The History and Future of Crash Dumps in FreeBSD

Sam W. Gwydir
sam@samgwydir.com
vBSDCon 2017

https://github.com/gwydirsam/bsd-coredump-history
Overview

- Background
- Timelines
Background

• Sam Gwydir
• Texas A&M University
  • Computer Engineering/Computer Science
  • Mathematics
• Groupon
• Joyent, Inc.

• I’ve used UNIX-like systems for ~12 years
• OpenBSD then later FreeBSD for the past 5
But Why?

At Work…
• $WORK-1 runs many FreeBSD machines
  • They crash sometimes
• Logs showed crash dumps were larger than swap
  • Dumps were very large
• In fact, some swap partitions were missing altogether!
  • Bug in provisioning script
• How can I get crash dumps without a swap partition?
  • Or just very small swap partitions?
But Why Really?

At School
• Technical Writing Seminar
• Found UNIX History Repo
  • A full history of FreeBSD
  • From “unnamed PDP-11 OS” to FreeBSD 12
• Wrote a paper detailing how a crash dump is made
Motivation

• Understanding how crash dumps work was crucial to solving my missing swap problem
• Deciding on a solution and avoiding reinventing the wheel was important
• UNIX history is always fun
Crash Dump :: A machine readable form of the state of a machine at some point in time, usually after a panic(9).

In English: “What was I thinking?!?”
The History

- The Odyssey of doadump()
- Starts at 6th Edition Research UNIX crash(8)
- Ends at FreeBSD 12-CURRENT’s Encrypted Dump
- Turn to Appendix for a more in depth history
  - Includes architecture support
  - Feature changes and larger bug fixes
- For even more depth, go to the org-mode file on GitHub
  - Includes commits, mailing list emails and copious notes.
“Well in 1979 I can remember doing a crash dump on a Harris S/210 24-bit machine to the line printer in octal, it only took 2 hours to print…”

– rgrimes
Core Dump Output Format Time Line

- 1960: Line Printer (Pre-UNIX)
- 1970: Magnetic Tape (UNIX)
- 1980: Paging Area (4.1 BSD)
- 2000: Network (FreeBSD 4.X)
- 2010:
- 2020:
Core Dump Extension Time Line

- netdump (~2004)
- Minidump (2007)
- Textdump (2008)
- Compressed Dump (2014)
- Encrypted Dump (2016)
- Modular Dump (2016)
- minidumpsz (2016)
Overview

- General Procedure
- FreeBSD
  - Quick How To
  - Full Dump
  - Mini Dump
  - Text Dump
  - Comparison?
How to take a Core Dump in FreeBSD

• Your are purposely panicking your machine
• (Do this in a VM)

# sysrc dumpdev="AUTO" dumpdir="/var/crash"
# mkdir /var/crash # create the dumpdir
# chmod 700 /var/crash # fix permissions
# sysctl debug.kdb.panic=1
General Dump Procedure (4.1 BSD - FreeBSD 12-CURRENT)

- Most OS have at least this functionality

```
• Started by a panic(9), reboot -d
  • sysctl debug.kdb.panic=1
  • dtrace -w -n 'BEGIN{ panic(); }'
• dumpsys() lands all/part of memory on swap in a particular format
• On reboot, savecore(8) writes dump to dumper for analysis
```
What is in a Core Dump?

• Three Types of Core Dumps in FreeBSD
  • Full and Mini Dumps
  • Text Dumps
• Full Dumps and Mini Dump Contents:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>Metadata about dump (time, panic string, hostname)</td>
</tr>
<tr>
<td>core.txt</td>
<td>System info (backtrace, ps, vmstat, netstat, fstat)</td>
</tr>
<tr>
<td>vmcore</td>
<td>core itself</td>
</tr>
</tbody>
</table>
Full Dump On-Disk Format

- Full Dump (FreeBSD 6.0)
  - A classic core dump – the full contents of memory at the time of a crash
  - ELF Format (a.out previous to FreeBSD 6.0)
  - Note padding ahead of dump
    - Some operations use swap on bootup, namely fsck(8)
Mini Dump On-Disk Format

- Mini Dump (FreeBSD 6.2) - Peter Wemm
  - Contains only memory pages in use by kernel
  - Much smaller than the full contents of memory, modern dumps can still be fairly large
  - Custom “minidump” format
What is in a Text Dump?

- textdump(4) :: “The textdump facility allows the capture of kernel debugging information to disk in a human-readable rather than the machine-readable form normally used with kernel memory dumps and minidumps.”

- Added by Robert Watson in FreeBSD 7.1
What is in a Text Dump?

- Textdump Contents:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version.txt</td>
<td>Kernel version string</td>
</tr>
<tr>
<td>panic.txt</td>
<td>Kernel panic message</td>
</tr>
<tr>
<td>msgbuf.txt</td>
<td>Kernel message buffer</td>
</tr>
<tr>
<td>config.txt</td>
<td>Kernel configuration</td>
</tr>
<tr>
<td>ddb.txt</td>
<td>Captured DDB output</td>
</tr>
</tbody>
</table>
Text Dump On-Disk Format

- Text Dump (FreeBSD 7.1)
  - Write Trailer first and last
  - Custom ddb scripting in lieu of a dump
  - Written backwards because size is unknown a priori
  - USTAR format
Core Dumps vs Textdumps

Both
• Useful when crashes aren’t predicted i.e. production
• Operators can debug crashes offline
• Allows archiving of crash data for later comparison

Core Dumps
• Do not need to know what you are looking for ahead of time
• Need source tree, debug symbols and built kernel for analysis

Text Dumps
• Less Complete but much smaller (A few MB vs Many GB)
• Sometimes easier to extract information using DDB over kgdb
Other OS/Tools

- What features do other OS have?
- Can we/should we port those features?
- We’ll cover Mac OS & Illumos

- What tools exist out there for working with crash dumps?
  - backtrace.io
Mac OS X

- Very different from the BSD dump procedure
  - Mach-O
  - Local or remote (network or Firewire)
- netdump - kdumpd(8)
  - Using a modified tftpd(8) from FreeBSD
- Compressed Dump
  - gzip compression
  - Both local and using kdumpd(8)
- Full Procedure in paper
Illumos

Not a BSD but the features are important

- Online dump size estimation
  - Includes different calculations for settings, e.g. compression
- Compressed Dump
  - gzip compression
- Dump to Swap on zvol
  - Versatility of zvols vs partitions
- Live Dump
  - Useful for production machines where interactive debugging is not possible
  - Especially for debugging hangs
backtrace.io

- backtrace.io curates kernel and userspace cores
- Snapshots allow for debugging on a laptop instead of directly on a crashed machine or similar environment
  - Snapshots are even smaller than mini dumps by intelligently choosing segments of dump
- Allows for asking questions like
  - Which panic is most common?
  - Correlated by datacenter, storage controller, hard drive model, timestamp (and more)
Core Dump Extensions

- Modular Dump Code
- netdump
- minidumpsz
- Compressed Dump
- Striped Dump
- Encrypted Dump
Modular Dump Code

- Mix and match features
  - You may need compressed dump but not net dump
  - In progress: rgrimes@ for information
netdump

• Started at Duke by Darrell Anderson
• Holding on since FreeBSD 4.x (~2000*)
• Picked up at Sandvine and later Isilon
• Almost part of FreeBSD 9.0
• markj@ for info

• *thanks Drew Gallatin!
minidumpsz

- Online minidump sizing estimation
- A “no op” version of the mini dump code
  - kernel module
  - minidumpsz for FreeBSD 10 and 11
  - Working on upstreaming
- rgrimes@ for information
Compressed Dump

• Confusing terminology

Pulp Tech Writer @mwlauthor · 18 Dec 2016
With compressed dumps, sounds reasonable.

Sam Gwydir @GwydirSam · 19 Dec 2016
Compressed dumps are here? The only reference I see is lists.freebsd.org/pipermail/free...

Pulp Tech Writer @mwlauthor

Replying to @GwydirSam @DLangille

savecore -Z

8:51 AM · 19 Dec 2016
Compressed Dump

- Confusing terminology

[Diagram showing RAM, dumpsys(), dumpdev (/dev/da0p2), Reboot, savecore -z, dumpdir (/var/crash)]

- Compressed Dump vs Save Compression

[Diagram showing RAM, dumpsys() gzip, dumpdev (/dev/da0p2), Reboot, savecore(8), dumpdir (/var/crash)]
Compressed Dump

- Save Compression
  - gzip dump on the fly before landing in swap
- Compression Ratio 6:1 to 14:1
- A 32 GB Core becomes 5.34 GB
- Fixing the patch so it applies to FreeBSD 12 after encrypted dump will take some work
Striped Dump

- Most setups have a small swap partition on each drive
- Large dumps cannot fit inside a single swap partition
- Why not span the swap partitions?

- Learned about this last night — has not hit paper yet https://lists.freebsd.org/pipermail/svn-src-all/2017-April/143773.html
- In addition Julian hints at being able to do a text dump AND a real dump sequentially
Encrypted Dump

- Kernel Crash Dumps can include sensitive data
- Thus encryption is needed to protect this information
- Encrypted Dump
  - Currently only AES-256-CBC

- dumpon(8) man page example is great
Encrypted Dump

- On disk format is slight altered from minidump
- Kernel Dump Key and Key Size are added to kdh struct
- A kernel dump key consists of an algorithm identifier, an IV and an encrypted symmetric key.
- Panic string is shortened by 4 bytes to allow for this
- Textdumps are not supported, only full and mini dump.

- Not yet in paper. See:
Proposed Core Dump Extensions

- Dump to swap on zvol
- Live Dump
  - A show of hands?
Using the appendix for Research

- https://github.com/gwydirsam/bsd-coredump-history

- Use the org-mode file
  - Includes many of the commit messages, emails, and code referenced
  - Bonus emails from jkh@
  - Includes information not included in the pdf
  - UNIX v5 and other incomplete sections and notes
  - Includes raw notes and various level of detail
  - Code is often included where applicable
  - Usually the file path as well
Using the appendix for Research

• Lets take a look
Links & Thanks

- [github.com/gwydirsam/bsd-coredump-history](https://github.com/gwydirsam/bsd-coredump-history)
- [github.com/dspinellis/unix-history-repo](https://github.com/dspinellis/unix-history-repo)
- [people.freebsd.org/~rgrimes/index.html#kerneldump](http://people.freebsd.org/~rgrimes/index.html#kerneldump)

• Thanks to
  • Deb Goodkin for bringing me into the FreeBSD Community
  • Rodney Grimes for help reading PDP-11 Assembly among other things
  • Michael Dexter for coming up with this idea and for asking me to thank him