/ [base]

Revision 343964

Jump to revision: 343964 Go

Author: kib

Date: Sun Feb 10 17:19:45 2019 UTC *(9 months, 4 weeks ago)*

Changed paths: **16**

Log Message:

```
Implement Address Space Layout Randomization (ASLR)

With this change, randomization can be enabled for all non-
fixed
mappings.  It means that the base address for the mapping is
selected
with a guaranteed amount of entropy (bits). If the mapping
was
requested to be superpage aligned, the randomization honours
the
superpage attributes.

Although the value of ASLR is diminshing over time as exploit
authors
work out simple ASLR bypass techniques, it elimintates the
trivial
exploitation of certain vulnerabilities, at least in theory.
This
implementation is relatively small and happens at the correct
architectural level.  Also, it is not expected to introduce
regressions in existing cases when turned off (default for
now), or
cause any significant maintaince burden.

The randomization is done on a best-effort basis - that is,
the
allocator falls back to a first fit strategy if fragmentation
prevents
entropy injection.  It is trivial to implement a strong mode
where
failure to guarantee the requested amount of entropy results
in
mapping request failure, but I do not consider that to be
usable.

I have not fine-tuned the amount of entropy injected right
```

```
now. It is
only a quantitive change that will not change the
implementation.  The
current amount is controlled by aslr_pages_rnd.

To not spoil coalescing optimizations, to reduce the page
table
fragmentation inherent to ASLR, and to keep the transient
superpage
promotion for the malloced memory, locality clustering is
implemented
for anonymous private mappings, which are automatically
grouped until
fragmentation kicks in.  The initial location for the anon
group range
is, of course, randomized.  This is controlled by
vm.cluster_anon,
enabled by default.

The default mode keeps the sbrk area unpopulated by other
mappings,
but this can be turned off, which gives much more breathing
bits on
architectures with small address space, such as i386.  This
is tied
with the question of following an application's hint about
the mmap(2)
base address. Testing shows that ignoring the hint does not
affect the
function of common applications, but I would expect more
demanding
code could break. By default sbrk is preserved and mmap hints
are
satisfied, which can be changed by using the
kern.elf{32,64}.aslr.honor_sbrk sysctl.

ASLR is enabled on per-ABI basis, and currently it is only
allowed on
FreeBSD native i386 and amd64 (including compat 32bit) ABIs.
Support
for additional architectures will be added after further
testing.

Both per-process and per-image controls are implemented:
- procctl(2) adds PROC_ASLR_CTL/PROC_ASLR_STATUS;
- NT_FREEBSD_FCTL_ASLR_DISABLE feature control note bit makes
it possible
  to force ASLR off for the given binary.  (A tool to edit
the feature
  control note is in development.)
Global controls are:
- kern.elf{32,64}.aslr.enable - for non-fixed mappings done
by mmap(2);
- kern.elf{32,64}.aslr.pie_enable - for PIE image activation
```

```
mappings;
- kern.elf{32,64}.aslr.honor_sbrk - allow to use sbrk area
for mmap(2);
- vm.cluster_anon - enables anon mapping clustering.

PR:     208580 (exp runs)
Exp-runs done by:       antoine
Reviewed by:    markj (previous version)
Discussed with: emaste
Tested by:      pho
MFC after:      1 month
Sponsored by:   The FreeBSD Foundation
Differential revision:  https://reviews.freebsd.org/D5603
```

Changed paths

Path	Details
head/sys/amd64/amd64/elf_machdep.c	modified , text changed
head/sys/arm/arm/elf_machdep.c	modified , text changed
head/sys/compat/freebsd32/freebsd32_misc.c	modified , text changed
head/sys/compat/ia32/ia32_sysvec.c	modified , text changed
head/sys/i386/i386/elf_machdep.c	modified , text changed
head/sys/kern/imgact_elf.c	modified , text changed
head/sys/kern/kern_exec.c	modified , text changed
head/sys/kern/kern_fork.c	modified , text changed
head/sys/kern/kern_procctl.c	modified , text changed
head/sys/sys/imgact.h	modified , text changed
head/sys/sys/proc.h	modified , text changed
head/sys/sys/procctl.h	modified , text changed
head/sys/sys/sysent.h	modified , text changed
head/sys/vm/vm_map.c	modified , text changed
head/sys/vm/vm_map.h	modified , text changed
head/usr.bin/proccontrol/proccontrol.c	modified , text changed

---