FreeBSD, Capsicum, GELI and ZFS as key components of a security appliance

Paweł Jakub Dawidek
<p.dawidek@wheelsystems.com>
<pjd@FreeBSD.org>

FUDO

WHEEL SYSTEMS

FreeBSD®

WHEEL SYSTEMS

FreeBSD®

FUDO

WHEEL SYSTEMS

FreeBSD®

Choosing version control system

WHEEL SYSTEMS

FreeBSD®

Choosing version control system

we tried most of them, though call

objective comparison to the rescue:

do they rhyme?

perforce rhymes with "**the right course**"

mercurial doesn't rhyme

git rhymes with ...

subversion rhymes with aversion or perversion

Building from source

Building from source

running **make** in top source directory produces:

fudo[-branch]-<changeset>.img
bootable installation image

fudo[-branch]-<changeset>.upg
signed upgrade image

WHEEL SYSTEMS

FreeBSD®

Building from source

libevent

libpng

net-snmp

pyasn1

PostgreSQL

html5lib

FreeRDP

OpenLDAP

curl

Imaging

rsync

ReportLab

FreeBSD

Django

south

sudo

msmtp

ffmpeg

ucarp

Python

PgBouncer

nginx

xhtml2pdf

flup

FreeType

jpeg

psycopg2

WHEEL SYSTEMS

FreeBSD®

Building from source

```
dbuild freerdp '
        cmake . [...]
        ${MAKE} clean
        ${MAKE} || exit 1
'
```

```
dbuild nginx nginx_tcp_proxy_module '
        ${MAKE} clean
        ./configure [...] || exit 1
        ${MAKE} ${MAKE_FLAGS} || exit 1
'
```

GELI

- block-level data encryption
  - AES-XTS, AES-CBC, Blowfish-CBC, Camellia-CBC, 3DES-CBC
- integrity verification / data authentication
  - HMAC/MD5, HMAC/RIPEMD160, HMAC/SHA{1,256,384,512}
- multifactor user keys
- two independent keys
- ability to encrypt root file system

WHEEL SYSTEMS

FreeBSD®

GELI

- passphrase strengthened using PKCS#5v2
- hardware acceleration (via opencrypto)
- suspend/resume
- ability to use one-time keys

ZFS

- storage pool, integrated volume manager
- dynamic striping
- copy-on-write transactional model
- RAIDZ{1,2,3}
- hybrid storage model
- snapshots and clones
- lightweight file systems creation
- adaptive endianness

ZFS

- compression
- deduplication
- I/O priority with deadline scheduling
- end-to-end checksumming
- intelligent scrubbing and resilvering
- ditto blocks
- file systems and disk-like devices
- per-dataset configuration

Capsicum

kernel infrastructure that provides:

- tight sandboxing (cap_enter(2))
- capabilities

Boot

- keys generation during first boot
- stored on two pen-drives
- one of them required for the following boots
- once the box is running, pen-drive can be removed

File systems layout

34	3907029101	disk0	GPT	(1.8T)
34	6	1	wheelsystems-config	(3.0k)
40	128	2	freebsd-boot	(64k)
168	8388608	3	freebsd-ufs	(4.0G)
8388776	8388608	4	freebsd-ufs [bootme]	(4.0G)
16777384	8388608	5	freebsd-ufs	(4.0G)
25165992	16777216	6	freebsd-swap	(8.0G)
41943208	3865085920	7	freebsd-zfs	(1.8T)

WHEEL SYSTEMS

FreeBSD®

Upgrade

```
  34  3907029101  disk0  GPT  (1.8T)
  34           6      1  wheelsystems-config  (3.0k)
  40         128      2  freebsd-boot  (64k)
 168     8388608      3  freebsd-ufs  (4.0G)
 8388776 8388608      4  freebsd-ufs  [bootme]  (4.0G)
16777384 8388608      5  freebsd-ufs  (4.0G)
25165992 16777216     6  freebsd-swap  (8.0G)
41943208 3865085920   7  freebsd-zfs  (1.8T)
```

Upgrade

```
  34   3907029101   disk0   GPT   (1.8T)
  34            6          1   wheelsystems-config   (3.0k)
  40          128          2   freebsd-boot   (64k)
 168      8388608          3   freebsd-ufs   (4.0G)
 8388776  8388608          4   freebsd-ufs   [bootme]   (4.0G)
 16777384 8388608          5   freebsd-ufs   [bootme,bootonce]   (4.0G)
 25165992 16777216         6   freebsd-swap   (8.0G)
 41943208 3865085920       7   freebsd-zfs   (1.8T)
```

WHEEL SYSTEMS

FreeBSD®

Upgrade

```
  34  3907029101  disk0  GPT  (1.8T)
  34           6      1  wheelsystems-config  (3.0k)
  40         128      2  freebsd-boot  (64k)
 168     8388608      3  freebsd-ufs  (4.0G)
8388776  8388608      4  freebsd-ufs  [bootme]  (4.0G)
16777384 8388608      5  freebsd-ufs  [bootfailed]  (4.0G)
25165992 16777216     6  freebsd-swap  (8.0G)
41943208 3865085920   7  freebsd-zfs  (1.8T)
```

WHEEL SYSTEMS

FreeBSD®

Upgrade

```
  34   3907029101   disk0   GPT   (1.8T)
  34            6          1   wheelsystems-config   (3.0k)
  40          128          2   freebsd-boot   (64k)
 168      8388608          3   freebsd-ufs   (4.0G)
8388776   8388608          4   freebsd-ufs   [bootme]   (4.0G)
16777384  8388608          5   freebsd-ufs   [bootonce]   (4.0G)
25165992  16777216         6   freebsd-swap   (8.0G)
41943208  3865085920       7   freebsd-zfs   (1.8T)
```

WHEEL SYSTEMS

FreeBSD®

Upgrade

```
  34  3907029101  disk0  GPT  (1.8T)
  34           6      1  wheelsystems-config  (3.0k)
  40         128      2  freebsd-boot  (64k)
 168     8388608      3  freebsd-ufs  (4.0G)
 8388776 8388608      4  freebsd-ufs  (4.0G)
16777384 8388608      5  freebsd-ufs  [bootme]  (4.0G)
25165992 16777216     6  freebsd-swap  (8.0G)
41943208 3865085920   7  freebsd-zfs  (1.8T)
```

Upgrade

- /etc/rc.d/gptboot
- /etc/upgrade/UPG<NUMBER>
- /etc/rc.d/copyupg
- /data/upgrade/todo/
- /data/upgrade/done/

ZFS

- a file system
- developed at Sun Microsystems (R.I.P. Why God^WLarry? WHY?!)
- ported to FreeBSD by some awesome, but inconspicuous Polish guy

FUDO goals

- gate to the most valuable and sensitive data
- full accountability
- strong authentication
- secure data storage
- secure protocols handling (many of them)
- performance (cryptography cost x3)

WHEEL SYSTEMS

FreeBSD®

On-disk data protection

- GELI AES-XTS-256
- keys on removable pen-drives
  - generated at customer's place on first boot
  - can be removed once the box is running
- ZFS checksum: SHA256
- AES-NI
- trusted timestamping
- RAIDZ2

WHEEL SYSTEMS

FreeBSD®

ZFS

- one big RAIDZ2
- compression provides twice as much space (soon lz4)
- snapshots used to replicate sessions data

WHEEL SYSTEMS

FreeBSD®

Session protection

- session handled by two processes: master and slave
- master has some privileges
- slave is closed in Capsicum sandbox
- slave runs all protocol logic (ssh, rdp, vnc, etc.)
- master is responsible for authentication
- master provides resources and capabilities to slave

Session protection

when slave is executed it gets:

- 1 second of CPU time
- 5 minutes of wall clock time
- 32MB of memory
- read-only access to configuration/resource directory
- communication socket to its master

Session protection

slave **cannot** access:

- file systems
- network
- process list
- SysV IPC
- global namespaces in general
- most system calls

Session protection

once the user is authenticated by master, slave gets:

- descriptor with open connection to the server
- credentials to log into the server
- additional 32MB of memory
- unlimited CPU time
- append-only descriptor to session dump

Session protection

even with additional capabilities slave **cannot**:

- open network connection to any other server
- access data of any other session
- access any processes in the system
- overwrite already stored session data
- access system configuration or any files in general

Multimaster clustering

- everything replicated async
- session data replicated using zfs send/recv
- session metadata (database) replicated using pair of daemons developed in house

Database replication

Potential problems Solutions Still to do

- what do we lose doing async instead of sync replication
- how to avoid collisions of row IDs within the cluster
- how to distinguish INSERT from DELETE
- how to deal with UPDATE collisions

Database replication

Solutions

Q: What do we lose doing async instead of sync replication?

A: Nothing.

WHEEL SYSTEMS

FreeBSD®

Database replication

Solutions

Q: How to avoid collisions of row IDs within the cluster?

A: ID starts at node serial number * 2^36.

WHEEL SYSTEMS

FreeBSD®

Database replication

Solutions

Q: How to distinguish INSERT from DELETE?

A: Never DELETE.

WHEEL SYSTEMS

FreeBSD®

Database replication

Solutions

Q: How to deal with UPDATE collisions?

A: Maintain modify_at, received_at rows for every table.

```
SELECT * FROM table WHERE

        (modified_on = myserial AND modified_at > nodetime) OR

        (modified_on != myserial AND received_at > nodetime)

ORDER BY created_at
```

WHEEL SYSTEMS

FreeBSD®

Conclusions

- name your VCS wisely
- use ZFS, GELI and Capsium
- don't forget to lock your FUDO panel
- avoid my talks

Questions?

WHEEL SYSTEMS

FreeBSD®

Questions?

WHEEL
SYSTEMS

FreeBSD®