

Re: [patch] lockf(3) user-exploitable kernel panic

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-04/0033.html>

From: Jilles Tjoelker (*jilles_at_stack.nl*)

Date: 04/14/04

Date: Wed, 14 Apr 2004 13:28:00 +0200
To: "Devon H. O'Dell" <dodell@sitetronics.com>

On Wed, Apr 14, 2004 at 10:26:32AM +0200, Devon H. O'Dell wrote:

> 1) I have to pass a struct proc * to change_ruid. If a user changes
> his/her uid, the number of advisory-mode locks needs to be transferred
> to the new uid and the only way I figured to do that would be to
> count the number of advisory-mode locks held by a process (I didn't need
> to track this across a fork() since POSIX locks are not inherited
> between processes). This means I have to move the definition out of
> /sys/sys/ucred.h and into /sys/kern/kern_prot.c. It also means that
> OSF/1 compatibility becomes broken on Alpha, since the setuid() in
> osf1_misc.c calls the change_[r/e/sv]uid functions that have been
> implemented in /sys/kern/kern_prot.c. Solutions to this include:
> a) creating a SuSv3-compatible setuid for use with the OSF/1, SVR4
> and Linux compat ABIs (since using the BSD setuid for these also isn't
> totally correct).
> b) require sys/proc.h wherever sys/ucred.h is included (which is
> very ugly)
> c) move the check out of change_ruid() (I don't think this is
> correct since the chgproccnt() function is called there as well)
> d) re-write the OSF/1 compatibility code to use its own
> change_ruid_osf1() function (bloated)

e) add a line 'struct proc;' to sys/ucred.h

> 3) Does this work justify my going through the modified files and doing
> style(9) changes on them? I'm willing to do this; mux@ has encouraged
> it; style(9) suggests that I do it if my code comprises 50% or more of
> the new files (which it doesn't). Again, if this is useful, I'll
> certainly do this.

Some of the files have a mixture of K&R-style and ANSI function definitions.

> 8) Are any of the modifications I've made too intrusive to the
> [proc/advlock/rlimit/sysctl] subsystem(s)?

Rather a lot of functions and programs (setusercontext(3) in libutil, limits(1), rlimit-related builtins in all shells) have knowledge of all

freebsd-arch: Re: [patch] lockf(3) user-exploitable kernel panic

the rlimits built into them. This is already a bit of a problem, for example bash doesn't support the socket buffer size rlimit. Also note that those programs often use single letters for the rlimits.

> 9) *What (extra) suggestions would you have for my patches for relevant manpages?*

> 10) *Have I missed any userland utilities that don't use libutil to check/set classes/limits (perhaps there are some in ports that I can patch as well)?*

limits(1), all shells.

> *This patch is against April 13th -CURRENT but backporting it is very simple since the main affected subsystem doesn't change much architecturally / structurally. However, this also brings into light that this problem may also affect the other BSDs (Dragonfly, Net, Open, Ekko). I cannot verify this as I do not have much experience with these other BSDs and do not know if they impose any limits on the amount of kernel memory a user can have or any other limits which would disallow this to exploit to "work". Should they be affected, what do I need to do to alert them of this?*

Limiting the number of locked regions is not uncommon, e.g. Solaris does it (the manpage seems to indicate a per-system limitation only, though).

Interesting part from Linux getrlimit(2) manpage:

RLIMIT_LOCKS

A limit on the combined number of flock() locks and fcntl() leases that this process may establish (Linux 2.4 and later).

Per-user instead of per-process limits are harder to implement but more effective.

```
> diff -ur lib/libc/sys/getrlimit.2 lib_lockfix/libc/sys/getrlimit.2
> --- lib/libc/sys/getrlimit.2 Tue Apr 13 23:53:52 2004
> +++ lib_lockfix/libc/sys/getrlimit.2 Tue Apr 13 23:58:24 2004
> @@ -98,6 +98,9 @@
> The maximum size (in bytes) of socket buffer usage for this user.
> This limits the amount of network memory, and hence the amount of
> mbufs, that this user may hold at any time.
> .It Li RLIMIT_ADVLOCK
> +The maximum number of POSIX-type (lockf(3) style) advisory-mode
> +locks available to this user.
> .El
> .Pp
> A resource limit is specified as a soft limit and a hard limit. When a
```

Refer to fcntl(2) in preference to lockf(3). While lockf(3) locks typically are implemented using fcntl(2), SUSv3 doesn't say anything about interaction between the two. Also, lockf(3) is marked XSI, but

freebsd-arch: Re: [patch] lockf(3) user-exploitable kernel panic

fcntl(2) locking is not.

The sysctl(3) and sysctl(8) manpages haven't been updated, but I'm not sure whether that's useful.

--

Jilles Tjoelker

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"