Bastille Documentation

Release beta

Christer Edwards

Nov 25, 2018

Contents

Bastille is a jail automation framework that allows you to quickly and easily create and manage FreeBSD jail.

CHAPTER 1

Installation

Bastille is not (yet) in the official ports tree, but I have built and verified binary packages.

To install using one of the BETA binary packages, copy the URL for the latest release here (TXZ file): https://github.com/bastillebsd/bastille/releases

Then, install via `pkg`. Example:

```
pkg add https://github.com/BastilleBSD/bastille/releases/download/0.3.20181120/
↪bastille-0.3.20181120.txz
```

BETA binary packages are signed. These can be verified with this pubkey:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq28OLDhJ12JmsKKcJpnn
pCW3fFYBNI1BtdvTvFx57ZXvQ2qecBvnR9+XWi83hKS9ALTKZI6CLC2uTv1fIsZl
u6rDRRNZwZFfITACSfwI+7UObMXz3oBZjk94J3rIegk49EyjDswKdVWv5k1EiVXF
SAwXSl2kA2hGfQJkj5NS4nrfoRBc0z6fm+BGdNuHKSTmeZh1dbLEHt9EArD20DJ7
HIr8vUSPLwONeqJCBFA/MeDO+GpwtwA/ldc2ZZy1RCPctdC2NeiGW7oy1yVDu6wp
mHCq8qDfmCx5Aex84rWUf9iH8TM92AWmegTaz2p+BgESctpjNRCUuSEwOCBIO6g5
3wIDAQAB
-----END PUBLIC KEY-----
```

CHAPTER 2

Basic Usage

```
ishmael ~ # bastille -h
Usage:
  bastille command [ALL|glob] [args]

Available Commands:
  bootstrap   Bootstrap a FreeBSD release for jail base.
  cmd         Execute arbitrary command on targeted jail(s).
  console     Console into a running jail.
  cp          cp(1) files from host to targeted jail(s).
  create      Create a new jail.
  destroy     Destroy a stopped jail.
  help        Help about any command
  htop        Interactive process viewer (requires htop).
  list        List jails (running and stopped).
  pkg         Manipulate binary packages within targeted jail(s). See pkg(8).
  restart     Restart a running jail.
  start       Start a stopped jail.
  stop        Stop a running jail.
  sysrc       Safely edit rc files within targeted jail(s).
  template    Apply Bastille template to running jail(s).
  top         Display and update information about the top(1) cpu processes.
  update      Update jail base -pX release.
  upgrade     Upgrade jail release to X.Y-RELEASE.

Use "bastille -v|--version" for version information.
Use "bastille command -h|--help" for more information about a command.
```

CHAPTER 3

0.3-beta

This document outlines the basic usage of the Bastille jail management framework. This release, obviously, is beta quality. I make no guarantees of quality, and if it screws up your system… Sorry, bro.

With all that said, here's how to use this tool in its current beta state…

CHAPTER 4

Network Requirements

In order to segregate jails from the network and from the world, Bastille attaches jails to a loopback interface only. The host system then acts as the firewall, permitting and denying traffic as needed.

First, create the loopback interface:

```
ishmael ~ # sysrc cloned_interfaces+=lo1
ishmael ~ # service netif cloneup
```

Second, enable NAT through the firewall:

```
ishmael ~ # sysrc pf_enable="YES"
```

Create the firewall config, or merge as necessary.

4.1 /etc/pf.conf

```
ext_if="vtnet0"

set block-policy drop
scrub in on $ext_if all fragment reassemble

set skip on lo
nat on $ext_if from !($ext_if) -> ($ext_if:0)

## rdr example
## rdr pass inet proto tcp from any to any port {80, 443} -> 10.88.9.45

block in log all
pass out quick modulate state
antispoof for $ext_if inet
pass in inet proto tcp from any to any port ssh flags S/SA keep state
```

- Make sure to change the `ext_if` variable to match your host system interface.

- Make sure to include the last line (`port ssh`) or you'll end up locked out.

Note: if you have an existing firewall, the key lines for in/out traffic to jails are:

```
nat on $ext_if from lo1:network to any -> ($ext_if)

## rdr example
## rdr pass inet proto tcp from any to any port {80, 443} -> 10.88.9.45
```

The `nat` routes traffic from the loopback interface to the external interface for outbound access.

The `rdr pass ...` will redirect traffic from the host firewall on port X to the ip of Jail Y. The example shown redirects web traffic (80 & 443) to the jails at `10.88.9.45`.

We'll get to that later, but when you're ready to allow traffic inbound to your jails, that's where you'd do it.

Finally, start up the firewall:

```
ishmael ~ # service pf restart
```

At this point you'll likely be disconnected from the host. Reconnect the ssh session and continue.

This step only needs to be done once in order to prepare the host.

CHAPTER 5

bastille bootstrap

The first step is to “bootstrap” a release. Current supported release is 11.2-RELEASE, but you can bootstrap anything in the ftp.FreeBSD.org RELEASES directory.

Note: your mileage may vary with unsupported releases and releases newer than the host system likely will NOT work at all.

To `bootstrap` a release, run the bootstrap sub-command with the release version as the argument.

```
ishmael ~ # bastille bootstrap 11.2-RELEASE
ishmael ~ # bastille bootstrap 12.0-RELEASE
```

This command will ensure the required directory structures are in place and download the requested release. For each requested release, `bootstrap` will download the base.txz and lib32.txz. These are both verified (sha256 via MANIFEST file) before they are extracted for use.

Downloaded artifacts are stored in the `cache` directory. “bootstrapped” releases are stored in `releases/version`.

The bootstrap subcommand is generally only used once to prepare the system. The only other use case for the bootstrap command is when a new FreeBSD version is released and you want to start building jails on that version.

To update a release as patches are made available, see the `bastille update` command.

CHAPTER 6

bastille create

Bastille create uses any available bootstrapped release to create a lightweight jailized system. To create a jail simply provide a name, release and a private (rfc1918) IP address.

- name
- release (bootstrapped)
- ip

```
ishmael ~ # bastille create folsom 11.2-RELEASE 10.8.62.1

RELEASE: 11.2-RELEASE.
NAME: folsom.
IP: 10.8.62.1.
```

This command will create a 11.2-RELEASE jail assigning the 10.8.62.1 ip address to the new system.

I recommend using private (rfc1918) ip address ranges for your jails. These ranges include:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Bastille does its best to validate the submitted ip is valid. This has not been thouroughly tested–I generally use the 10/8 range.

CHAPTER 7

bastille start

To start a jail you can use the `bastille start` command.

```
ishmael ~ # bastille start folsom
Targeting specified jails.
folsom

[folsom]:
folsom: created
```

CHAPTER 8

bastille stop

To stop a jail you can use the `bastille stop` command.

```
ishmael ~ # bastille stop folsom
Targeting specified jails.
folsom

[folsom]:
folsom: removed
```

CHAPTER 9

bastille restart

To restart a jail you can use the `bastille restart` command.

```
ishmael ~ # bastille restart folsom
Targeting specified jails.
folsom

[folsom]:
folsom: removed

Targeting specified jails.
folsom

[folsom]:
folsom: created
```

CHAPTER 10

bastille cmd

To execute commands within the jail you can use `bastille cmd`.

```
ishmael ~ # bastille cmd folsom 'ps -auxw'
Targeting specified jails.
folsom

[folsom]:
USER   PID %CPU %MEM   VSZ  RSS TT  STAT STARTED    TIME COMMAND
root 71464  0.0  0.0 14536 2000  -  IsJ   4:52PM 0:00.00 /usr/sbin/syslogd -ss
root 77447  0.0  0.0 16632 2140  -  SsJ   4:52PM 0:00.00 /usr/sbin/cron -s
root 80591  0.0  0.0 18784 2340  1  R+J   4:53PM 0:00.00 ps -auxw
```

CHAPTER 11

bastille pkg

To manage binary packages within the jail use `bastille pkg`.

```
ishmael ~ # bastille pkg folsom 'install vim-console git-lite zsh'
Targeting specified jails.
folsom

[folsom]:
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 pkg+http://pkg.FreeBSD.org/FreeBSD:10:amd64/quarterly, please wait...
Verifying signature with trusted certificate pkg.freebsd.org.2013102301... done
[folsom] Installing pkg-1.10.5_5...
[folsom] Extracting pkg-1.10.5_5: 100%
Updating FreeBSD repository catalogue...
pkg: Repository FreeBSD load error: access repo file(/var/db/pkg/repo-FreeBSD.sqlite) failed: No such file or directory
[folsom] Fetching meta.txz: 100%    944 B   0.9kB/s    00:01
[folsom] Fetching packagesite.txz: 100%    6 MiB   3.4MB/s    00:02
Processing entries: 100%
FreeBSD repository update completed. 32550 packages processed.
All repositories are up to date.
Updating database digests format: 100%
The following 10 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
    vim-console: 8.1.0342
    git-lite: 2.19.1
    zsh: 5.6.2
    expat: 2.2.6_1
    curl: 7.61.1
    libnghttp2: 1.33.0
    ca_root_nss: 3.40
    pcre: 8.42
```

(continues on next page)

(continued from previous page)

```
    gettext-runtime: 0.19.8.1_1
    indexinfo: 0.3.1

Number of packages to be installed: 10

The process will require 77 MiB more space.
17 MiB to be downloaded.

Proceed with this action? [y/N]: y
[folsom] [1/10] Fetching vim-console-8.1.0342.txz: 100%    5 MiB   5.8MB/s    00:01
[folsom] [2/10] Fetching git-lite-2.19.1.txz: 100%    4 MiB   2.1MB/s    00:02
[folsom] [3/10] Fetching zsh-5.6.2.txz: 100%    4 MiB   4.4MB/s    00:01
[folsom] [4/10] Fetching expat-2.2.6_1.txz: 100%  109 KiB 111.8kB/s    00:01
[folsom] [5/10] Fetching curl-7.61.1.txz: 100%    1 MiB   1.2MB/s    00:01
[folsom] [6/10] Fetching libnghttp2-1.33.0.txz: 100%  107 KiB 109.8kB/s    00:01
[folsom] [7/10] Fetching ca_root_nss-3.40.txz: 100%  287 KiB 294.3kB/s    00:01
[folsom] [8/10] Fetching pcre-8.42.txz: 100%    1 MiB   1.2MB/s    00:01
[folsom] [9/10] Fetching gettext-runtime-0.19.8.1_1.txz: 100%  148 KiB 151.3kB/s    ␣
↪00:01
[folsom] [10/10] Fetching indexinfo-0.3.1.txz: 100%    6 KiB   5.7kB/s    00:01
Checking integrity... done (0 conflicting)
[folsom] [1/10] Installing libnghttp2-1.33.0...
[folsom] [1/10] Extracting libnghttp2-1.33.0: 100%
[folsom] [2/10] Installing ca_root_nss-3.40...
[folsom] [2/10] Extracting ca_root_nss-3.40: 100%
[folsom] [3/10] Installing indexinfo-0.3.1...
[folsom] [3/10] Extracting indexinfo-0.3.1: 100%
[folsom] [4/10] Installing expat-2.2.6_1...
[folsom] [4/10] Extracting expat-2.2.6_1: 100%
[folsom] [5/10] Installing curl-7.61.1...
[folsom] [5/10] Extracting curl-7.61.1: 100%
[folsom] [6/10] Installing pcre-8.42...
[folsom] [6/10] Extracting pcre-8.42: 100%
[folsom] [7/10] Installing gettext-runtime-0.19.8.1_1...
[folsom] [7/10] Extracting gettext-runtime-0.19.8.1_1: 100%
[folsom] [8/10] Installing vim-console-8.1.0342...
[folsom] [8/10] Extracting vim-console-8.1.0342: 100%
[folsom] [9/10] Installing git-lite-2.19.1...
===> Creating groups.
Creating group 'git_daemon' with gid '964'.
===> Creating users
Creating user 'git_daemon' with uid '964'.
[folsom] [9/10] Extracting git-lite-2.19.1: 100%
[folsom] [10/10] Installing zsh-5.6.2...
[folsom] [10/10] Extracting zsh-5.6.2: 100%
```

The PKG sub-command can, of course, do more than just `install`. The expectation is that you can fully leverage the pkg manager. This means, `install`, `update`, `upgrade`, `audit`, `clean`, `autoremove`, etc., etc.

```
ishmael ~ # bastille pkg ALL upgrade
Targeting all jails.

[bastion]:
Updating iniquity.io repository catalogue...
[bastion] Fetching meta.txz: 100%    560 B   0.6kB/s    00:01
[bastion] Fetching packagesite.txz: 100%  118 KiB 121.3kB/s    00:01
```

(continues on next page)

(continued from previous page)

```
Processing entries: 100%
iniquity.io repository update completed. 493 packages processed.
All repositories are up to date.
Checking for upgrades (1 candidates): 100%
Processing candidates (1 candidates): 100%
Checking integrity... done (0 conflicting)
Your packages are up to date.

[unbound0]:
Updating iniquity.io repository catalogue...
[unbound0] Fetching meta.txz: 100%    560 B   0.6kB/s    00:01
[unbound0] Fetching packagesite.txz: 100%  118 KiB 121.3kB/s    00:01
Processing entries: 100%
iniquity.io repository update completed. 493 packages processed.
All repositories are up to date.
Checking for upgrades (0 candidates): 100%
Processing candidates (0 candidates): 100%
Checking integrity... done (0 conflicting)
Your packages are up to date.

[unbound1]:
Updating iniquity.io repository catalogue...
[unbound1] Fetching meta.txz: 100%    560 B   0.6kB/s    00:01
[unbound1] Fetching packagesite.txz: 100%  118 KiB 121.3kB/s    00:01
Processing entries: 100%
iniquity.io repository update completed. 493 packages processed.
All repositories are up to date.
Checking for upgrades (0 candidates): 100%
Processing candidates (0 candidates): 100%
Checking integrity... done (0 conflicting)
Your packages are up to date.

[squid]:
Updating iniquity.io repository catalogue...
[squid] Fetching meta.txz: 100%    560 B   0.6kB/s    00:01
[squid] Fetching packagesite.txz: 100%  118 KiB 121.3kB/s    00:01
Processing entries: 100%
iniquity.io repository update completed. 493 packages processed.
All repositories are up to date.
Checking for upgrades (0 candidates): 100%
Processing candidates (0 candidates): 100%
Checking integrity... done (0 conflicting)
Your packages are up to date.

[nginx]:
Updating iniquity.io repository catalogue...
[nginx] Fetching meta.txz: 100%    560 B   0.6kB/s    00:01
[nginx] Fetching packagesite.txz: 100%  118 KiB 121.3kB/s    00:01
Processing entries: 100%
iniquity.io repository update completed. 493 packages processed.
All repositories are up to date.
Checking for upgrades (1 candidates): 100%
Processing candidates (1 candidates): 100%
The following 1 package(s) will be affected (of 0 checked):

Installed packages to be UPGRADED:
    nginx-lite: 1.14.0_14,2 -> 1.14.1,2
```

(continues on next page)

(continued from previous page)

```

Number of packages to be upgraded: 1

315 KiB to be downloaded.

Proceed with this action? [y/N]: y
[nginx] [1/1] Fetching nginx-lite-1.14.1,2.txz: 100%   315 KiB 322.8kB/s    00:01
Checking integrity... done (0 conflicting)
[nginx] [1/1] Upgrading nginx-lite from 1.14.0_14,2 to 1.14.1,2...
===> Creating groups.
Using existing group 'www'.
===> Creating users
Using existing user 'www'.
[nginx] [1/1] Extracting nginx-lite-1.14.1,2: 100%
You may need to manually remove /usr/local/etc/nginx/nginx.conf if it is no longer␣
↪needed.
```

CHAPTER 12

bastille destroy

Jails can be destroyed and thrown away just as easily as they were created. Note: jails must be stopped before destroyed.

```
ishmael ~ # bastille stop folsom
Targeting specified jails.
folsom

[folsom]:
folsom: removed

ishmael ~ # bastille destroy folsom
Deleting Jail: folsom.
Note: jail console logs not destroyed.
/usr/local/bastille/logs/folsom_console.log
```

CHAPTER 13

bastille template

Bastille supports a templating system allowing you to apply files, pkgs and execute commands inside the jail automatically.

Currently supported template hooks are: `PRE`, `CONFIG`, `PKG`, `SYSRC`, `CMD`. Planned template hooks include: `FSTAB`, `PF`

Templates are created in `${bastille_prefix}/templates` and can leverage any of the template hooks. Simply create a new directory named after the template. eg;

```
mkdir -p /usr/local/bastille/templates/base
```

To leverage a template hook, create an UPPERCASE file in the root of the template directory named after the hook you want to execute. eg;

```
echo "zsh vim-console git-lite htop" > /usr/local/bastille/templates/base/PKG
echo "/usr/bin/chsh -s /usr/local/bin/zsh" > /usr/local/bastille/templates/base/CMD
echo "etc root usr" > /usr/local/bastille/templates/base/CONFIG
```

Template hooks are executed in specific order and require specific syntax to work as expected. This table outlines those requirements:

| HOOK | format | example | |———|—————|———————————| | PRE/CMD | /bin/sh command | /usr/bin/chsh -s /usr/local/bin/zsh | | CONFIG | path | etc root usr | | PKG | port/pkg name(s) | vim-console zsh git-lite tree htop | | SYSRC | sysrc command(s) | nginx_enable=YES |

Note: SYSRC requires NO quotes or that quotes (`"`) be escaped. ie; `\"`)

In addition to supporting template hooks, Bastille supports overlaying files into the jail. This is done by placing the files in their full path, using the template directory as "/".

An example here may help. Think of `/usr/local/bastille/templates/base`, our example template, as the root of our filesystem overlay. If you create an `etc/hosts` or `etc/resolv.conf` *inside* the base template directory, these can be overlayed into your jail.

Note: due to the way FreeBSD segregates user-space, the majority of your overlayed template files will be in `usr/local`. The few general exceptions are the `etc/hosts`, `etc/resolv.conf`, and `etc/rc.conf.local`.

After populating `usr/local/` with custom config files that your jail will use, be sure to include `usr` in the template CONFIG definition. eg;

```
echo "etc usr" > /usr/local/bastille/templates/base/CONFIG
```

The above example "etc usr" will include anything under "etc" and "usr" inside the template. You do not need to list individual files. Just include the top-level directory name.

CHAPTER 14

Applying Templates

Jails must be running to apply templates.

Bastille includes a `template` sub-command. This sub-command requires a target and a template name. As covered in the previous section, template names correspond to directory names in the `bastille/templates` directory.

```
ishmael ~ # bastille template folsom base
[folsom]:
Copying files...
Copy complete.
Installing packages.
...[snip]...
Executing final command(s).
chsh: user information updated
Template Complete.
```

CHAPTER 15

bastille top

This one simply runs `top` in that jail. This command is interactive, as `top` is interactive.

CHAPTER 16

bastille htop

This one simply runs htop inside the jail. This one is a quick and dirty addition. note: won't work if you don't have htop installed in the jail.

CHAPTER 17

bastille sysrc

The `sysrc` sub-command allows for safely editing system configuration files. In jail terms, this allows us to toggle on/off services and options at startup.

```
ishmael ~ # bastille sysrc nginx nginx_enable="YES"
Targeting specified jails.
nginx

[nginx]:
nginx_enable: NO -> YES
```

See `man sysrc(8)` for more info.

CHAPTER 18

bastille console

This sub-command launches a login shell into the jail. Default is password-less root login.

```
ishmael ~ # bastille console folsom
Targeting specified jails.
folsom

[folsom]:
FreeBSD 11.2-RELEASE-p4 (GENERIC) #0: Thu Sep 27 08:16:24 UTC 2018

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@folsom:~ #
```

At this point you are logged in to the jail and have full shell access. The system is yours to use and/or abuse as you like. Any changes made inside the jail are limited to the jail.

CHAPTER 19

bastille cp

Note: this sub-command may need a little work.

This sub-command allows efficiently copying files from host to jail(s).

```
ishmael ~ # bastille cp ALL /tmp/resolv.conf-cf etc/resolv.conf
Targeting all jails.

[bastion]:

[unbound0]:

[unbound1]:

[squid]:

[nginx]:

[folsom]:
```

Unless you see errors reported in the output the `cp` was successful.

CHAPTER 20

bastille list

This sub-command will show you the running jails on your system.

```
ishmael ~ # bastille list
 JID             IP Address      Hostname                      Path
 bastion         10.88.9.65      bastion                       /usr/local/bastille/
↪jails/bastion/root
 unbound0        10.88.9.60      unbound0                      /usr/local/bastille/
↪jails/unbound0/root
 unbound1        10.88.9.61      unbound1                      /usr/local/bastille/
↪jails/unbound1/root
 squid           10.88.9.30      squid                         /usr/local/bastille/
↪jails/squid/root
 nginx           10.88.9.45      nginx                         /usr/local/bastille/
↪jails/nginx/root
 folsom          10.8.62.1       folsom                        /usr/local/bastille/
↪jails/folsom/root
```

CHAPTER 21

bastille update

The `update` command targets a release instead of a jail. Because every jail is based on a release, when the release is updated all the jails are automatically updated as well.

To update all jails based on the 11.2-RELEASE `release`:

Up to date 11.2-RELEASE:

```
ishmael ~ # bastille update 11.2-RELEASE
Targeting specified release.
11.2-RELEASE

Looking up update.FreeBSD.org mirrors... 2 mirrors found.
Fetching metadata signature for 11.2-RELEASE from update4.freebsd.org... done.
Fetching metadata index... done.
Inspecting system... done.
Preparing to download files... done.

No updates needed to update system to 11.2-RELEASE-p4.
No updates are available to install.
```

Updating 10.4-RELEASE:

```
ishmael ~ # bastille update 10.4-RELEASE
Targeting specified release.
10.4-RELEASE

Looking up update.FreeBSD.org mirrors... 2 mirrors found.
Fetching metadata signature for 10.4-RELEASE from update1.freebsd.org... done.
Fetching metadata index... done.
Fetching 2 metadata patches.. done.
Applying metadata patches... done.
Fetching 2 metadata files... done.
Inspecting system... done.
Preparing to download files... done.

```

(continues on next page)

(continued from previous page)

```
The following files will be added as part of updating to 10.4-RELEASE-p13:
...[snip]...
```

To be safe, you may want to restart any jails that have been updated live.

CHAPTER 22

bastille upgrade

This sub-command lets you upgrade a release to a new release. Depending on the workflow this can be similar to a `bootstrap`.

```
ishmael ~ # bastille upgrade 11.2-RELEASE 12.0-RELEASE
...
```

CHAPTER 23

bastille verify

This sub-command scans a bootstrapped release and validates that everything looks in order. This is not a 100% comprehensive check, but it compares the release against a "known good" index.

If you see errors or issues here, consider deleting and re-bootstrapping the release.

CHAPTER 24

Example (create, start, console)

This example creates, starts and consoles into the jail.

```
ishmael ~ # bastille create alcatraz 11.2-RELEASE 10.9.8.7

RELEASE: 11.2-RELEASE.
NAME: alcatraz.
IP: 10.9.8.7.
```

```
ishmael ~ # bastille start alcatraz
Targeting specified jails.
alcatraz

[alcatraz]:
alcatraz: created
```

```
ishmael ~ # bastille console alcatraz
Targeting specified jails.
alcatraz

[alcatraz]:
FreeBSD 11.2-RELEASE-p4 (GENERIC) #0: Thu Sep 27 08:16:24 UTC 2018

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
```

(continues on next page)

(continued from previous page)

```
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@alcatraz:~ #
```

```
root@alcatraz:~ # ps -auxw
USER   PID %CPU %MEM  VSZ  RSS TT  STAT STARTED    TIME COMMAND
root 83222  0.0  0.0 6412 2492  -  IsJ  02:21   0:00.00 /usr/sbin/syslogd -ss
root 88531  0.0  0.0 6464 2508  -  SsJ  02:21   0:00.01 /usr/sbin/cron -s
root  6587  0.0  0.0 6912 2788  3  R+J  02:42   0:00.00 ps -auxw
root 92441  0.0  0.0 6952 3024  3  IJ   02:21   0:00.00 login [pam] (login)
root 92565  0.0  0.0 7412 3756  3  SJ   02:21   0:00.01 -csh (csh)
root@alcatraz:~ #
```

CHAPTER 25

Project Goals

These tools are created initially with the mindset of function over form. I want to simply prove the concept is sound for real work. The real work is a sort of meta-jail-port system. Instead of installing the MySQL port directly on a system, you would use Bastille to install the MySQL port within a jail template built for MySQL. The same goes for DNS servers, and everything else in the ports tree.

Eventually I would like to have Bastille templates created for popular FreeBSD-based services. From Plex Media Servers to ad-blocking DNS resolvers. From tiny SSH jails to dynamic web servers.

I don't want to tell you what you can and can't run within this framework. There are no arbitrary limitations based on what I think may or may not be the best way to design systems. This is not my goal.

My goal is to provide a secure framework where processes and services can run isolated. I want to limit the scope and reach of bad actors. I want to severely limit the target areas available to anyone that has (or has gained!) access.

25.1 Possible Jail names

prisons:

- arkham
- ashecliffe
- azkaban
- coldmountain
- dolguldur
- foxriver
- litchfield
- oswald
- shawshank
- stockton

- stormcage
- ziggurat
- astralqueen
- alcatraz
- rikers
- leavenworth
- folsom
- attica
- singsing
- sanquentin
- corcoran
- pelicanbay

CHAPTER 26

Networking Tips

26.1 Tip #1:

Ports and destinations can be defined as lists. eg;

```
rdr pass inet proto tcp from any to any port {80, 443} -> {10.88.9.45, 10.88.9.46, 10.88.9.47, 10.88.9.48}
```

This rule would redirect any traffic to the host on ports 80 or 443 and round-robin between jails with ips 45, 46, 47, and 48 (on ports 80 or 443).

26.2 Tip #2:

Ports can redirect to other ports. eg;

```
rdr pass inet proto tcp from any to any port 8080 -> 10.7.6.5 port 80
rdr pass inet proto tcp from any to any port 8081 -> 10.7.6.5 port 8080
rdr pass inet proto tcp from any to any port 8181 -> 10.7.6.5 port 443
```

26.3 Tip #3:

Don’t worry too much about IP assignments. Initially I spent time worrying about what IP addresses to assign. In the end I’ve come to the conclusion that it *really* doesn’t matter. Pick *any* private address and be done with it. These are all isolated networks. In the end, what matters is you can map host:port to jail:port reliably, and we can.

CHAPTER 27

Community Support

We would love to hear your feedback on Bastille! Please join us on the BastilleBSD Chat Server and let us know what you think. Registration is currently open pending email verification.

Be mindful of the Bastille Code of Conduct when participating in the chat rooms.

If you've found a bug in Bastille, please submit it to the Bastille Issue Tracker.