September 04, 2017

Mechanism for Fine-Grained Software Controlled Core Dumps

Matthew D. Fleming
Pure Storage, Inc.

Robert Lee
Pure Storage, Inc.

Boris Feigin
Pure Storage, Inc.

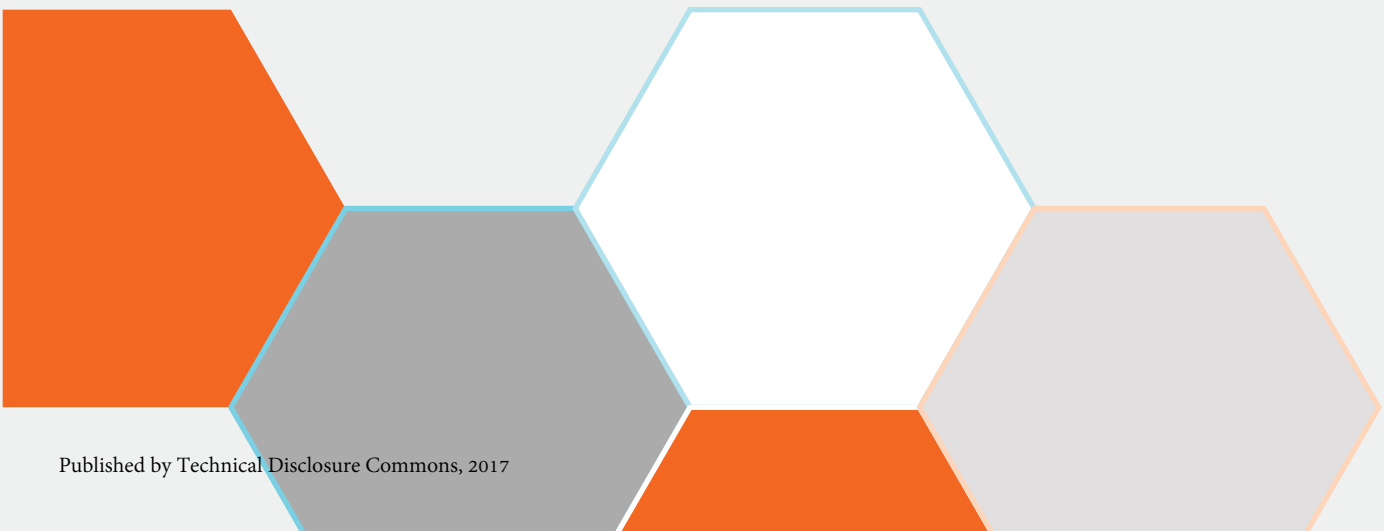
PURE STORAGE DEFENSIVE PUBLICATION

Mechanism for Fine-grained Software Controlled Core Dumps

Matthew D. Fleming

Robert Lee

Boris Feigin



Summary

We present a mechanism to generate a minidump file that will only include the exact memory an application wants when the application receives a fatal signal – the application itself can write a file in valid ELF CORE file format.

Motivation

Kernel core dumps have several problems with somewhat limited solutions. The core dump itself is fantastic for deubugging, but it can take tens of seconds to write if the application has a large amount of memory allocated. In addition, the core dump includes all the application memory.

It is possible to use madvise(MADV_NOCORE) to mark ranges of memory as not being in the core dump, but this must be done before the dump starts. This requires semi-statically knowing which regions of memory should be dumped and which should not. When using a general-purpose memory allocator, regions of virtual memory may become interesting or uninteresting over the lifetime of the system.

Description

At signal time, the faulting thread sends a signal to all other threads, to suspend their execution. The non-faulting threads will store off their register state and stack address using information provided to the signal handling routine. The original faulting thread will wait a brief time for this data to become available, and then write a core file.

The core file can consist of all library and application text and data, as well as the relevant portion of each thread's stack. Additionally, we can look at the contents of the stack and registers, and tread these as "anchor" addresses, writing additional data to the core file that is likely to be of interest. Any register or stack word that points into application memory is likely a valid, live pointer, and the contents of memory near that pointer is potentially of interest.

By writing just the static data, stack, and the memory referenced from anchor addresses, the core dump can be written in a small fraction of the time as the full kernel core dump. Additionally, it is relatively easy to elide certain regions of memory from the dump, such as private keys. The core dump produced is a small fraction of the size of a core dump produced by the kernel, saving space on small boot drives.

PURESTORAGE

Pure Storage, Inc.
Twitter: @purestorage
www.purestorage.com

650 Castro Street, Suite #260
Mountain View, CA 94041

T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com