Building a FreeBSD based Virtual Appliance

How we built the Razorback appliance

VRT

SOURCEfire®

Introduction

VRT

SOURCEfire®

About Your Presenter

Currently

- Senior Research Engineer with the Vulnerability Research Team at Sourcefire Inc.
  - Working on Razorback
    http://razorbacktm.sourceforge.net/

Previously

- Senior Network Architect with Mintel International Ltd.
  - Responsible for maintaining ~200 physical FreeBSD systems and 400+ Jails spread across 7 sites
  - Providing tools for the system administration team to perform their day to day duties.

What is Razorback?

Razorback is…

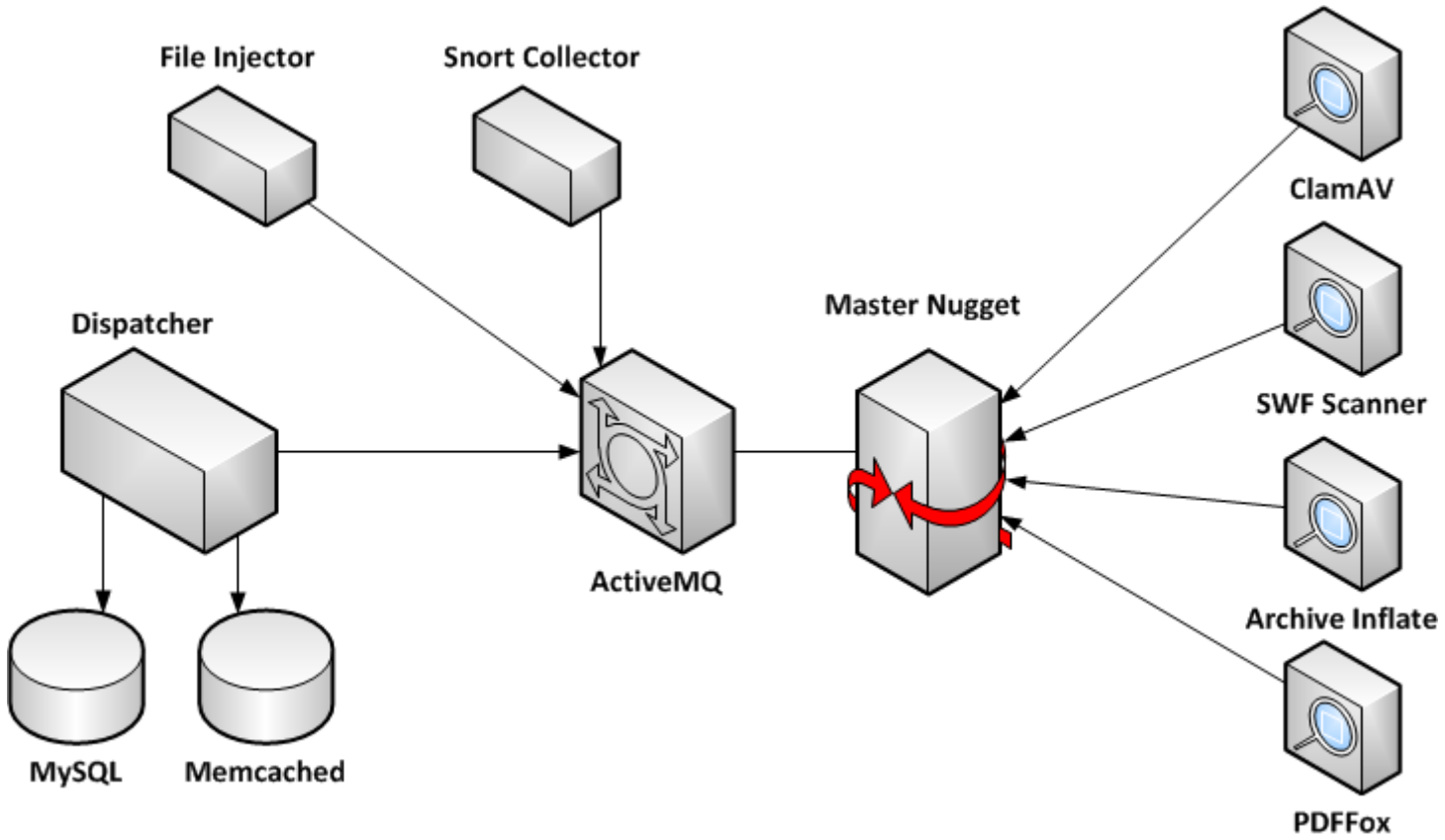
- An Open Source security framework
- An advanced data inspection system
- A data capture and distribution system
- An event correlation and alerting system
- Easy to extend with new detection

- Our answer the the evolving threat landscape

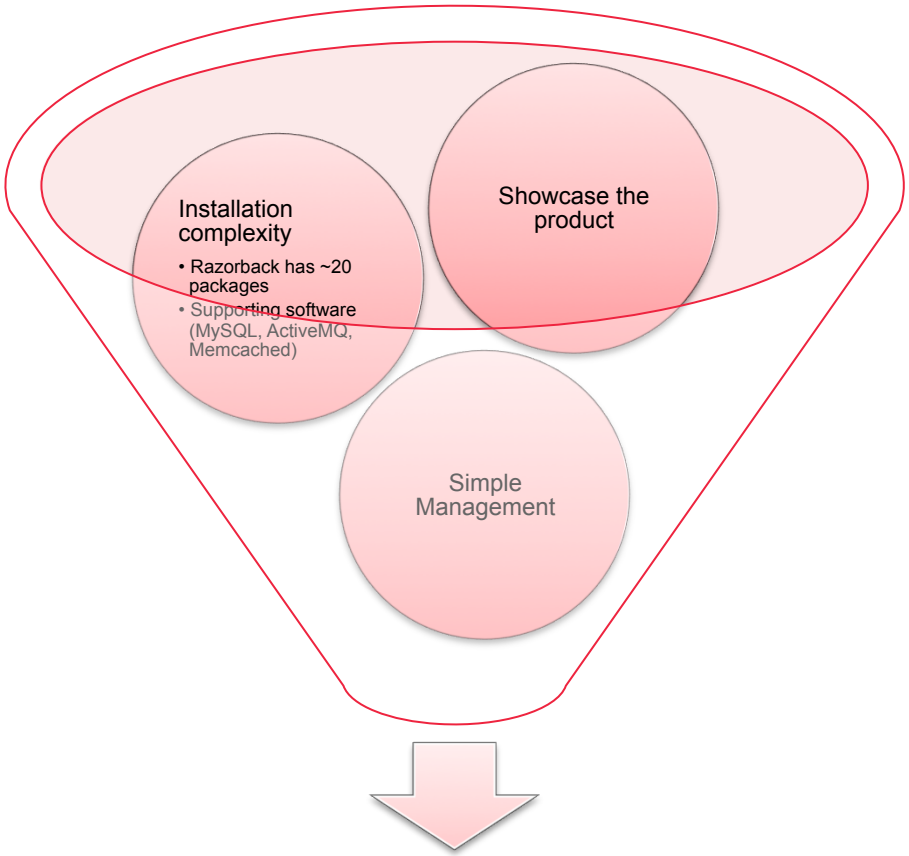
VRT

SOURCEfire®

Razorback Overview



Why did we build an appliance?



Improved adoption

Why FreeBSD?

- Experience
- Supported by Razorback
- Secure
- Familiarity with the ports system

- The alternatives:
  - VMWare Appliance Builder
    - Huge scary wedge of documentation
    - Unfamiliar with linux packaging systems
    - Nothing to give back

Appliance Overview

System Components

- System Management Interface
  - Network
  - Services
  - Users
- Razorback Management Interface
  - Nugget Configuration
  - Nugget Control
- Razorback Core
  - Analyst Interface
  - Backend Services

VRT

SOURCEfire®

The Solutions – System Management

- FreeNAS Management Interface
  - Extensible
  - Simple
  - Python + Dojo
  - Works with NanoBSD (future work)
- Webmin
  - Extensible
  - Over reaches requirements
  - Perl

Extracting the FreeNAS interface

- Started with a svn snapshot r10153
- Reworked the backend
  - Renamed vendor scripts to fadmin
  - Remove assumptions of a NanoBSD based system
- Reworked the frontend
  - Configurable branding
  - Configurable applications and services
- Up on SourceForge as the ‘freebsdadmin’ project: http://sf.net/projects/freebsdadmin/

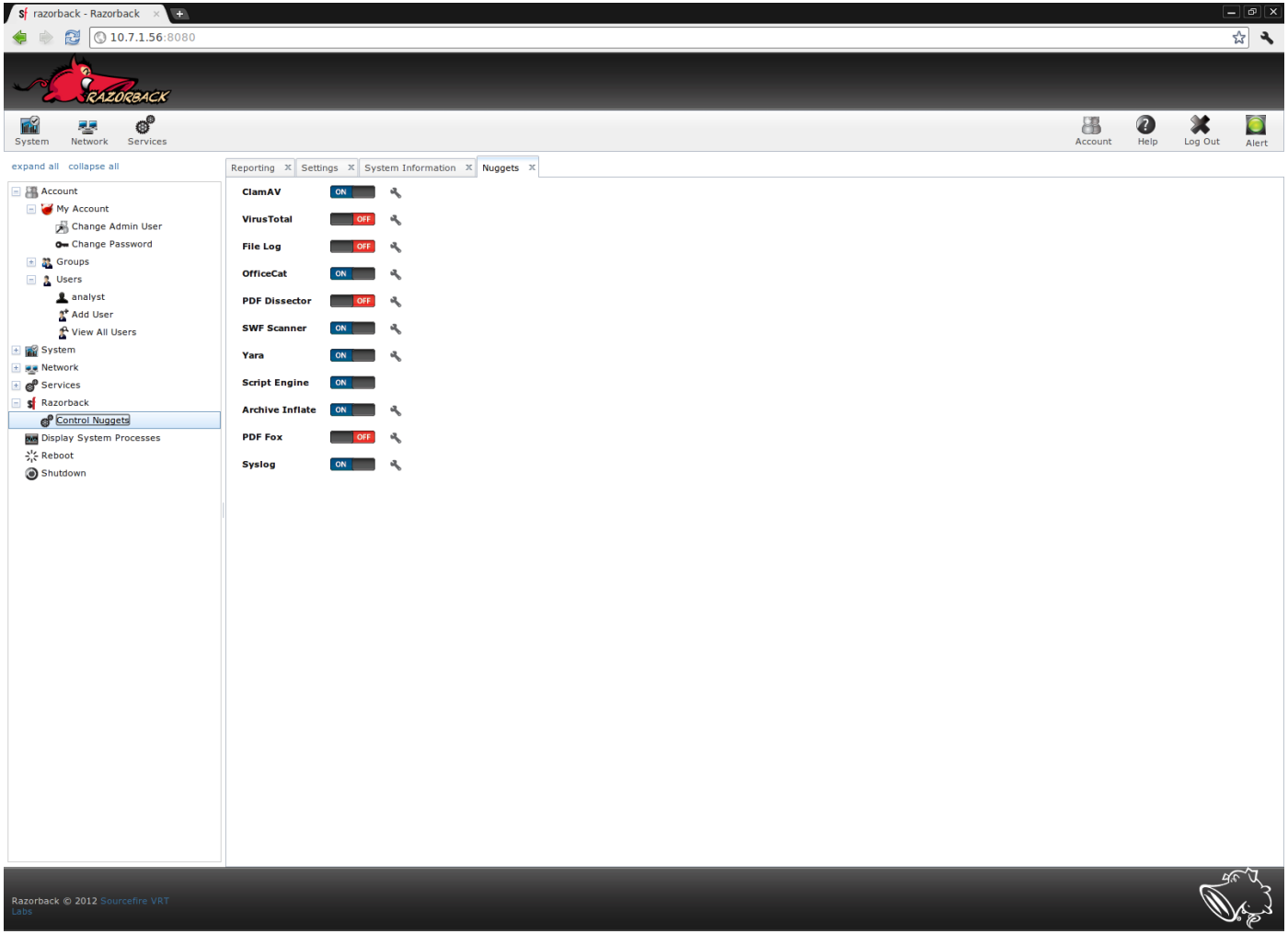
FreeBSD Admin – Customization

- New Services
  - MySQL
  - ActiveMQ
  - Razorback Dispatcher
  - Razorback Master Nugget
  - ClamAV clamd
- Customized Branding
- New Razoback application
  - Nugget configuration
  - Backend scripts

VRT

SOURCEfire®

End Result

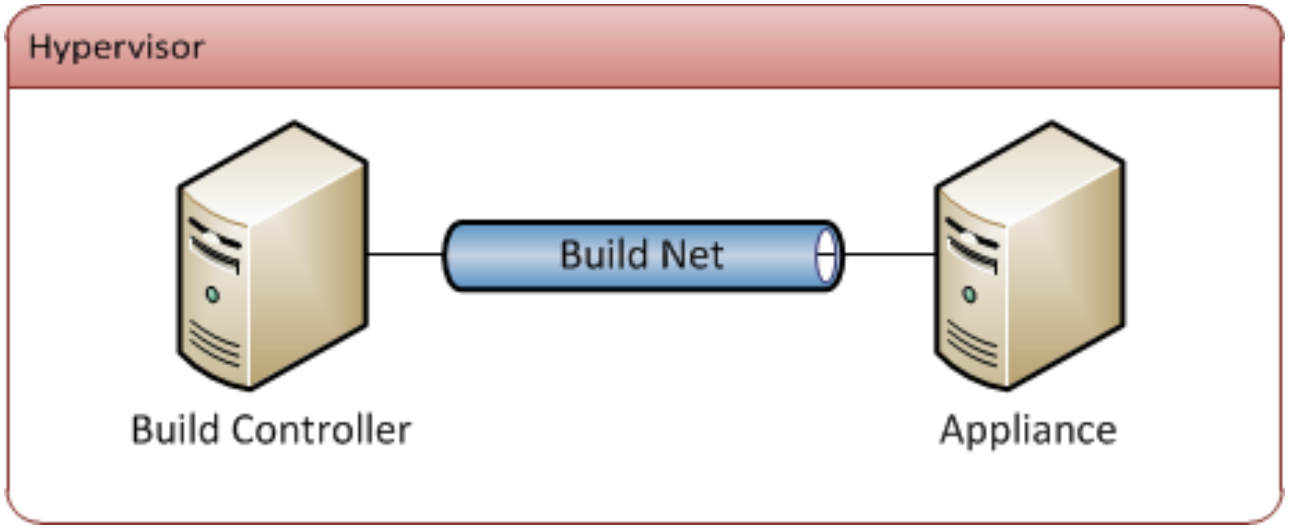

razorback - Razorback
10.7.1.56:8080
RAZORBACK
System
Network
Services
Account
Help
Log Out
Alert
expand all
collapse all
Account
My Account
Change Admin User
Change Password
Groups
Users
analyst
Add User
View All Users
System
Network
Services
Razorback
Control Nuggets
Display System Processes
Reboot
Shutdown
Reporting
Settings
System Information
Nuggets
ClamAV ON
VirusTotal OFF
File Log OFF
OfficeCat ON
PDF Dissector OFF
SWF Scanner ON
Yara ON
Script Engine ON
Archive Inflate ON
PDF Fox OFF
Syslog ON
Razorback © 2012 Sourcefire VRT
Labs

Build System

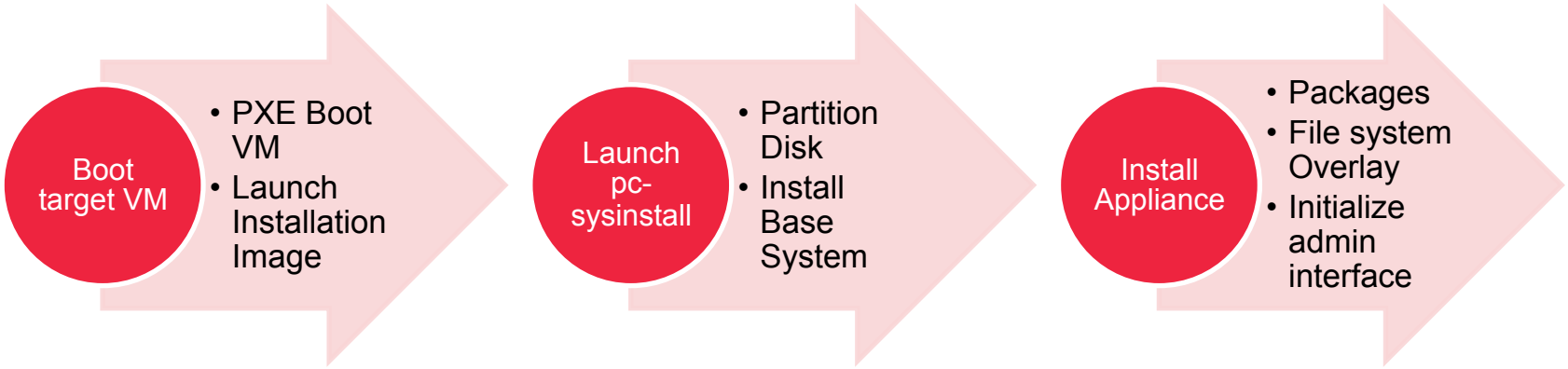
Build Overview

- Dark Ages
  - Single VM with snapshots
  - Hand applied updates
  - FreeBSD 8.1 Based
- Now
  - PXE Installation Environment
  - Fully automated build
  - FreeBSD 9.0 Based
- Future Work
  - Hypervisor Integration
  - Web Interface

Network Layout



Build Process



Build Controller - Overview

- NFS – Installation Image Root
- DNS
- DHCP – PXE
- TFTP – PXE
- Tinderbox – Package Building
- FTP – Package Installation

Build Controller – Required Packages

- FreeBSD 9.0 Base Install
- net/isc-dhcp42-server
- net/rsync
- databases/mysql55-server
- ports-mgmt/tinderbox-devel
- www/apache22
- lang/php5 (With apache module)
- ftp/lftp
- devel/subversion

Build Controller – Network Configuration

- Hostname – master.install.local
- Interface 0 – LAN connection (internet access)
- Interface 1 – Build LAN

/etc/rc.conf:

```
hostname=“master.install.local”
ifconfig_em0=“DHCP”
ifconfig_em1=“inet 172.17.0.1/24”
```

DHCP Client Configuration

/etc/dhclient.conf:

```
supersede domain-name-servers 127.0.0.1;
supersede domain-name "install.local";
```

Build Controller – DNS Server

Forward Zone – install.local.

/etc/namedb/master/install.db:

```
$TTL 3h
install.local. SOA install.local. nobody.install.local. 42 1d 12h 1w 3h
                NS  master.install.local.
master          A   172.17.0.1
10              A   172.17.0.10
11              A   172.17.0.11
```

Reverse Zone – 0.17.172.in-addr.arpa.

/etc/namedb/master/install.rev:

```
$TTL 3h
0.17.172.in-addr.arpa. SOA 0.17.172.in-addr.arpa. nobody.install.local. 42 1d 12h 1w 3h
                NS    master.install.local.
1               PTR   master.install.local.
10              PTR   10.install.local.
11              PTR   11.install.local.
```

Build Controller – DNS Server Cont.

Add the zones to the end of /etc/namedb/named.conf

```
zone "install.local" {
    type master;
    file "/etc/namedb/master/install.db”;
};
zone "0.17.172.in-addr.arpa" {
    type master;
    file "/etc/namedb/master/install.rev”;
};
```

Update the listen directive:

```
listen-on       {
        127.0.0.1;
        172.17.0.1;
};
```

Build Controller – DHCP Server

Installation network DHCP configuration

/usr/local/etc/dhcpd.conf:

```
option domain-name "install.local”;
option domain-name-servers master.install.local;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
authoritative;
log-facility local7;

filename "pxeboot”;
option root-path "172.17.0.1:/install/nfs”;
server-name "master.install.local”;
server-identifier 172.17.0.1;

subnet 172.17.0.0 netmask 255.255.255.0 {
  range 172.17.0.10 172.17.0.11;
  next-server 172.17.0.1;
  option broadcast-address 172.17.0.255;
  option routers master.install.local;
}
```

Build Controller – File Servers

Add anonymous FTP user

```
pw user add ftp -d /install
```

Setup NFS exports

/etc/exports:

```
/install -alldirs -maproot=0:0 -network 172.17.0.0/16
```

Enable ftp and tftp in inetd

/etc/inetd.conf:

```
tftp            dgram   udp wait        root    /usr/libexec/tftpd  tftpd -l -s /install/tftp
ftp             stream  tcp   nowait    root     /usr/libexec/ftpd    ftpd -l -A
```

Build Controller – Services

Enable services in /etc/rc.conf

```
named_enable="YES”
dhcpd_enable="YES”
dhcpd_ifaces="em1”
inetd_enable="YES”
nfs_server_enable="YES”
rpcbind_enable="YES”
rpc_statd_enable="YES”
rpc_lockd_enable="YES”
mountd_enable="YES"
```

Build Controller – Install Image

Mount the 9.0 Release CD

CD Drive:

```
mount -t cd9660 /dev/cd0 /mnt
```

ISO Image:

```
MDDEV=`mdconfig -a -t vnode -f FreeBSD-9.0-RELEASE-i386-diskc1.iso`
mount -t cd9660 /dev/${MDDEV} /mnt
```

Extract the install image

```
cd /mnt
mkdir /install
mkdir /install/nfs
mkdir /install/tftp
rsync -av . /install/nfs
```

Build Controller – Install Image Cont.

Unmount the CD

```
cd /
umount /mnt
```

ISO Image:

```
mdconfig -d -u ${MDDEV}
```

Install the PXE boot loader

```
cp  /install/nfs/boot/pxeboot /install/tftp/
```

Build Controller – Install Image Cont. (2)

Prepare the installation file for pc-sysinstall:

```
cd /install/nfs/usr/freebsd-dist
cp base.txz image.txz
unxz image.txz
unxz kernel.txz
tar -rf image.tar @kernel.tar
xz kernel.tar
xz image.tar
```

Clean image fstab:

```
echo "tmpfs /tmp tmpfs rw,mode=777 0 0" > /install/nfs/etc/fstab
```

Enable diskless mode:

```
echo "diskless_enable=\"YES\"" >> /install/nfs/etc/rc.conf
echo "tmpfs_enable"YES\"" >> /install/nfs/etc/rc.conf
```

Start installer at boot:

```
echo "pc-sysinstall start-autoinstall /boot/autoinstall.conf" > /install/nfs/etc/rc.local
```

Build Controller – Setup Tinderbox

Enable services in rc.conf:

```
mysql_enable="YES”
apache22_enable="YES”
tinderd_enable="YES”
tinderd_directory="/usr/local/tinderbox/scripts"
```

Secure and start MySQL:

```
mysql_secure_installation
service mysql-server start
```

Setup PHP:

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
echo “date.timezone = America/New_York” >> /usr/local/etc/php.ini
```

Update the following files appropriately:

webui/inc_tinderbox.php
webui/inc_ds.php

Build Controller – Setup Tinderbox Cont.

Setup tinderbox:

```
cd /usr/local/tinderbox/scripts
./tc Setup
echo "/usr/local/tinderbox -alldirs -maproot=0:0 localhost" >> /etc/exports
killall -HUP mountd
cp webui/inc_ds.php.dist webui/inc_ds.php
cp webui/inc_tinderbox.php.dist webui/inc_tinderbox.php
mkdir /usr/local/tinderbox/options
./tc configOptions –e –d /options
```

Update the following files appropriately:

- /usr/local/tinderbox/scripts/webui/inc_tinderbox.php
- /usr/local/tinderbox/scripts/webui/inc_ds.php

Build Controller – Setup Tinderbox Cont.

Configure apache, append to:
/usr/local/etc/apache22/httpd.conf:

```
AddType application/x-httpd-php .php
DirectoryIndex index.html index.php
RewriteEngine on
RewriteRule   ^/$  /tb/  [R]
Alias /tb/logs/ "/usr/local/tinderbox/logs/”
Alias /tb/packages/ "/usr/local/tinderbox/packages/”
Alias /tb/errors/ "/usr/local/tinderbox/errors/”
Alias /tb/wrkdirs/ "/usr/local/tinderbox/wrkdirs/”
Alias /tb/ "/usr/local/tinderbox/scripts/webui/"

<Directory "/usr/local/tinderbox/”>
        Order allow,deny
        Allow from all
</Directory>
```

Start the service:

```
service apache22 start
```

Build Controller – Create Package Build

Setup build environment:

/usr/local/tinderbox/scripts/etc/env/9.0-i386:

```
export ARCH=i386
export MACHINE_ARCH=i386
export UNAME_m=i386
export UNAME_p=i386
```

Create build in tinderbox:

```
cd /usr/local/tinderbox/scripts
./tc createJail -j 9.0-i386 -d "FreeBSD 9.0-RELEASE (i386)" -t 9.0-RELEASE -u LFTP -H ftp.freebsd.org -a i386
./tc createPortsTree -p FreeBSD -d "FreeBSD ports tree" -w http://www.freebsd.org/cgi/cvsweb.cgi/ports/
./tc createBuild -b 9.0-FreeBSD-i386 -j 9.0-i386 -p FreeBSD -d "9.0-RELEASE (i386)"
```

Start the builder:

```
service tinderbox start
```

Build Controller – Setup FreeBSD Admin

Checkout the code:

```
cd /install/nfs
svn checkout svn://svn.code.sf.net/p/freebsdadmin/code/trunk freebsdadmin
```

Build the required packages in tinderbox:

```
/install/nfs/freebsdadmin/bin/build-pkgs.sh -b 9.0-FreeBSD-i386
```

Generate the package list:

```
/install/nfs/freebsdadmin/bin/gen-pkg-list.sh -b 9.0-FreeBSD-i386
```

Fix a bug in pc-sysinstall:

```
/install/nfs/freebsdadmin/bin/fix-autoinstall.sh
```

Install rsync in the install image:

```
mkdir -p /install/nfs/usr/local/bin
cp /usr/local/bin/rsync /install/nfs/usr/local/bin/
```

Create an appliance

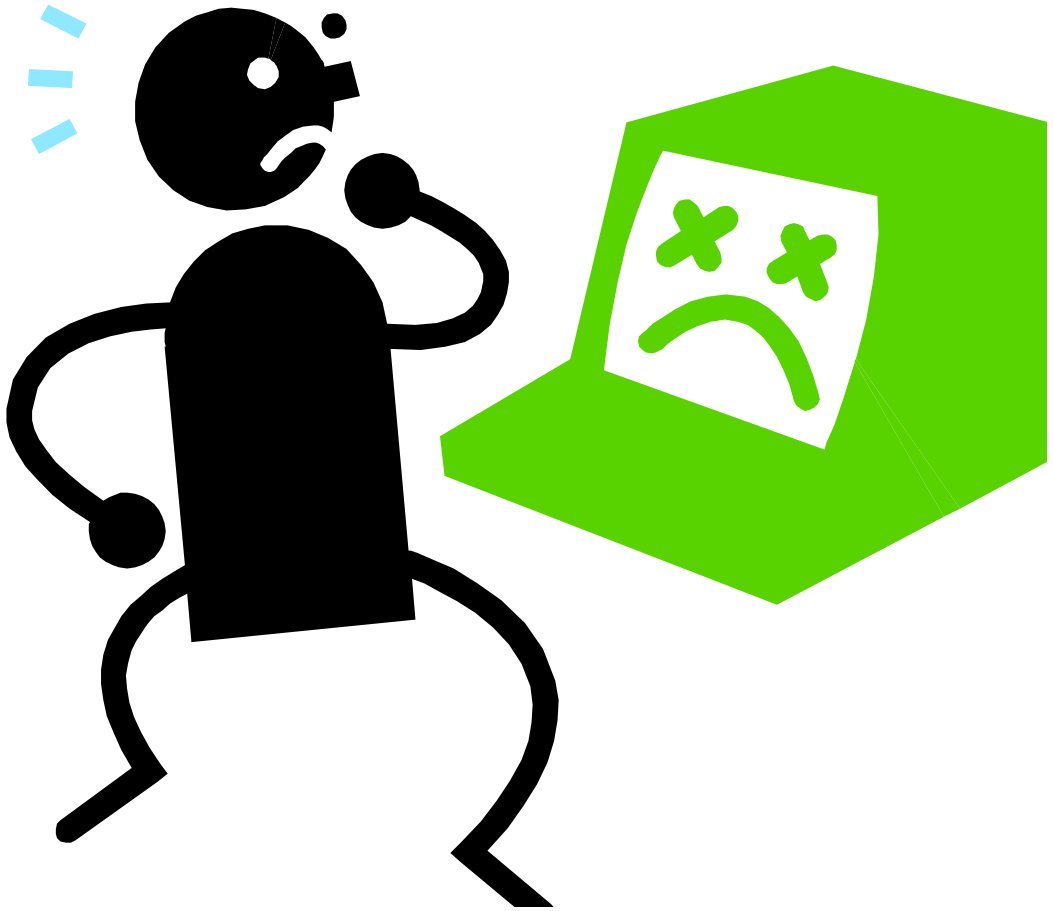
- VM Details:
  - Primary NIC in appliance build virtual network.
  - SCSI disk controller
  - PXE Enabled
- Boot the vm
- Export OVA

Live Demo

VRT

SOURCEfire®

Did it work?



The Important Bits

Information

Projects:

- FreeBSD Admin - http://sf.net/projects/freebsdadmin/
- Razorback™ - http://razorbacktm.sourceforge.net/

Contact:

- Email: tjudge@sourcefire.com, tom@tomjudge.com
- IRC: t_j on FreeNode and EFNet
- Twitter: @amishHatchet

VRT:

- Email: research@sourcefire.com
- Twitter: @Sourcefire_VRT
- Blog: http://vrt-blog.snort.org/
- Web: http://labs.snort.org/

Questions