

Exploring the FreeBSD/arm64 port

ARM TechCon Hands-On Session

Ed Maste

November 12, 2015

Objectives

- ▶ What is FreeBSD?
- ▶ Logging in and installing packages
- ▶ ZFS
- ▶ DTrace
- ▶ HWPMC (maybe)

What is FreeBSD?

What is FreeBSD?

- ▶ Open Source

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System
- ▶ UNIX

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System
- ▶ UNIX (like), POSIX

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System
- ▶ UNIX (like), POSIX
- ▶ Full system

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System
- ▶ UNIX (like), POSIX
- ▶ Full system
- ▶ Over 30 years of history
 - ▶ FreeBSD started 1993, history to 1972

What is FreeBSD?

- ▶ Open Source Permissively Licensed Operating System
- ▶ UNIX (like), POSIX
- ▶ Full system
- ▶ Over 30 years of history
 - ▶ FreeBSD started 1993, history to 1972

Logging in

TBD, based on equipment for session

- ▶ SSH
- ▶ Console

???

Booting – Console

```
The default boot selection will start in 3 seconds
[1] FreeBSD (efi stub)
[2] Shell
[3] Boot Manager
Start: 2
```

Booting – UEFI

```
EFI Interactive Shell v2.0
```

```
EDK II
```

```
UEFI v2.40 (QEMU EFI Jan  8 2015 16:24:23, 0x00000000)
```

```
Mapping table
```

```
  FS0: Alias(s):HD37b:;BLK2:
```

```
    VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,003E000A00000000)/HD(1,MBR,0x00000000,0x1,0x6
```

```
BLK5: Alias(s):
```

```
    VenHw(F9B94AE2-8BA6-409B-9D56-B9B417F53CB3)
```

```
BLK0: Alias(s):
```

```
    VenHw(8047DB4B-7E9C-4C0C-8EBC-DFBBAACACE8F)
```

```
BLK1: Alias(s):
```

```
    VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,003E000A00000000)
```

```
BLK3: Alias(s):
```

```
    VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,003E000A00000000)/HD(2,MBR,0x00000000,0x641,0
```

```
BLK4: Alias(s):
```

```
    VenHw(837DCA9E-E874-4D82-B29A-23FE0E23D1E2,003E000A00000000)/HD(3,MBR,0x00000000,0x28006
```

```
Press ESC in 5 seconds to skip startup.nsh or any other key to continue.
```

```
Shell> fs0:
```

```
FS0:\> bootaa64
```

Booting – FreeBSD Loader

```
>> FreeBSD EFI boot block
  Loader path: /boot/loader.efi
Consoles: EFI console
Image base: 0x136e95000
EFI version: 2.40
EFI Firmware: QEMU EFI Jan  8 2015 16:24:23 (rev 0.00)

FreeBSD/arm64 EFI loader, Revision 1.1
(emaste@feynman, Wed Oct 21 12:42:56 EDT 2015)
Loading /boot/defaults/loader.conf
/boot/kernel/kernel data=0x6bcde8+0x2f5898 syms=[0x8+0xdbf48+0x8+0xd8a80]
/boot/entropy size=0x1000

Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [/boot/kernel/kernel] in 10 seconds...
```

Booting – FreeBSD Kernel

```
Using DTB provided by EFI at 0x13ffa1000.
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2015 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
    The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 11.0-CURRENT #29 r289313+6dc6440 (arm64.experiment)-dirty: Wed Oct 21 14:13:12 EDT 2015
    emaste@feynman:/tank/emaste/obj/arm64.aarch64/tank/emaste/src/freebsd-arm64/sys/GENERIC arm64
FreeBSD clang version 3.7.0 (tags/RELEASE_370/final 246257) 20150906
WARNING: WITNESS option enabled, expect reduced performance.
CPU(0): ARM Cortex-A57 rlp0
FreeBSD/SMP: Multiprocessor System Detected: 1 CPUs
random: unblocking device.
random: entropy device external interface
ACPI: Table initialisation failed: AE_NOT_FOUND
ACPI: Try disabling either ACPI or apic support.
ofwbus0: <Open Firmware Device Tree>
simplebus0: <Flattened device tree simple bus> on ofwbus0
```

Logging in (Console)

```
Creating and/or trimming log files.  
Starting syslogd.  
Starting casperd.  
Clearing /tmp (X related).  
Updating motd:.  
Mounting late file systems:.  
Starting sendmail_submit.  
Starting sendmail_msp_queue.  
Starting cron.  
Starting background file system checks in 60 seconds.
```

```
Wed Oct 21 18:19:53 UTC 2015
```

```
FreeBSD/arm64 (Amnesiac) (ttyu0)
```

```
login: root
```

```
Password: root
```


Logging in (Console)

```
Oct 21 18:21:43 login: ROOT LOGIN (root) ON ttyu0
FreeBSD 11.0-CURRENT (GENERIC) #29 r289313+6dc6440(arm64.experiment)-dirty: Wed Oct 21 14:13:12 EDT 2015
```

```
Welcome to FreeBSD!
```

```
Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:  https://www.FreeBSD.org/security/
FreeBSD Handbook:     https://www.FreeBSD.org/handbook/
FreeBSD FAQ:          https://www.FreeBSD.org/faq/
Questions List:       https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:      https://forums.FreeBSD.org/
```

```
Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.
```

```
Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier
```

```
Edit /etc/motd to change this login announcement.
root@:~ #
```

Logging in (ssh)

```
% ssh root@192.168.11.12
Password for root@: root
Last login: Wed Oct 21 18:36:38 2015
FreeBSD 11.0-CURRENT (GENERIC) #29 r289313+6dc6440(arm64-dev): Wed Oct 21 14:13:12 EDT 2015

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook:    https://www.FreeBSD.org/handbook/
FreeBSD FAQ:         https://www.FreeBSD.org/faq/
Questions List:      https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:     https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

Edit /etc/motd to change this login announcement.
root@:~ #
```

Bootstrapping the `pkg` package manager

On FreeBSD third-party packages are managed by the `pkg` package manager. `pkg` itself is a part of the third-party repository, and needs to be bootstrapped before first use.

```
root@:~ # pkg bootstrap
The package management tool is not yet installed on your system.
Do you want to fetch and install it now? [y/N]: y
Bootstrapping pkg from http://dirty.ysv.freebsd.org/packages/11aarch64-default, please wait...
Installing pkg-1.5.6...
Extracting pkg-1.5.6: 100%
Message for pkg-1.5.6:
If you are upgrading from the old package format, first run:

    # pkg2ng
root@:~ #
```

Installing packages

```
root@:~ # pkg install zsh tmux git nano?
Updating arm64 repository catalogue...
Fetching meta.txz: 100%   264 B   0.3kB/s   00:01
Fetching packagesite.txz: 100%   3 MiB 890.0kB/s   00:04
Processing entries: 100%
arm64 repository update completed. 16750 packages processed.
Updating database digests format: 100%
The following 24 package(s) will be affected (of 0 checked):
```

```
New packages to be INSTALLED:
```

```
zsh: 5.1 [arm64]
tmux: 2.0_1 [arm64]
git: 2.5.1 [arm64]
libevent2: 2.0.22_1 [arm64]
expat: 2.1.0_3 [arm64]
:
:
```

```
The process will require 158 MiB more space.
32 MiB to be downloaded.
```

```
Proceed with this action? [y/N]: y
```

Installing packages (fetching)

```
Fetching zsh-5.1.txz: 100%    4 MiB 938.1kB/s    00:04
Fetching tmux-2.0_1.txz: 100% 188 KiB 192.9kB/s    00:01
Fetching git-2.5.1.txz: 100%   3 MiB 768.3kB/s    00:04
Fetching libevent2-2.0.22_1.txz: 100% 204 KiB 208.6kB/s    00:01
Fetching expat-2.1.0_3.txz: 100%  82 KiB  84.3kB/s    00:01
Fetching p5-Authen-SASL-2.16_1.txz: 100%  44 KiB  45.1kB/s    00:01
Fetching p5-GSSAPI-0.28_1.txz: 100%   38 KiB  38.7kB/s    00:01
Fetching perl5-5.20.2_6.txz: 100% 13 MiB  1.1MB/s    00:12
Fetching p5-Digest-HMAC-1.03_1.txz: 100%   9 KiB   9.5kB/s    00:01
Fetching p5-Net-SMTP-SSL-1.03.txz: 100%  11 KiB  10.8kB/s    00:01
Fetching p5-IO-Socket-SSL-2.016.txz: 100% 142 KiB 145.1kB/s    00:01
Fetching p5-Mozilla-CA-20141217.txz: 100% 140 KiB 143.2kB/s    00:01
Fetching p5-Net-SSLeay-1.70.txz: 100% 215 KiB 220.3kB/s    00:01
Fetching p5-Socket-2.020.txz: 100%  36 KiB  37.2kB/s    00:01
Fetching p5-IO-Socket-IP-0.37.txz: 100%  27 KiB  27.4kB/s    00:01
Fetching python27-2.7.10.txz: 100%  10 MiB  1.3MB/s    00:08
Fetching readline-6.3.8.txz: 100% 269 KiB 275.7kB/s    00:01
Fetching indexinfo-0.2.3.txz: 100%   5 KiB   4.8kB/s    00:01
Fetching libffi-3.2.1.txz: 100%  30 KiB  30.4kB/s    00:01
Fetching gettext-runtime-0.19.5.1.txz: 100% 135 KiB 138.0kB/s    00:01
Fetching p5-Error-0.17024.txz: 100%  24 KiB  24.8kB/s    00:01
Fetching curl-7.44.0.txz: 100%   1 MiB 707.8kB/s    00:02
Fetching ca_root_nss-3.20.txz: 100% 335 KiB 343.2kB/s    00:01
Fetching cvsps-2.1_1.txz: 100%  31 KiB  32.0kB/s    00:01
Checking integrity... done (0 conflicting)
```

Installing packages (installing)

```
[1/24] Installing perl5-5.20.2_6...
[1/24] Extracting perl5-5.20.2_6: 100%
[2/24] Installing p5-Socket-2.020...
[2/24] Extracting p5-Socket-2.020: 100%
[3/24] Installing p5-Mozilla-CA-20141217...
[3/24] Extracting p5-Mozilla-CA-20141217: 100%
[4/24] Installing p5-Net-SSLeay-1.70...
[4/24] Extracting p5-Net-SSLeay-1.70: 100%
[5/24] Installing p5-IO-Socket-IP-0.37...
[5/24] Extracting p5-IO-Socket-IP-0.37: 100%
[6/24] Installing indexinfo-0.2.3...
[6/24] Extracting indexinfo-0.2.3: 100%
:
:
[22/24] Installing zsh-5.1...
[22/24] Extracting zsh-5.1: 100%
[23/24] Installing tmux-2.0_1...
[23/24] Extracting tmux-2.0_1: 100%
[24/24] Installing git-2.5.1...
==> Creating users and/or groups.
Creating group 'git_daemon' with gid '964'.
Creating user 'git_daemon' with uid '964'.
[24/24] Extracting git-2.5.1: 100%
```

Installing packages (post-install and messages)

```
Updating /etc/shells
```

```
Message for perl5-5.20.2_6:
```

```
The /usr/bin/perl symlink has been removed starting with Perl 5.20.
```

```
For shebangs, you should either use:
```

```
#!/usr/local/bin/perl
```

```
or
```

```
#!/usr/bin/env perl
```

```
The first one will only work if you have a /usr/local/bin/perl,
```

```
the second will work as long as perl is in PATH.
```

```
Message for ca_root_nss-3.20:
```

```
***** WARNING *****
```

```
FreeBSD does not, and can not warrant that the certification authorities  
whose certificates are included in this package have in any way been  
audited for trustworthiness or RFC 3647 compliance.
```

```
Assessment and verification of trust is the complete responsibility of the  
system administrator.
```

```
:  
:
```

► `pkg info -Da` to review messages for all installed packages

ZFS

- ▶ Combined File system and Logical Volume Manager

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones
- ▶ Quotas and Reservations

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones
- ▶ Quotas and Reservations
- ▶ Transparent Compression (lz4, gzip)

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones
- ▶ Quotas and Reservations
- ▶ Transparent Compression (lz4, gzip)
- ▶ Incremental Replication (zfs send / recv)

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones
- ▶ Quotas and Reservations
- ▶ Transparent Compression (lz4, gzip)
- ▶ Incremental Replication (zfs send / recv)
- ▶ Individual and Inherited Properties

ZFS

- ▶ Combined File system and Logical Volume Manager
- ▶ Integrated redundancy (Mirroring, RAID-Z) - integrity guarantee
- ▶ Pooled Storage
- ▶ Multi-Level Cache (ARC, L2ARC, SLOG)
- ▶ Copy-on-Write
- ▶ Sanpshots and Clones
- ▶ Quotas and Reservations
- ▶ Transparent Compression (lz4, gzip)
- ▶ Incremental Replication (zfs send / recv)
- ▶ Individual and Inherited Properties
- ▶ Fine-grained Delegation

ZFS history

Sun Microsystems / OpenSolaris / Illumos etc. / OpenZFS ZoL
FreeBSD

Creating ...

Sun Microsystems / OpenSolaris / Illumos etc. / OpenZFS ZoL
FreeBSD

Enabling ZFS on FreeBSD

```
root@:~ # edit /etc/rc.conf
- add
zfs_enable="YES"
- save and quit - Esc - Leave Editor - Save

root@:~ # service zfs start
```

Creating a ZFS Storage Pool (basic)

```
root@:~ # gpart add -t freebsd vtbd0
root@:~ # zpool create tank vtbd0s4
root@:~ # zpool status -v tank
  pool: tank
  state: ONLINE
    scan: none requested
config:

      NAME                STATE          READ  WRITE  CKSUM
      tank                 ONLINE         0     0     0
      vtbd0s4              ONLINE         0     0     0

errors: No known data errors
root@:~ #
```

Creating a ZFS Storage Pool (mirror)

```
root@:~ # zpool create tank mirror md0 md1 mirror md2 md3
root@:~ # zpool status -v tank
pool: tank
state: ONLINE
scan: none requested
config:

    NAME      STATE    READ  WRITE CKSUM
    tank      ONLINE   0     0     0
      mirror-0 ONLINE   0     0     0
        md0   ONLINE   0     0     0
        md1   ONLINE   0     0     0
      mirror-1 ONLINE   0     0     0
        md2   ONLINE   0     0     0
        md3   ONLINE   0     0     0

errors: No known data errors
root@:~ #
```

- ▶ For reference only (we'll use a mirror)

Creating a ZFS Storage Pool (raidz2)

```
root@:~ # zpool create tank raidz2 md0 md1 md2 md3
root@:~ # zpool status -v
pool: tank
state: ONLINE
scan: none requested
config:

    NAME      STATE    READ  WRITE CKSUM
    tank      ONLINE   0     0     0
      raidz2-0 ONLINE   0     0     0
        md0   ONLINE   0     0     0
        md1   ONLINE   0     0     0
        md2   ONLINE   0     0     0
        md3   ONLINE   0     0     0

errors: No known data errors
root@:~ #
```

- ▶ For reference only (we'll use a mirror)

Creating a ZFS File System

```
root@:~ # zfs create tank/home
root@:~ # zfs set compression=on tank/home
root@:~ # zfs create tank/home/bob
root@:~ # zfs create tank/home/emaste
root@:~ # zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank	134K	17.3G	19K	/tank
tank/home	57K	17.3G	19K	/tank/home
tank/home/bob	19K	17.3G	19K	/tank/home/bob
tank/home/emaste	19K	17.3G	19K	/tank/home/emaste

```
root@:~ #
```



```
root@:~ # dd if=/dev/zero of=/tank/home/bob/emptyfile bs=1M count=128
```

```
128+0 records in
```

```
128+0 records out
```

```
134217728 bytes transferred in 6.410060 secs (20938608 bytes/sec)
```

```
root@:~ # df -h
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/ufs/freebsd_root	20G	2.6G	16G	14%	/
devfs	1.0K	1.0K	0B	100%	/dev
tank	17G	19K	17G	0%	/tank
tank/home	17G	19K	17G	0%	/tank/home
tank/home/bob	17G	19K	17G	0%	/tank/home/bob
tank/home/emaste	17G	19K	17G	0%	/tank/home/emaste

```
root@:~ #
```

ZFS ?

```
root@:~ # zfs set copies=2 tank/home
root@:~ # dd if=/dev/random of=/tank/home/bob/randomfile bs=1M count=4
4+0 records in
4+0 records out
4194304 bytes transferred in 0.427910 secs (9801838 bytes/sec)
root@:~ # df -h
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/ufs/freebsd_root	20G	2.6G	16G	14%	/
devfs	1.0K	1.0K	0B	100%	/dev
tank	17G	19K	17G	0%	/tank
tank/home	17G	19K	17G	0%	/tank/home
tank/home/bob	17G	8.0M	17G	0%	/tank/home/bob
tank/home/emaste	17G	19K	17G	0%	/tank/home/emaste

```
root@:~ #
```

```
\begin{frame}[fragile]
  \frametitle{ZFS snapshots}
  \begin{lstlisting}
root@:~ # zfs snapshot tank/home/bob@2015-11-12
root@:~ # df -h
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/ufs/freebsd_root	20G	2.6G	16G	14%	/
devfs	1.0K	1.0K	0B	100%	/dev
tank	17G	19K	17G	0%	/tank
tank/home	17G	19K	17G	0%	/tank/home
tank/home/bob	17G	8.0M	17G	0%	/tank/home/bob
tank/home/emaste	17G	19K	17G	0%	/tank/home/emaste

```
root@:~ # rm /tank/home/bob/randomfile
root@:~ # zfs diff tank/home/bob@2015-11-12
M /tank/home/bob/
- /tank/home/bob/randomfile
root@:~ # df -h
```

```
Filesystem shortname Size Used Avail Capacity Mounted on
```

Snapshots

```
root@:~ # zpool scrub tank
root@:~ # zpool status
  pool: tank
  state: ONLINE
    scan: scrub repaired 0 in 0h0m with 0 errors on Thu Oct 22 19:00:05 2015
config:

      NAME      STATE    READ  WRITE CKSUM
      tank      ONLINE   0     0     0
          vtbd0s4  ONLINE   0     0     0

errors: No known data errors
root@:~ #
```

Snapshots

```
root@:~ # zfs list  
...
```

What is DTrace?

- ▶ Dynamic tracing framework for software
- ▶ Minimal impact on system performance
- ▶ No (very low) cost when not in use
- ▶ Always available in production

What can DTrace show me?

- ▶ When a function is being called
- ▶ A function's arguments
- ▶ The frequency of function calls
- ▶ A whole lot more...

A Simple Example

```
dtrace -n syscall:::  
dtrace: description 'syscall:::' matched 2144 probes  
CPU      ID          FUNCTION:NAME  
1  51079      ioctl:return  
1  51078      ioctl:entry  
1  51079      ioctl:return  
1  51078      ioctl:entry  
1  51079      ioctl:return  
1  51632      sigprocmask:entry  
1  51633      sigprocmask:return  
1  51784      sigaction:entry
```

- ▶ Look at all system calls

How does DTrace Work?

- ▶ Various probes are added to the system
- ▶ The probes are activated using the dtrace program
- ▶ A small number of assembly instructions are modified at run-time to get the system to run in the probe

A more complex example

```
dtrace -n 'syscall::write:entry /arg2 != 0/ { printf("write size % d\n", arg2); } '  
dtrace: description 'syscall::write:entry ' matched 2 probes  
CPU      ID          FUNCTION:NAME  
0  50978          write:entry write size 1  
0  50978          write:entry write size 55  
0  50978          write:entry write size 2
```

DTrace Glossary

Probe A way of specifying what to trace

Provider A DTrace defined module that provides information about something in the system

Module A software module, such as `kernel`

Function A function in a module, such as `ether_input`

Predicate A way of filtering DTrace probes

Action A set of D language statements carried out when a probe is matched

Providers

- fbt** Function Boundary Tracing (49408)
- syscall** System Calls (2145)
- profile** Timing source
 - proc** Process Operations
- sched** Scheduler
 - io** I/O calls
 - ip** Internet Protocol
- udp** UDP
- tcp** TCP
- vfs** Filesystem Routines

Dissecting a Probe

- ▶ `syscall::write:entry`
 - Provider `syscall`
 - Module `None`
 - Function `write`
 - Name `entry`
- ▶ `fbt:kernel:ether_input`
 - Provider `fbt`
 - Module `kernel`
 - Function `ether_input`
 - Name `entry`

DTrace Requirements

- ▶ A kernel with DTrace support built in
 - ▶ Default on FreeBSD 10 or later
- ▶ The ability to sudo or be root

DTrace Command

- ▶ The complete syntax is covered in the dtrace manual page
- ▶ Requires sudo or that dtrace be installed with setuid

Finding Probes

- ▶ Listing all the probes gets you 50000 to choose from
- ▶ Judicious use of providers, modules and grep
- ▶ e.g. `dtrace -l -P syscall`

The D Language

- ▶ A powerful subset of C
- ▶ Includes features specific to DTrace, such as aggregations
- ▶ Anything beyond some simple debugging usually required a D script

DTrace One Liners

▶ A set of useful single line scripts

```
# Trace file opens with process and filename:
dtrace -n 'syscall::open*:entry { printf("%s %s", execname, copyinstr(arg0)); }'

# Count system calls by program name:
dtrace -n 'syscall:::entry { @[execname] = count(); }'

# Count system calls by syscall:
dtrace -n 'syscall:::entry { @[probefunc] = count(); }'
```

Count System Calls

```
dtrace -n 'syscall:::entry { @[probefunc] = count(); }'  
dtrace: description 'syscall:::entry ' matched 1072 probes
```

```
^C
```

fstat	1
setitimer	1
getpid	2
read	2
sigreturn	2
write	3
getsockopt	4
select	6
sigaction	6
_umtx_op	7
__sysctl	8
munmap	18
mmap	19
sigprocmask	23
clock_gettime	42
ioctl	45

Aggregations

- ▶
dtrace -n 'syscall:::entry { @[probefunc] = count()'
- ▶ The @[probefunc] syntax
- ▶ Aggregates data during a run for later output
- ▶ Extremely powerful feature of D language

Quantization

```
# Summarize requested write() sizes by program name, as power-of-2 distributions (bytes):
dtrace -n 'syscall::write:entry { @[execname] = quantize(arg2); }'
dtrace: description 'syscall::write:entry ' matched 2 probes
^C
```

```
find
value  ----- Distribution -----  count
   1 |
   2 |
   4 |
   8 |
  16 |@          6940
  32 |@          13666
  64 |          59
 128 |          0
```

Probing the stack

- ▶ Find out how we got where we are
- ▶ The `stack()` routine

Who called malloc()?

```
1 29371          malloc:entry
      kernel `cloneuio+0x2c
      kernel `vn_io_fault1+0x3b
      kernel `vn_io_fault+0x18b
      kernel `dofile_read+0x95
      kernel `kern_readv+0x68
      kernel `sys_read+0x63
      kernel `amd64_syscall+0x351
      kernel `0xffffffff80d0aa6b
```

- ▶ Read upwards from the bottom

DTrace Toolkit

- ▶ An open source set of tools written to use D scripts
- ▶ Currently specific to Solaris
- ▶ Exists as a FreeBSD port (thanks to Steve)
- ▶ Currently being updated

An example script: hotkernel

```
./hotkernel  
Sampling... Hit Ctrl-C to end.
```

```
^C
```

FUNCTION	COUNT	PCNT
kernel`lookup	1	0.1%
kernel`unlock_mtx	1	0.1%
kernel`_vm_page_deactivate	1	0.1%
...		
kernel`amd64_syscall	9	0.5%
kernel`pmap_remove_pages	9	0.5%
kernel`hpet_get_timecount	13	0.7%
kernel`pagezero	15	0.8%
kernel`0xfffffffff80	34	1.9%
kernel`spinlock_exit	486	27.0%
kernel`acpi_cpu_c1	965	53.6%

Predicates

- ▶ Filtering probes based on relevant data
- ▶ Useful for excluding common conditions
- ▶ `/arg0 != 0/` Ignore a normal return value

Tracking a Specific Process

- ▶ `pid` is used to track a Process ID
- ▶ Used in predicates
- ▶ `/pid == 1234/`

Running a Program Under DTrace

- ▶ DTrace is most often used on running systems
- ▶ DTrace can be attached at runtime to a program
 - ▶
- ▶ Run a program completely under the control of DTrace
 - ▶

Going too far

- ▶ Overly broad probes slow down the system
 - ▶ Watching everything in the kernel
 - ▶ Registering a probe on a module

The Probe Effect

- ▶ Each probe point has a cost
- ▶ Every action has a reaction
- ▶ Any action code requires time to run
- ▶ Impacts system performance

DTrace Lab Exercises

- ▶ Bring up OSCourse Virtual Machine
- ▶ Find the current list of providers
- ▶ Count the probes available
- ▶ Trace all the system calls used by sshd
- ▶ Summarize requested write() sizes by program name
- ▶ Summarize return values from write() by program name
- ▶ Find and modify three (3) of the DTrace one liners