Before

```
% ./ls -l
total 8068
-r-xr-xr-x  1 pproxy  wheel  4096352 Apr 30 12:07 cat
-r-xr-xr-x  1 pproxy  wheel  4096352 Apr 30 12:07 ls
-rw-r--r--  1 pproxy  wheel        6 Apr 30 12:27 myfile
-rw-r--r--  1 root    wheel        4 Apr 30 12:27 otherfile
% ./cat myfile
hello
% ./cat otherfile
bye
```

After

```
% ./ls -l
ls: otherfile: Permission denied
total 8066
-r-xr-xr-x  1 3007  0  4096352 Apr 30 12:07 cat
-r-xr-xr-x  1 3007  0  4096352 Apr 30 12:07 ls
-rw-r--r--  1 3007  0        6 Apr 30 12:27 myfile
% ./cat myfile
hello
% ./cat otherfile
cat: otherfile: Permission denied
```

Extended BSDextended

- More criterion for matching on.
- Subject uid, gid, jailid.
- Object on uid, gid, filesys, suid, sgid, type.

```
# ugidfw add subject uid 1000:9000 object uid 0:99 mode rsx
# ugidfw add subject uid 1000:9000 object uid_of_subject mode arswx
# ugidfw add subject uid 1000:9000 object mode n
# ugidfw add subject uid httpd object fs /local mode n
# ugidfw add subject uid httpd object suid mode n
```

GEOM

- System for providing disklike objects.
- Partitions: apple, bsd, gpt, mbr, ...
- Storage: concat, mirror, vinum, ...
- More exotic: gate, uzip, bde, eli, ...

GBDE: Setup

```
# gbde init /dev/ad0s1g -L /etc/ad0s1g.lock
Enter new passphrase:
Reenter new passphrase:
# gbde attach ad0s1g -l /etc/ad0s1g.lock
Enter passphrase:
# newfs /dev/ad0s1g.bde
# mount /dev/ad0s1g.bde /stuff
# df /stuff
Filesystem      Sizes Used  Avail Capacity  Mounted on
/dev/ad0s1g.bde   60G   4k    55G     0%    /stuff
```

GBDE: Use

Mounting Filesystems at Boot

```
% fgrep /stuff /etc/fstab
/dev/ad0s1g.bde  /stuff  ufs  rw  2  2
% fgrep gbde /etc/rc.conf
gbde_devices="AUTO"
```

Automatic swap:

```
% fgrep swap /etc/fstab
/dev/ad0s1b.bde  none  swap  sw  0  0
% fgrep gbde /etc/rc.conf
gbde_swap_enable="YES"
```

Thanks!