Running the new FreeBSD jails

"A cheap substitute for doing it the right way"

James Gritton
<jamie@FreeBSD.org>

EuroBSDCon 2010
Karlsruhe, Germany

Simpler days

```
struct jail {
    uint32_t        version;
    char            *path;
    char            *hostname;
    uint32_t        ip_number;
};
```

But then...

```
struct jail {
    uint32_t        version;
    char            *path;
    char            *hostname;
    char            *jailname;
    uint32_t        ip4s;
    uint32_t        ip6s;
    struct in_addr  *ip4;
    struct in6_addr *ip6;
};
```

```
security.jail.enforce_statfs: 2
security.jail.mount_allowed: 0
security.jail.chflags_allowed: 0
security.jail.allow_raw_sockets: 0
security.jail.sysvipc_allowed: 0
security.jail.socket_unixiproute_only: 1
security.jail.set_hostname_allowed: 1
security.jail.jail_max_af_ips: 255
```

A new strategy

```
struct jail {
    uint32_t        version;
    char            *path;                  path=/usr/jail/foo
    char            *hostname;              host.hostname=foo.com
    char            *jailname;              name=foo
    uint32_t        ip4s;
    uint32_t        ip6s;
    struct in_addr  *ip4;                   ip4.addr=1.2.3.4,5.6.7.8
    struct in6_addr *ip6;                   ip6.addr=1234:5678::8765::4321
};

security.jail.enforce_statfs: 2             enforce.statfs=2
security.jail.mount_allowed: 0              allow.nomount
security.jail.chflags_allowed: 0            allow.nochflags
security.jail.allow_raw_sockets: 0          allow.noraw_sockets
security.jail.sysvipc_allowed: 0            allow.nosysvipc
security.jail.socket_unixiproute_only: 1    allow.nosocket_af
security.jail.set_hostname_allowed: 1       allow.set_hostname
security.jail.jail_max_af_ips: 255
```

A new strategy

```
struct jail {
    uint32_t        version;
    char            *path;                  path=/usr/jail/foo
    char            *hostname;              host.hostname=foo.com
    char            *jailname;              name=foo
    uint32_t        ip4s;
    uint32_t        ip6s;
    struct in_addr  *ip4;                   ip4.addr=1.2.3.4,5.6.7.8
    struct in6_addr *ip6;                   ip6.addr=1234:5678::8765::4321
};

security.jail.enforce_statfs: 2             enforce.statfs=2
security.jail.mount_allowed: 0              allow.nomount
security.jail.chflags_allowed: 0            allow.nochflags
security.jail.allow_raw_sockets: 0          allow.noraw_sockets
security.jail.sysvipc_allowed: 0            allow.nosysvipc
security.jail.socket_unixiproute_only: 1    allow.nosocket_af
security.jail.set_hostname_allowed: 1       allow.set_hostname
security.jail.jail_max_af_ips: 255          security.jail.jail_max_af_ips: 255
```

A new ABI

OUT:

```
struct jail j;
j.version = JAIL_API_VERSION;
j.path = "/usr/jail/foo";
j.jailname = "foo";
i.ip4s = 1;
j.ip6s = 0;
j.ip4 = &iaddr;
inet_aton("1.2.3.4", &iaddr);
jid = jail(&j);
```

IN:

```
struct iovec iov[6];
iov[0].iov_base = "name";
iov[0].iov_len = sizeof("name");
iov[1].iov_base = "foo";
iov[1].iov_len = sizeof("foo");
iov[2].iov_base = "path";
iov[2].iov_len = sizeof("path");
iov[3].iov_base = "/usr/jail/foo";
iov[3].iov_len = sizeof("/usr/jail/foo");
iov[4].iov_base = "ip4.addr";
iov[4].iov_len = sizeof("ip4.addr");
iov[5].iov_base = &iaddr;
iov[5].iov_len = sizeof iaddr;
inet_aton("1.2.3.4", &iaddr);
jid = jail_set(iov, 6,
    JAIL_CREATE | JAIL_ATTACH);
```

A new ABI

OUT:

```
struct jail j;
j.version = JAIL_API_VERSION;
j.path = "/usr/jail/foo";
j.jailname = "foo";
i.ip4s = 1;
j.ip6s = 0;
j.ip4 = &iaddr;
inet_aton("1.2.3.4", &iaddr);
jid = jail(&j);
```

IN:

```
struct jailparam params[3];
jailparam_init(&params[0], "name");
jailparam_init(&params[1], "path");
jailparam_init(&params[2], "ip4.addr");
jailparam_import(&params[0], "foo");
jailparam_import(&params[1], "/usr/jail/foo");
jailparam_import(&params[2], "1.2.3.4");
jid = jailparam_set(params, 3,
     JAIL_CREATE | JAIL_ATTACH);
jailparam_free(params);
```

A new ABI

OUT:

```
struct jail j;
j.version = JAIL_API_VERSION;
j.path = "/usr/jail/foo";
j.jailname = "foo";
i.ip4s = 1;
j.ip6s = 0;
j.ip4 = &iaddr;
inet_aton("1.2.3.4", &iaddr);
jid = jail(&j);
```

IN:

```
jid = jail_setv(JAIL_CREATE | JAIL_ATTACH,
    "name", "foo",
    "path", "/usr/jail/foo",
    "ip4.addr", "1.2.3.4", NULL);
```

Overview of jail parameters

`jid`	same as before
`name`	"
`path`	"
`host.hostname`	was just the hostname, now a few related things
`ip4.addr`	IP addresses, as before, with some tweaks
`ip6.addr`	
`securelevel`	This all used to be global sysctls, now per-jail controls
`enforce_statfs`	
`allow.this`	
`allow.that`	
`allow.the_other`	
`persist`	Jail can exist without processes
`vnet`	Virtual network stack

Functional groups

`ip4 = inherit`	**IPv4 isn't “jailed” - same addresses as host system**
`ip4 = new`	**Jail uses the IPv4 restriction feature**
`ip4.addr =1.2.3.4`	**The address is a detail of this feature**
`ip4 = disabled`	**Jail can't use IPv4 at all**

Extending...

```
linux = new
linux.osname = Linux
linux.osrelease = 2.6.16
linux.oss_version = 198144
```

A more flexible command line

OLD:

```
jail -n foo /usr/jail/foo foo.bar 1.2.3.4 /bin/csh
```

NEW:

```
jail -c name=foo path=/usr/jail/foo host.hostname=foo.bar \
    ip4.addr=1.2.3.4 command=/bin/csh
```

Examples

!

Give named a jail

```
--- contrib/bind9/bin/named/unix/os.c   2009-05-30 23:42:58.000000000 -0600
+++ contrib/bind9/bin/named/unix/os.c   2010-10-04 12:23:21.000000000 -0600
@@ -508,7 +511,9 @@
 #endif
        if (root != NULL) {
 #ifdef HAVE_CHROOT
-               if (chroot(root) < 0) {
+               if (jail_setv(JAIL_CREATE | JAIL_ATTACH, "host", "inherit",
+                   "ip4", "inherit", "ip6", "inherit", "path", root, NULL) < 0)
+               {
                        isc__strerror(errno, strbuf, sizeof(strbuf));
                        ns_main_earlyfatal("chroot(): %s", strbuf);
                }
```

Jails in jail

Jails in jail

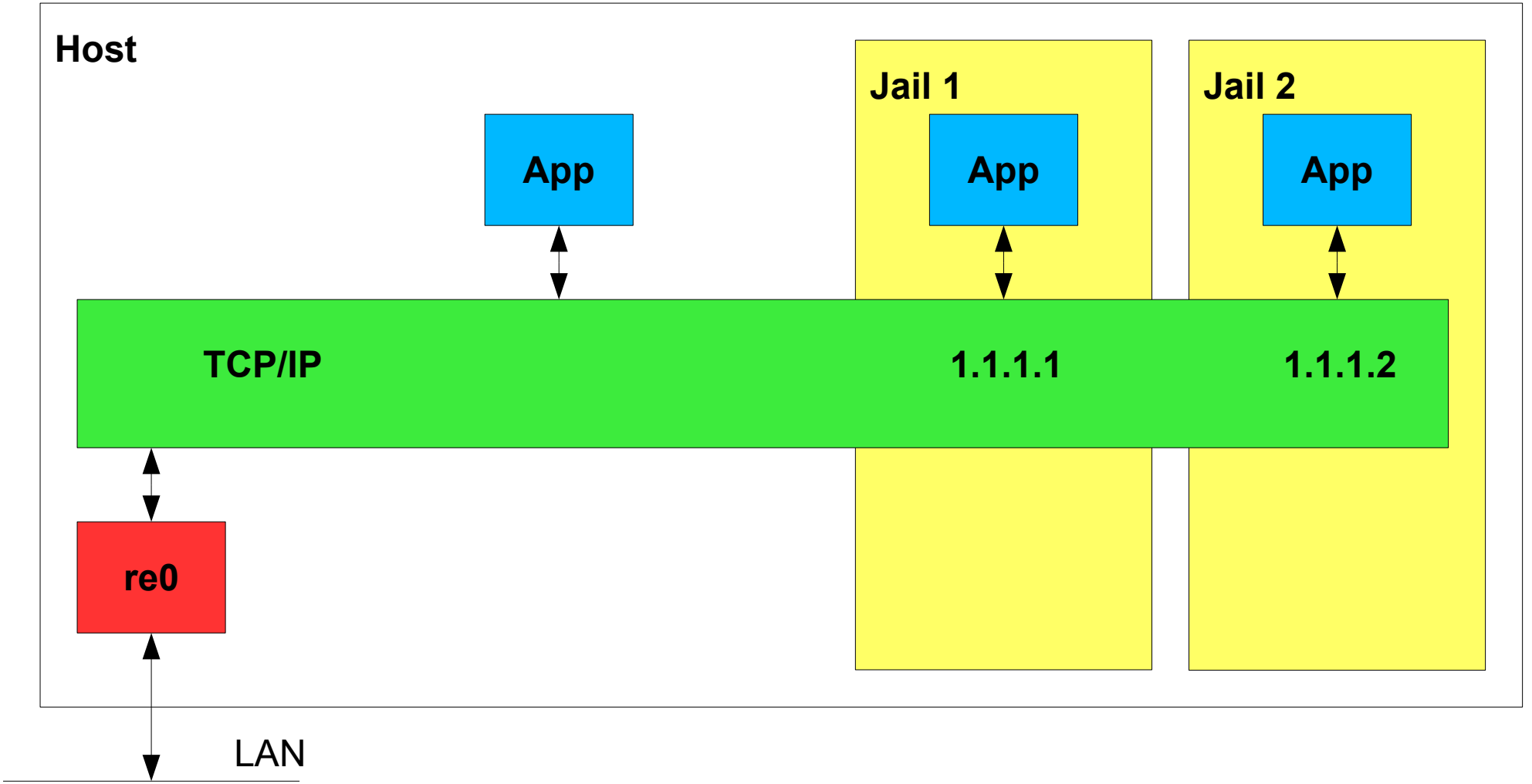
Coming soon...

Config files for jail(8)
http://people.FreeBSD.org/~jamie/jail.tbz
Expected in FreeBSD 9

More parameters!

`exec.start`	**Program(s) to run when starting a jail**
`exec.stop`	**Program(s) to run when stopping a jail**
`exec.other_stuff`	
`mount`	**A filesystem to mount, perhaps loopback or union**
`mount.devfs`	**Mount jail's /dev, run devfs(8) on it**
`interface`	**Configure the jails IP addresses here**

Simple VPS jail



Simple VPS jail

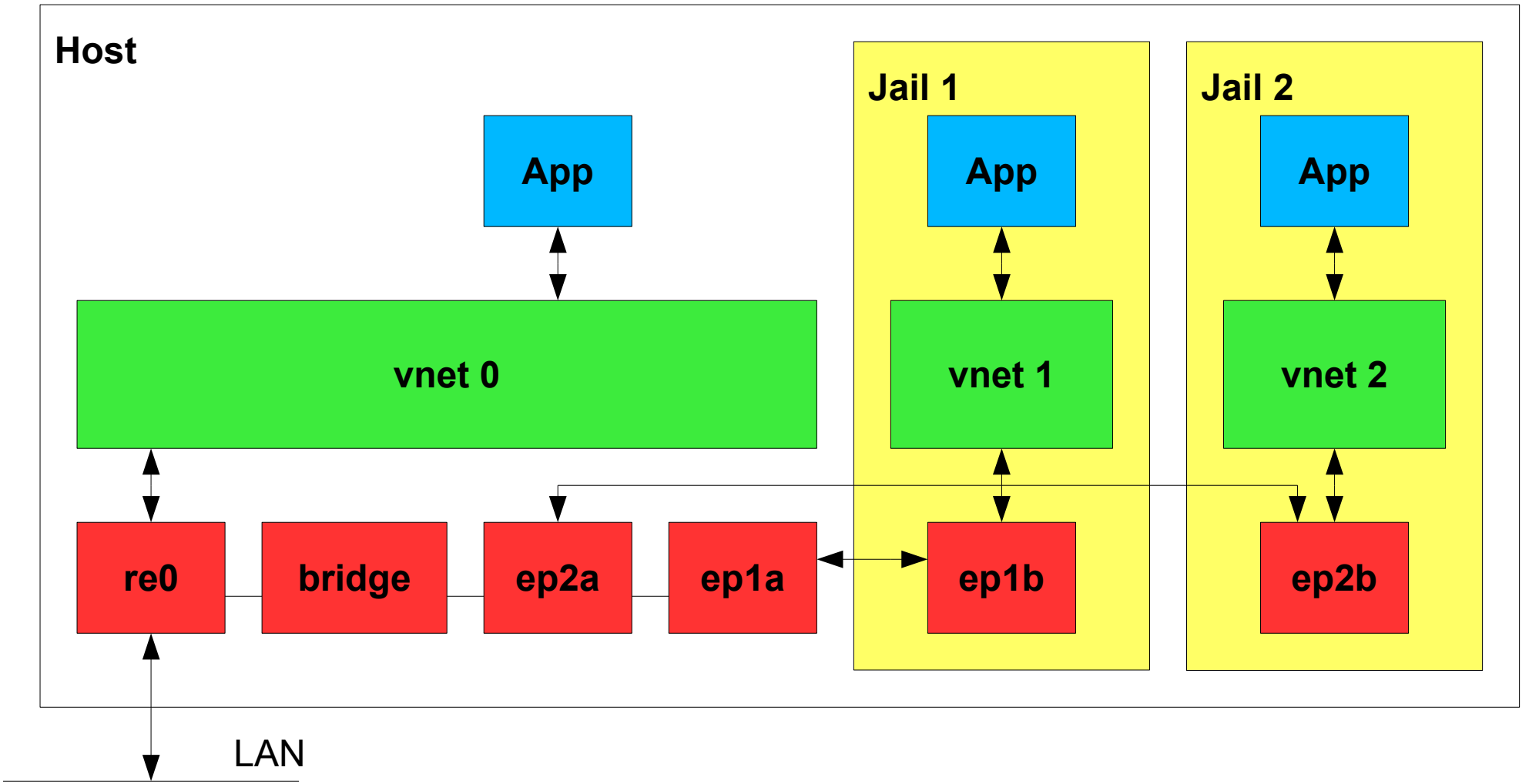
```
* {
    exec.start = "/bin/sh /etc/rc";
    exec.stop  = "/bin/sh /etc/rc.shutdown";
    exec.consolelog = /dev/null;
    exec.clean;
    mount.devfs;
    interface = msk0;

    path = /usr/jail/$name;
}

merry {
    host.hostname = merry.gritton.org;
    ip4.addr = 1.1.1.1;
}

pippin {
    host.hostname = pippin.gritton.org;
    ip4.addr = 1.1.1.2;
}
```

Vnet VPS jail



Vnet VPS jail

```
exec.prestart  = "ifconfig epair$if create up >/dev/null";
exec.prestart += "ifconfig bridge0 addm epair${if}a";
exec.start     = "ifconfig lo0 up 127.1";
exec.start    += "ifconfig epair${if}b up $ipaddr";
exec.start    += "sh /etc/rc";
exec.stop      = "sh /etc/rc.shutdown";
exec.poststop  = "ifconfig epair${if}a destroy";

vnet;
vnet.interface = "epair${if}b";

merry {
    $if = 1;
    $ipaddr = 1.1.1.1;
}

pippin {
    $if = 2;
    $ipaddr = 1.1.1.2;
}
```

```
"Running the new FreeBSD jails" {
    author = "Jamie Gritton";
    email = <jamie@FreeBSD.org>;
    url =
        "http://people.FreeBSD.org/~jamie/jail.tbz";
    location =
        "EuroBSDCon 2010, Karlsruhe, Germany";
    questions = "?";
}
```