

MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

PROTECTING DYNAMIC WEBSITES IN FREEBSD

INSIDE

UNBOUND ON FREEBSD 8.2

KEEPING UP TO DATE IN PC-BSD 9

USING LIFE PRESERVER TO BACKUP A PC-BSD 9.0 SYSTEM TO FREENAS™ 8.0.1

DRAGONFLYBSD: RECOVERING DATA WITH HAMMER

INSTALLING APACHE, PHP, MYSQL AND MODSECURITY IN FREEBSD 8.2

MYSQL UNLEASHED!

TERMINAL DESCRIPTIONS FOR OPENBSD AMD/INTEL CONSOLES

(AB)USING VIDEOLAN

NETBSD INTRUSION DETECTION SERVER

VOL.4 NO.9
ISSUE 09/2011(26)
1898-9144



800-820-BSDI
<http://www.ixsystems.com>
Enterprise Servers for Open Source



✓ Increased Performance ✓ Impressive Energy Savings

TrueNAS™ Pro Storage Appliance: You are the Cloud

With a rock-solid FreeBSD® base, Zettabyte File System support, and a powerful Web GUI, TrueNAS™ Pro pairs easy-to-manage software with world-class hardware for an unbeatable storage solution.



*Expansion
Shelves
Available*



TrueNAS™ 2U Pro System



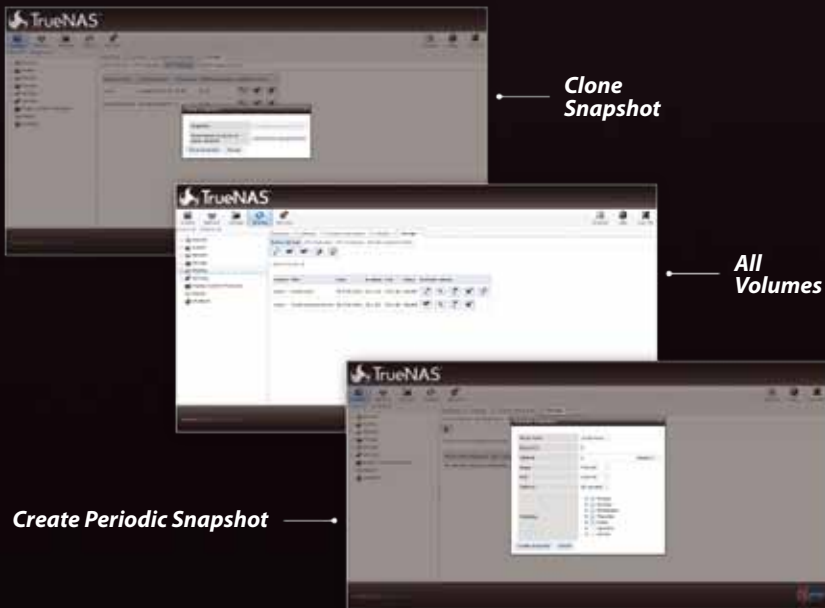
TrueNAS™ 4U Pro System



Storage. Speed. Stability.

In order to achieve maximum performance, the TrueNAS™ Pro 2U and 4U Systems, equipped with the Intel® Xeon® Processor 5600 Series, support Fusion-io's Flash Memory cards and 10GbE Network Cards. Titan TrueNAS™ Pro 2U and 4U Appliances are an excellent storage solution for video streaming, file hosting, virtualization, and more. Paired with optional JBOD expansion units, the TrueNAS™ Pro Systems offer excellent capacity at an affordable price.

For more information on the **TrueNAS™ 2U Pro** and **TrueNAS™ 4U Pro**, or to request a quote, visit: <http://www.ixsystems.com/TrueNAS>.



TrueNAS™ 2U PRO KEY FEATURES

- Supports One or Two Quad-Core or Six-Core, Intel® Xeon® Processor 5600 Series
- 12 Hot-Swap Drive Bays - Up to 36TB of Data Storage Capacity*
- Periodic Snapshots Feature Allows You to Restore Data from a Previously Generated Snapshot
- Remote Replication Allows You to Copy a Snapshot to an Offsite Server, for Maximum Data Security
- Up to 4.48TB of Fusion-io Flash Memory
- 2 x 1GbE Network Interface (Onboard) + Up to 4 Additional 1GbE Ports or Single/Dual Port 10GbE Network Cards

TrueNAS™ 4U PRO KEY FEATURES

- Supports One or Two Quad-Core or Six-Core, Intel® Xeon® Processor 5600 Series
- 24 or 36 Hot-Swap Drive Bays - Up to 108TB of Data Storage Capacity*
- Periodic Snapshots Feature Allows You to Restore Data from a Previously Generated Snapshot
- Remote Replication Allows You to Copy a Snapshot to an Offsite Server, for Maximum Data Security
- Up to 14.08TB of Fusion-io Flash Memory
- 2 x 1GbE Network Interface (Onboard) + Up to 4 Additional 1GbE Ports or Single/Dual Port 10GbE Network Cards

JBOD expansion is available on the 2U and 4U Pro Systems

** 2.5" drive options available; please consult with your Account Manager*



Call iXsystems toll free or visit our website today!

1-855-GREP-4-IX | www.ixsystems.com



Dear Readers,

I'm presenting you with the newest issue of BSD magazine: Protecting dynamic websites in FreeBSD.

We warm up with Darrel Levitch article about insalling and configuring DNSSEC for small networks using Unbound.

Then we move on to the Developers Corner, which is very PC-BSD oriented this month – with two articles written by Kris Moore and Dru Lavigne. You will learn how to easily update your PC-BSD and how to backup it to FreeNAS with Life Preserver.

We also couldn't miss news from DragonflyBSD project – provided by Justin Sherrill.

How Tos first article is our cover story written by Stavros Shaeles – his tutorial will guide us step by step and show how to install and configure various applications to successfully protect our dynamic websites from various attacks.

It is followed by Sufyan bin Uzayr and his article explaining how to tune and optimize MySQL databases for best performance, and Alexei Malinin who describes his work with OpenBSD consoles for AMD/Intel PC's.

After that Michael Bushkov will show us some tricks of how we can use our video and audio using VideoLAN command line interface.

In the last article, written by Svetoslav Chukov, we will take a look at NetBSD Intrusion Detection Server.

I hope you will find this issue to be both interesting and educating. Remember we always await your feedback so don't hesitate to mail us if you have any questions or suggestions :)

Yours,

*Zbigniew Puchciński
Editor in Chief*

zbigniew.puchcinski@software.com.pl

MAGAZINE BSD

Editor in Chief:

Zbigniew Puchciński
zbigniew.puchcinski@software.com.pl

Contributing:

Darrel Levitch, Kris Moore, Dru Lavigne, Justin C. Sherrill,
Stavros N. Shaeles, Sufyan bin Uzayr, Alexei Malinin,
Michael Bushkov, Svetoslav Chukov

Proofreaders:

Sander Reiche, Tristan Karstens

Special Thanks:

Denise Ebery

Art Director:

Ireneusz Pogroszewski

DTP:

Ireneusz Pogroszewski

Senior Consultant/Publisher:

Paweł Marciniak pawel@software.com.pl

CEO:

Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director:

Andrzej Kuca
andrzej.kuca@software.com.pl

Executive Ad Consultant:

Ewa Dudzic
ewa.dudzic@software.com.pl

Advertising Sales:

Zbigniew Puchciński
zbigniew.puchcinski@software.com.pl

Publisher :

Software Press Sp. z o.o. SK
ul. Bokszerska 1, 02-682 Warszawa
Poland

worldwide publishing

tel: 1 917 338 36 31

www.bsdmag.org

Software Press Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org

All trade marks presented in the magazine were used only for informative purposes. All rights to trade marks presented in the magazine are reserved by the companies which own them.

The editors use automatic DTP system **AOPDS**

Mathematical formulas created by Design Science MathType™.

Get Started

06 DNSSEC resolution and IPv6 Unbound on FreeBSD 8.2

Darrel Levitch

Unbound runs on FreeBSD, OpenBSD, NetBSD, Linux, and Microsoft Windows. It provides a reasonably simple way to implement DNSSEC in a local-area network. With Unbound forward and reverse resolution is possible for small networks where IPv6 is implemented

Developers Corner

08 Keeping up to date in PC-BSD 9

Kris Moore

Since the early days of PC-BSD, there has been various GUI mechanisms for performing critical system and security updates.

10 Using Life Preserver to Backup a PC-BSD 9.0 System to FreeNAS™ 8.0.1

Dru Lavigne

This article demonstrates how to use the built-in Life Preserver program to backup a PC-BSD 9.0 desktop system to a FreeNAS™ 8.0.1 NAS system. Users can refer to the Guides at http://wiki.pcbbsd.org/index.php/PC-BSD_9_Handbook and <http://doc.freenas.org> for instructions on how to install PC-BSD and FreeNAS™.

16 Recovering data with hammer

Justin C. Sherrill

We've all experienced instant regret. That's the feeling that comes within a second of executing a command like „rm -rf * .txt” (note the space) or of cutting the wrong cluster of wires at the end of a long conduit. Not that I am quoting from experience, or anything like that, no...

How Tos

18 Apache2, php5, mysql5, modsecurity2.5 installation and configuration in order to protect dynamic websites from various attacks, in Freebsd 8.2

Stavros N. Shaeles

In the last years there is a tremendous increment in dynamic website and cms using php. A very large piece

of the market of this websites are served by Apache Webserver using Mysql as database basically in Unix systems. Also this tremendous increment of php in dynamic website and opensource cms like joomla increase and hackers attacks in order to compromise a website or hack the server to use it in botnet. So someone can wonder, is there anything that can protect my websites except from backups and upgrading our system and software? The answer is yes.

28 MySQL Unleashed!

Sufyan bin Uzayr

We explore some tips and tricks that you can use to gain better performance with MySQL

34 Terminal Descriptions for OpenBSD AMD/Intel consoles

Alexei Malinin

In this article I would like to describe the results of my work of tuning OpenBSD consoles for AMD/Intel PCs. These results are also applicable to computers with the same hardware architecture (amd64 or i386, see <http://www.openbsd.org/plat.html>): servers, workstations, notebooks, etc.

Tips and Tricks

38 (Ab)using VideoLAN: Learn what you can do with your video and audio using powerful VideoLAN command line interface

Michael Bushkov

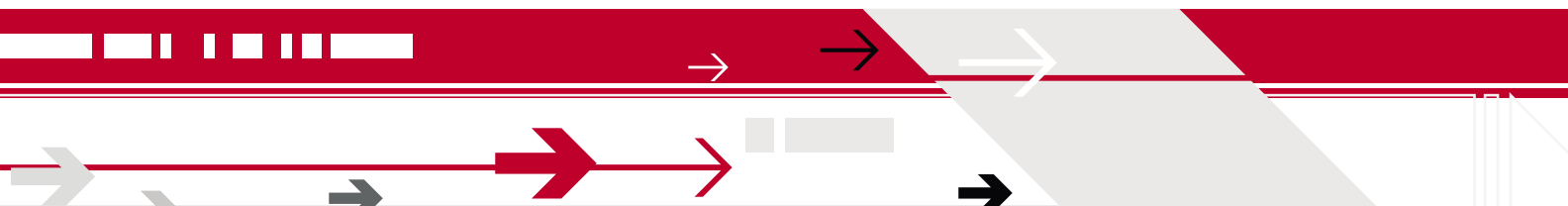
Dealing with video and audio data is the part of our everyday life. Sometimes, though, we need to do things that fall into „advanced” category. What tools should we use then?

Security

42 NetBSD Intrusion Detection Server. How can we describe the functions of such a server?

Svetoslav Chukov

Sometimes special type of systems are needed to be running on the server. This server will serve different purposes, it will take care of the network security.



DNSSEC resolution and IPv6

Unbound on FreeBSD 8.2

Unbound runs on FreeBSD, OpenBSD, NetBSD, Linux, and Microsoft Windows.

What you will learn...

- how to install and configure DNSSEC for small networks

What you should know...

- basic FreeBSD concepts
- basic DNS concepts

Unbound provides a reasonably simple way to implement DNSSEC in a local-area network. With Unbound forward and reverse resolution is possible for small networks where IPv6 is implemented.

You could modify this example installation against your network, and possibly have Unbound serving DNS on your network in a few hours. This example configures a authoritative, validating, recursive, and caching DNS server.

Before installing the Unbound DNS validating resolver, it might be a good idea to have a recent version of OpenSSL from ports:

```
# cd /usr/ports/security/openssl
# make install clean
```

I enabled `TLS_EXTRACTOR` and `SCTP` for the case that it might be interesting to use them sometime in the future.

Next, install the resolver:

```
# cd ../../dns/unbound
# make install clean
```

Even though Paul Vixie might disagree- I did not want to have much limitation on outgoing ports, so I enabled `LIBEVENT`. I did not think of a reason to enable `THREADS` or `GOST`. If you have Python programming, perhaps you will enable `PYTHON`.

Before modifying the configuration file, get a copy of *root.hints*:

```
# wget ftp://FTP.INTERNIC.NET/domain/named.cache -O \
  /usr/local/etc/unbound/root.hints
```

To use DNSSEC put a key file in `/usr/local/etc/unbound` and name it *root.key*. The file will contain one line:

```
. IN DS 19036 8 2 \
  49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE3
  2F24E8FB5
```

You can check for a more recent version at:

```
http://data.iana.org/root-anchors/root-anchors.xml.
# chown unbound /usr/local/etc/unbound/root.key
```

Before moving on to the Unbound configuration file, make some changes to FreeBSD. Technical example addressing is used in this example, so if you do not already have private IPv6 addresses, then search and study unique local addressing. Add to `/etc/rc.conf`:

```
ipv6_enable="YES"
ipv6_ifconfig_ethCard0="2001:0db8::xxxx:xxxx:xxxx:xxxx/32"
unbound_enable="YES"
```

Listing 1. *unbound.conf* after modifications

```

interface 192.0.2.2
interface: ::1
2001:0db8::xxxx:xxxx:xxxx:xxx2
outgoing-interface: 192.0.2.2
outgoing-port-permit: "49152-65535"
do-tcp: no
access-control: 192.0.2.0/28 allow
access-control: 2001:0db8::/32 allow
root-hints: "/usr/local/etc/unbound/root.hints"
hide-identity: yes
hide-version: yes
do-not-query-localhost: yes
val-log-level: 2
local-zone: "example.org." typetransparent
local-data: "host1.example.org. A 192.0.2.1"
local-data-ptr: "192.0.2.1 host1.example.org"
local-data: "host2.example.org A 192.0.2.2"
local-data-ptr: "192.0.2.2 host2.example.org"
local-data: "host3.example.org. A 192.0.2.3"
local-data-ptr: "192.0.2.3 host3.example.org"
local-data: "host1.example.org AAAA 2001:0db8::xxxx:xxxx:xxxx:xxx1"
local-data-ptr: "2001:0db8::xxxx:xxxx:xxxx:xxx1 host1.example.org"
local-data: "host2.example.org AAAA 2001:0db8::xxxx:xxxx:xxxx:xxx2"
local-data-ptr: "2001:0db8::xxxx:xxxx:xxxx:xxx2 host2.example.org"
local-data: "host3.example.org AAAA 2001:0db8::xxxx:xxxx:xxxx:xxx3"
local-data-ptr: "fdcd:30b3:af99::xxxx:xxxx:xxxx:xxx3 host3.example.org"

```

On the server change `/etc/resolv.conf`:

```
::1
```

For the hosts `resolv.conf`:

```
2001:0db8::xxxx:xxxx:xxxx:xxx2
```

From here, let us move on to the Unbound configuration file. The `unbound(8)` configuration file can be found in `/usr/local/etc/unbound`. Copy `unbound.conf.sample` to `unbound.conf`. Before actually using the file, the utility `unbound-checkconf(8)` can be run to check for errors; e.g.,

```
% unbound-checkconf
unbound-checkconf:
no errors in /usr/local/etc/unbound/unbound.conf
```

Next are the modifications to `unbound.conf(5)`. Most of the default entries are left alone in this example. You could do some performance tweaking for your server.

Run unbound-checkconf

Since we are using DNSSEC, run `unbound-anchor(8)` before starting the server:

```
# unbound-anchor -a „/usr/local/etc/unbound/root.key“
```

Type `unbound` or restart the server.

Now, if you have configured some of your applications using IPv6 then the hostnames will be available; e.g., if you run `ntp.org` then the standard NTP query program will return hostnames instead of IPv6 addresses, which is very handy if you are looking at a terminal window. :)

```
% ntpq -p
```

DARREL

Darrel is still recovering from a car crash and found that writing this article is not nearly as depressing as supineness. Due to a thunderstorm that began shortly after that sentence- the main thing keeping this article moving now is his uninterruptable power supply.

Keeping up to date in PC-BSD 9

Since the early days of PC-BSD, there has been various GUI mechanisms for performing critical system and security updates.

PC-BSD

While these tools were necessary, they were badly in need of an overhaul to provide traditional command-line functionality, along with mechanisms for performing a greater variety of update types. In the upcoming PC-BSD 9 the new `pc-updatemanager` makes its debut, with many new features, pure command-line functionality and a streamlined GUI which makes desktop updating as painless as possible. First let us take a look at some of the functionality of this new tool from the command-line perspective.

In PC-BSD 9, all upgrade functionality can now be performed via the console, using the commands `freebsd-update`, for system security advisories, and `pc-updatemanager`, for updates to packages, tools and major system versions. The former command, `freebsd-update` is included within the FreeBSD base operating system, and can now be safely used to perform security updates to the underlying operating system kernel and world environment. PC-BSD has always shipped with a default FreeBSD world environment, but starting in 9 it will include the GENERIC kernel as well, allowing `freebsd-update` to manage the full spectrum of security updates. More information on the usage of this built-in command can be found in the FreeBSD handbook below: <http://www.freebsd.org/doc/handbook/updating-freebsdupdate.html>.

The `pc-updatemanager` command is unique to PC-BSD 9, and provides a few easy to use commands which can be used to check for, and install several different types of updates. To start checking for updates, the first command to run is:

```
# pc-updatemanager check
```

This command will connect to the PC-BSD update server and fetch the latest digitally signed patch data for your specific version / architecture. If no updates are found, or your system is already updated then the command will exit with a message to that effect. If an update is found, then another message with details about the available update will be printed, as shown in the example below:

```
# pc-updatemanager check
```

```
The following updates are available:
```

```
-----
```

```
NAME: System Update to 9.0-BETA2
```

```
TYPE: SYSUPDATE
```

```
VERSION: 9.0-BETA2
```

```
DATE: 2011-08-18
```

```
TAG: release-9.0-BETA2
```


DETAILS: <http://www.pcbsd.org>

To install this update run „`pc-updatemanager install release-9.0-BETA2`”

In this example only a single update has been found, which will upgrade the system (in this case one running 9.0-BETA1) to BETA2. The command to start the particular update is always printed at the end of the update details, making it easy for the user to immediately begin the update process. Most updates are small patches which can be downloaded and installed in only a few minutes without a reboot. Usually this will be simply fetching and updating a particular package, such as the latest NVIDIA driver, or some newer version of a PC-BSD utility with important bug fixes. In this example we will look at a more complex update of the entire operating system to a newer release.

By starting the update in the example above, the `pc-updatemanager` would first begin by analyzing the system configuration and determining which desktops / meta-pkgs are installed, such as KDE, GNOME, LXDE, NVIDIA drivers, etc. After building this list, the update manager will start downloading the newer packages for these components, along with a new FreeBSD world / kernel. Once all files are downloaded and checksums verified, the user will be prompted to reboot the system and begin the upgrade. After rebooting, the update manager will start by removing the users old system packages and installing the newer kernel / world environment. When done, the system will automatically reboot, and finish the update by installing the updated desktop / meta-pkgs. This process is entirely automated, and requires no interaction from the user, apart from rebooting the system to begin the update. This initial reboot is used to allow the user to finish working on their desktop, without the worry of a critical package being modified at a inconvenient moment.

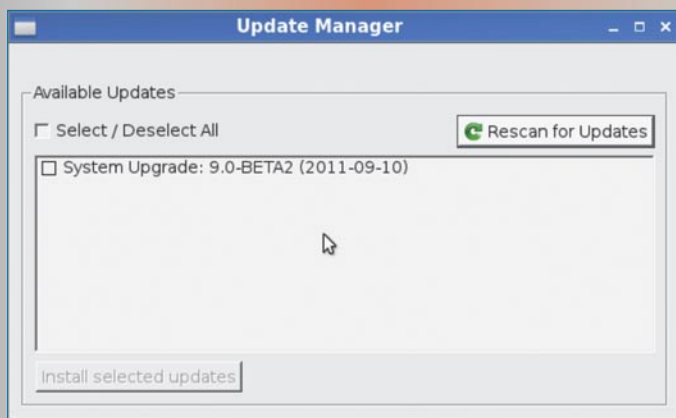


Figure 1. Update GUI for PC-BSD 9

While the `pc-updatemanager` is capable of handling a wide variety of update types, the configuration of it is relatively simple with only a few important options to take note of. Nearly all settings are stored in the main configuration file, `/usr/local/etc/pcbsd.conf`. Some common settings are listed below, with a brief description of each. In addition, all of these settings may also be set via the GUIs in the PC-BSD control panel, allowing users uncomfortable with the command-line to customize with only a few clicks.

```
# Mirror for System Updates / Meta-Pkgs
PCBSD_MIRROR: ftp://ftp.pcbsd.org/pub/mirror
```

```
# Proxy Server URL
PCBSD_PROXYURL: http://proxy.example.org
```

```
# Proxy Server Port
PCBSD_PROXYPORT: 8080
```

```
# Proxy Username
PCBSD_PROXYUSER: kris
```

```
# Proxy Password
PCBSD_PROXYPASS: example
```

At the moment the only settings normally adjusted are the ones shown above, such as changing the default mirror server, or adjusting the system to use a proxy server for connectivity. These can also be set in the System Manager and Network Manager GUI's respectively.

We've taken a look at the command-line functionality of the new `pc-updatemanager`, but for most desktop users a GUI solution is often the only viable one. In 9.0 the GUI tools have been slimmed down and streamlined into a single interface which can perform updates from both the `pc-updatemanager` and `freebsd-update` CLI backends.

With both a fully command-line driven backend, and easy to use front-end PC-BSD has never been easier to keep up to date with the latest security patches and versions. Administrators also have a new degree of control, by being able to disable the GUI entirely via `sudo`, and perform updates via the command-line transparent to the desktop user.

KRIS MOORE

Kris Moore is the founder and lead developer of PC-BSD. He lives with his wife and four children in East Tennessee (USA), and enjoys building custom PC's and gaming in his (limited) spare time. kris@pcbsd.org

Using Life Preserver

to Backup a PC-BSD 9.0 System to FreeNAS™ 8.0.1

This article demonstrates how to use the built-in Life Preserver program to backup a PC-BSD 9.0 desktop system to a FreeNAS™ 8.0.1 NAS system. Users can refer to the Guides at http://wiki.pcbbsd.org/index.php/PC-BSD_9_Handbook and <http://doc.freenas.org> for instructions on how to install PC-BSD and FreeNAS™.

PC-BSD

What you will learn...

- how to create an automated backup solution

What you should know...

- how to install PC-BSD and FreeNAS™

PC-BSD provides a graphical Life Preserver utility to make it easy for a desktop user to back up their home directory to another computer or storage appliance using rsync and SSH. Once a full backup has been created, rsync will only send the files that have changed since the last backup to the backup device. The

data is protected while being transferred over the network due to the encryption provided by SSH.

Configure FreeNAS™

In order to prepare the FreeNAS™ system to store the backups created by Life Preserver, you will need to: create a dataset to store the user's backup, create a user account that has permission to access that dataset, and enable the SSH and rsync services.

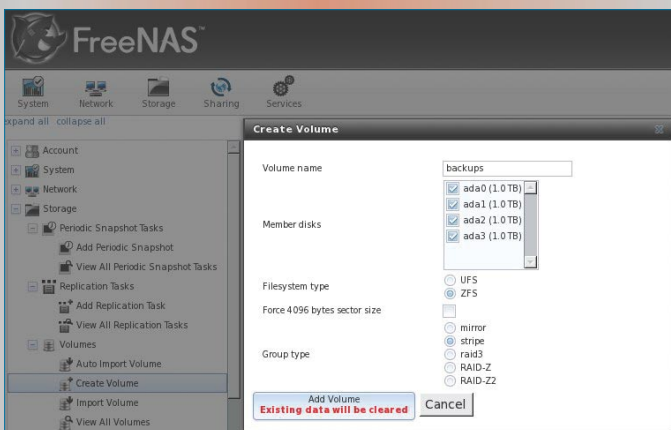


Figure 1. Create a ZFS Volume+



Figure 2. Creating a Dataset from a ZFS Volume



Figure 3. Creating a ZFS Dataset

Create a Dataset

In ZFS terminology, a dataset is a portion of a ZFS volume. Datasets allow you to create a storage area for an individual user; datasets also allow you to configure compression and a storage quota on a per dataset basis. Users will only see the data on their own dataset and are restricted to the disk space that you configure for the dataset.

Before you can create a dataset, you must first create a ZFS volume. In the FreeNAS™ 8.0.1 web administration interface, go to *Storage->Volumes->Add Volume*. As seen in Figure 1, the available (unformatted) disks will be listed.

In this example, the FreeNAS™ system has four 1TB drives. If I select to create a ZFS stripe using all four drives, the resulting volume will have the maximum storage capacity (~3.6TB) but will not have any redundancy (if one drive fails, the entire volume fails). If I select to create a ZFS RAIDZ1, the resulting volume will provide redundancy (can survive the failure of one disk), but will have reduced storage capacity (~2.8 TB) due to the parity information. I have chosen to create a ZFS

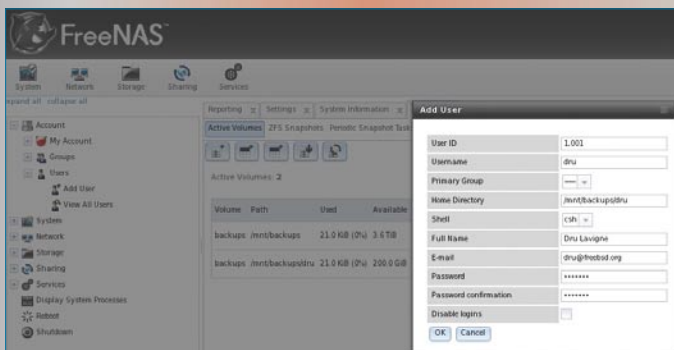


Figure 4. Creating a User Account

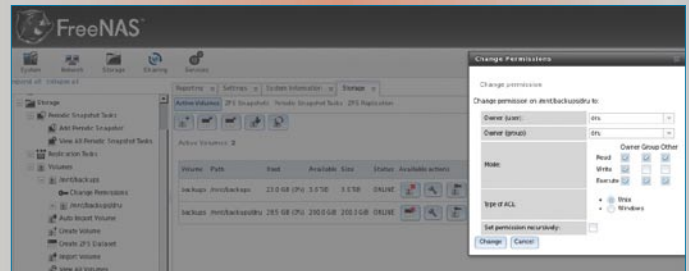


Figure 5. Viewing a Dataset's Permissions

stripe named backups. Once the volume is created, it will appear in *Storage->Volumes->View all Volumes*, as seen in Figure 2.

Click the icon Create ZFS Dataset to see the screen shown in Figure 3. In this example, a dataset named dru was created with a disk quota of 200GB. If your network contains multiple PC-BSD desktops or if several users share the PC-BSD system, create a dataset for each user. You can make as many datasets as you wish, assuming that free disk space still exists on the ZFS volume.

If you choose to use quotas, be sure to give the dataset sufficient space to store a full backup and the amount of incremental backups that you will schedule (e.g. a week's or a month's worth of daily backups).

Create a User

Once you have created the dataset, create a user account to associate with each dataset. To create a user account, go to *Account->Users->Add User*. In the example shown in Figure 4, a user account has been created for dru.

IMPORTANT

Change the Home Directory to the full pathname of the dataset for this user; in this example it is `/mnt/backups/dru`.

If you are configuring backups for several users, create a user account for each user, being sure to give each user their own dataset as their home directory.

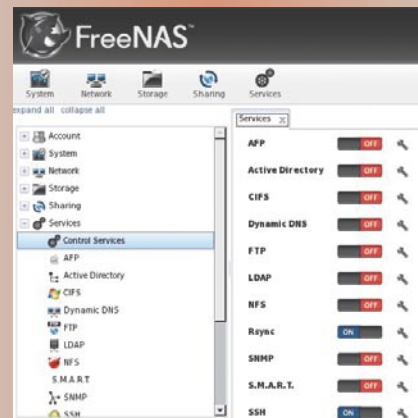


Figure 6. Enable the Rsync and SSH Services

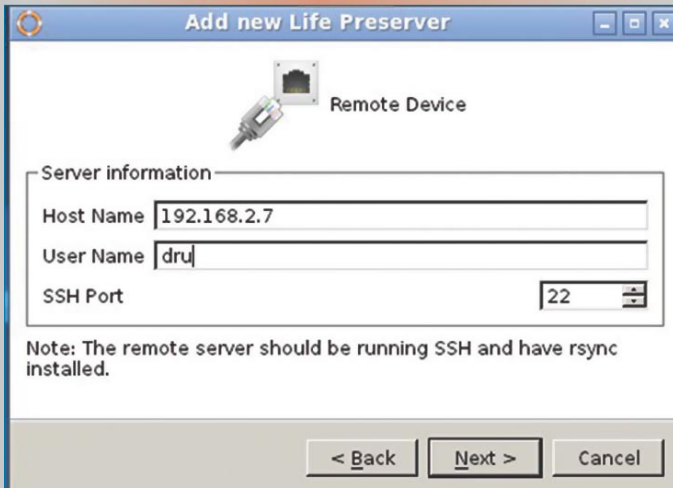


Figure 7. Input the IP Address and Username

Check Dataset Permissions

You can verify that the dataset's permissions are correct by going to *Storage->Volumes->View All Volumes* and clicking the Change Permissions icon (third from the left). In the example shown in Figure 5, the user *dru* has permission to the dataset; this was automatically configured when the dataset path was selected as the user's home directory. Depending upon your needs, you may wish to remove the read permissions for group and other; note that this will not affect the superuser's ability to read the files in the backup. Do not change the type of ACL (keep it at Unix).

Configure SSH and Rsync

To enable the rsync and SSH services on FreeNAS™, go to *Services->Control Services*. Click the red OFF button next to Rsync. After a second or so, it will change to a blue ON, indicating that the service has been enabled. Repeat for the SSH service.

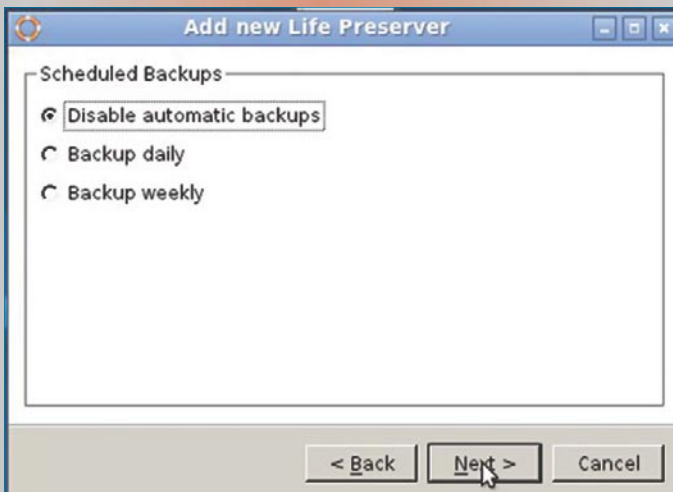


Figure 8. Select the Backup Schedule



Figure 9. Testing the Connection to the FreeNAS™ System

Configure Life Preserver

In version 9.0 of PC-BSD, Life Preserver appears as an icon in the system tray. It can also be launched from *Control Panel->Life Preserver*.

The first time you run Life Preserver, the Life Preserver Wizard will launch, indicating that you need to know the IP address and username/password to connect to the backup device. Click the Get Started button, then Next to see the screen shown in Figure 7. Input the IP address of the FreeNAS™ system and the name of the user account that you created and associated with a dataset.

Click Next and select how often you would like the backup to occur, as seen in Figure 8. The default is to not create an automatic backup, meaning that you will perform the backup manually as needed. You can choose to instead automatically backup your home directory once a day or once a week.

After making your selection, click Next then Finish. The Wizard will display a message indicating that it will test the connection to the FreeNAS™ system. Click Finish again and input the word *yes* and then the user's password when prompted, as seen in Figure 9.

Once the connection is successful, the *preserver* (the configuration for the backup) will appear in the preservers list, as seen in Figure 10, with the following information:

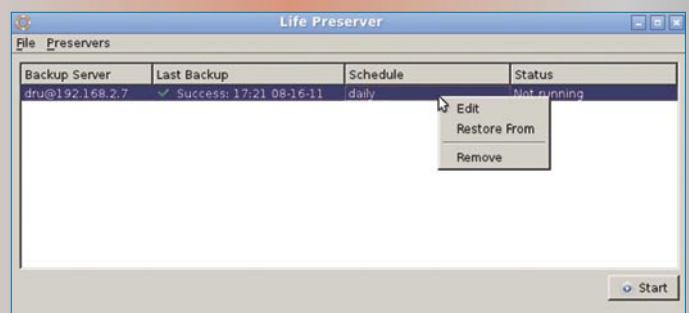


Figure 10. Daily Preserver with a Successful Backup

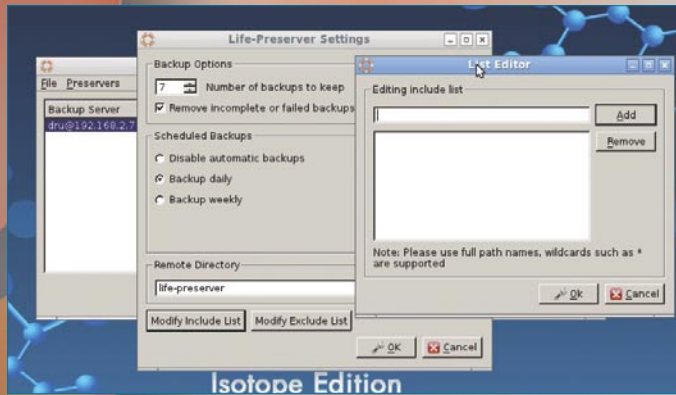


Figure 11. Editing a Preserver's Settings and Include List

Backup Server

Will indicate the user account and IP address of the backup server.

Last Backup

Will indicate whether or not the last backup was successful. If you chose to automate backups, the first backup will happen immediately. Otherwise, a backup will not occur until you press the Start button. How long the first backup takes depends upon the size of your home directory and the speed of your network.

Schedule

Will indicate disabled, daily, or weekly.

Status

Running indicates that the backup is occurring now, otherwise will show as not running.

If you right-click the preserver, you can choose to edit the settings, restore from a backup, or remove the configuration.

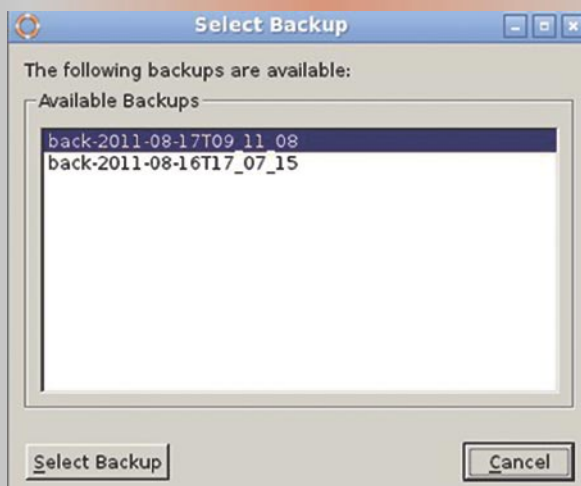


Figure 12. List of Backups

The BSD Certification Group Inc. (BSDCG) is a non-profit organization committed to creating and maintaining a global certification standard for system administration on BSD based operating systems.

? WHAT CERTIFICATIONS ARE AVAILABLE?

BSDA: Entry-level certification suited for candidates with a general Unix background and at least six months of experience with BSD systems.

BDSP: Advanced certification for senior system administrators with at least three years of experience on BSD systems. Successful BDSP candidates are able to demonstrate strong to expert skills in BSD Unix system administration.

✓ WHERE CAN I GET CERTIFIED?

We're pleased to announce that after 7 months of negotiations and the work required to make the exam available in a computer based format, that the BSDA exam is now available at several hundred testing centers around the world. Paper based BSDA exams cost \$75 USD. Computer based BSDA exams cost \$150 USD. The price of the BDSP exams are yet to be determined.

Payments are made through our registration website:
<https://register.bsdcertification.org/register/payment>

i WHERE CAN I GET MORE INFORMATION?

More information and links to our mailing lists, LinkedIn groups, and Facebook group are available at our website:
<http://www.bsdcertification.org>

Registration for upcoming exam events is available at our registration website:
<https://register.bsdcertification.org/register/get-a-bsdcg-id>

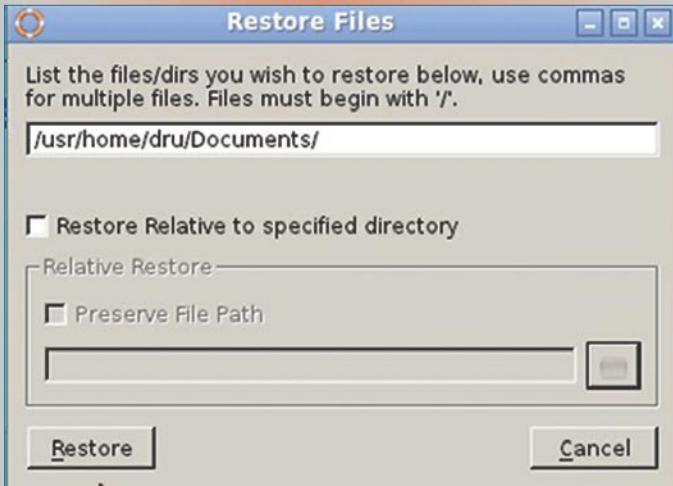


Figure 13. Choosing Which File or Directory to Restore

Figure 11 shows the screen if you select Edit, as well as the screen if you also select Modify Include List.

By default, Life Preserver makes a backup of the user's home directory and stores the last 7 backups. If you wish to exclude files from your home directory or include files outside of your home directory, use the buttons to Modify Exclude List or Modify Include List.

Restoring Files

If you choose the option Restore From, you will be presented with a list of the stored backups. In the example shown in Figure 12, the preserver is scheduled to backup daily and a backup exists for August 17 (`back-2011-08-17T09_11_08`) and August 16 (`back-2011-08-16T17_07_15`). If I highlight the backup for August 17 and click Select Backup, I'll see the screen in Figure 13. In this example, I've chosen to restore my Documents directory.

When doing a restore, give the full path to the file or directory. The full path will always begin with `/usr/home/$USERNAME/` where you replace `$USERNAME` with the name of your user.

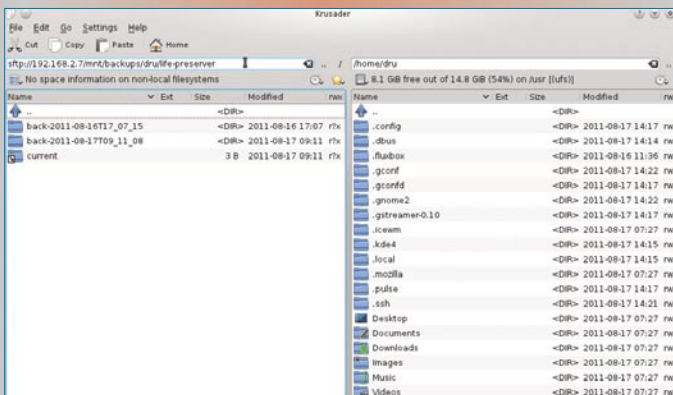


Figure 14. Using Krusader to Browse Backups

Using a Graphical File Manager to View Backups

Since Life Preserver uses SSH to transfer the backups and restores, users can use SSH utilities such as `ssh`, `scp`, and `sftp` to view and copy the files in their dataset. If you prefer to use a graphical rather than a command-line utility, there are several options available. Depending upon which desktop you are logged into, you may or may not have a graphical utility that understands SSH. If you're not sure, Krusader2 is available from AppCafe™ and provides a dual-pane file manager interface that understands `sftp`.

To access the FreeNAS™ system using Krusader, type `sftp:/102.168.2.7` into the address bar of one of the panes, replacing the IP address with the value for your FreeNAS™ system. When the login prompt appears, input the username and password of your user.

Figure 14 shows a listing of the stored backups in the life-preserver directory of the left pane and the user's home directory on their PC-BSD system in the right pane. If you expand either a backup or current (a shortcut to the latest backup), you can navigate to `usr/home/$USER` and view the contents of your user's home directory. You can then highlight the files/directories that you wish to restore, right-click on the selection, click Copy, and the selection will be copied to the home directory on the PC-BSD system.

Summary

This article demonstrated how easy it is to backup a user's home directory to a FreeNAS™ system using PC-BSD's built-in Life Preserver utility. It also demonstrated how to use the graphical Krusader utility to view backups and perform file and directory restores.

DRU LAVIGNE

Dru Lavigne is author of BSD Hacks, The Best of FreeBSD Basics, and The Definitive Guide to PC-BSD. As Director of Community Development for the PC-BSD Project, she leads the documentation team, assists new users, helps to find and fix bugs, and reaches out to the community to discover their needs. She is the former Managing Editor of the Open Source Business Resource, a free monthly publication covering open source and the commercialization of open source assets. She is founder and current Chair of the BSD Certification Group Inc., a non-profit organization with a mission to create the standard for certifying BSD system administrators, and serves on the Board of the FreeBSD Foundation.

EXOnetric



Reliable FreeBSD Jails and hosting at the heart of the UK Internet



**Find out what we
can do for you today...**

Exonetric Consulting Ltd.
Tel. +44 (0) 870 787 9394
Fax. +44 (0) 870 787 9395

www.exonetric.com
info@exonetric.com

Recovering data with hammer

We've all experienced instant regret. That's the feeling that comes within a second of executing a command like „rm -rf * .txt” (note the space), or of cutting the wrong cluster of wires at the end of a long conduit. Not that I am quoting from experience, or anything like that, no...

Hammer, DragonFly's default file system, can help with that. Perhaps not with the cut wires, but with the loss of important files. Hammer will keep a record of the data changed every time a disk is synced, so approximately every 30 seconds or so file history is saved, under normal circumstances.

Hammer also support snapshots, where the state of an entire filesystem is saved for live access later. Since this file history and snapshots only contains the changes to the data, it's relatively sparse and doesn't eat much more storage space.

These two aspects together mean that if you are going to make a mistake, doing it while on a Hammer-running operating system can make your life much easier. This article contains some „case studies” of the various ways Hammer fixes what you did wrong.

Simple case: I scrambled a file

The most simple case: you've scrambled a file. Maybe you rewrote several lines and saved it, or accidentally



DragonFlyBSD

mashed the keyboard, but either way, the file is still present – just wrong.

By default, the undo tool will output the previous version of a file with a note about the timestamp for that last change, prefixed with >>>. See Listing 1.

Other options exist, like using -i to iterate over all previous versions saved to disk, or ways to generate a diff. What if you delete the file? It'll still work.

Listing 1. simple undo

```
undo filename

>>> filename 0000 0x00000001c059c3d0 20-Aug-2011 19:
      24:18

(previous version info here)
```


More complicated: A lost file

This is all fine when you still have the known location of the file, but what if it's a month later, and you need one file out of hundreds in a directory? Manually retrieving each file and searching it would either be a large amount of labor, or some time writing an appropriate shell script.

This is where snapshots come in. Hammer volumes automatically take snapshots, and do so by default on a daily basis, storing up to 60 days of snapshots.

Snapshots are stored in disk meta-data, so they can be listed using the hammer command. See Listing 2.

Each one of those unique transaction IDs points to this system's /var as it looked at that date in time. The directory /var/hammer contains links to the history of each Hammer pseudo-filesystem. Listing 3 shows example contents for the usr directory in that setup.

Notice that the default name on each of the transaction links shows the date of the snapshot, so getting an initial snapshot list may not even be necessary,

It's possible to cd into the appropriate directory and perform operations as if it was a normal directory. It's read-only, of course, since it's a historical snapshot.



DragonFlyBSD

While this example shows automatic snapshots, it's possible to trigger snapshots at any arbitrary time. For example, it's possible to perform before-and-after comparisons when installing software, by taking a snapshot before installation and just after, and then using normal filesystem tools to compare the affected disk areas afterwards.

Really, really catastrophic recovery

If your Hammer filesystem becomes corrupt, perhaps due to bad disk firmware, there is a 'hammer recover' command. This command looks for any files that can be reconstructed based on what data is left on the disk, and rebuilds them. Even if the metadata that outlines the system is corrupted, the data itself may be still physically present and identifiable.

It's even possible to take an image of a Hammer volume and mount it on a virtual machine, and run 'hammer recover' there, to rebuild data without risking further loss from physical disk activity, in a scenario

where the hardware is itself damaged and likely to scramble itself further.

Note that I didn't say anything about a power outage; Hammer is designed to survive sudden cuts of power. Anything's possible in a power surge or loss, of course, but one of the initial tests for Hammer was starting intensive disk operations and then yanking power from the running system, so some thought has been put into preventing power issues.

Conclusion

With Hammer, you can see every version of your file that's ever committed to disk, limited only by the Hammer settings and the available disk space. There's a lot more possible with Hammer. Snapshots can be streamed to other Hammer volumes over the network, for remote backup. Snapshots can be kept independently on those remote volumes, too... but that's another article.

Listing 2. Snapshot meta-data listing

```
# hammer snaps /var
Snapshots on /var      PFS #1
Transaction ID        Timestamp              Note
0x00000001b40cbf10    2011-06-21 03:01:06 EDT -
0x00000001b421f010    2011-06-22 03:01:05 EDT -
[listing trimmed to save paper]
0x00000001bfe1b6a0    2011-08-19 03:01:06 EDT -
0x00000001c028a210    2011-08-20 03:01:06 EDT -
```

Listing 3: Automatic snapshots

```
# ls /var/hammer/usr
snap-20110622-0301 -> /usr/@0x00000001b422f0f0
snap-20110623-0301 -> /usr/@0x00000001b4389140
snap-20110624-0301 -> /usr/@0x00000001b45050f0
[again, trimmed to save paper]
snap-20110818-0303 -> /usr/@0x00000001bf97cc30
snap-20110819-0301 -> /usr/@0x00000001bfe1b820
snap-20110820-0301 -> /usr/@0x00000001c028a310
```

Apache2, php5, mysql5, modsecurity2.5

5 installation and configuration in order to protect dynamic websites from various attacks, in FreeBSD 8.2

In the last years there is a tremendous increment in dynamic website and cms using php. A very large piece of the market of this websites are served by Apache Webserver using Mysql as database basically in Unix systems. Also this tremendous increment of php in dynamic website and opensource cms like Joomla increase and hackers attacks in order to compromise a website or hack the server to use it in botnet. So someone can wonder, is there anything that can protect my websites except from backups and upgrading our system and software? The answer is yes.

What you will learn...

- Installing and configuring apache 2.2.x
- Installing and configuring php5.3.x
- Installing modules for php5
- Installing and configuring Mysql5
- Installing and configure mod_security 2.5
- How to test your site for attacks like sql injection and Cross Site Scripting

What you should know...

- Installing FreeBSD 8.2
- Using vi or any Console editor
- Basic unix command like mv, cp etc
- Installing Joomla 1.7 CMS
- Using phpmyadmin

In this article i am going to guide you step by step how to install apache2.2.X web server, php5.3.x and configure apache run php scripts in order to host dynamic website or CMS like Joomla in FreeBSD. I will also show the procedure to install mysql and phpmyadmin in order to manage mysql database easily. Then we will secure apache web server from various attacks like XSS using modsecurity and finally we will install Joomla CMS and then trying some hacking on it to see if the web server is secured. First add

```
hostname="your.hostname.com"
```

```
to /etc/rc.conf.  
Update ports tree
```

```
#portsnap fetch
```

If you run portsnap for first time then use

```
#portsnap extract
```

And then

```
#portsnap update
```

Else you can use portsnap update directly without first need to use command portsnap extract.

Or you can use `pkg_add` utility but i prefer using ports and compiling my packages instead using precompile packages.

Installing portaudit

Portaudit is a very nice utility that check install ports or ports that are going to be installed if are vulnerable.

```
#cd /usr/ports/ports-mgmt/portaudit  
#make instal clean
```

Reload shell commands

```
#rehash
```

```
localhost# portaudit -F  
auditfile.tbz 100% of 69 kB 48 kBps  
New database installed.  
localhost#
```

Figure 1. Choosing apache modules to be installed > this will go in installing apache above modules

Update portaudit db to get new vulnerabilities

```
#portaudit -F
```

Installing apache

Go to port directory

```
#cd /usr/port/www/apache22
#make install clean
```

In the menu that appears we can disable modules or enable modules that we will need. In this setup we are going to use the webserver to serve websites not svn so I disable modules like `mod_dav` because of some vulnerabilities. We enable or disable features using spacebar and tab to go to OK button (Figure 1).

About modules

- `mod_access` – Provides access control based on client hostname, IP address, or other characteristics of the client request.
- `mod_actions` – This module provides for executing CGI scripts based on media type or request method.
- `mod_alias` – Provides for mapping different parts of the host filesystem in the document tree and for URL redirection
- `mod_asis` – Sends files that contain their own HTTP headers
- `mod_auth` – User authentication using text files
- `mod_auth_anon` – Allows *anonymous* user access to authenticated areas
- `mod_auth_dbm` – Provides for user authentication using DBM files
- `mod_auth_digest` – User authentication using MD5 Digest Authentication.
- `mod_auth_ldap` – Allows an LDAP directory to be used to store the database for HTTP Basic authentication.
- `mod_autoindex` – Generates directory indexes, automatically, similar to the Unix `ls` command or the Win32 `dir` shell command
- `mod_cache` – Content cache keyed to URIs.
- `mod_cern_meta` – CERN httpd metafile semantics
- `mod_cgi` – Execution of CGI scripts
- `mod_cgid` – Execution of CGI scripts using an external CGI daemon
- `mod_charset_lite` – Specify character set translation or recoding
- `mod_dav` – Distributed Authoring and Versioning (WebDAV) functionality
- `mod_dav_fs` – filesystem provider for `mod_dav`
- `mod_deflate` – Compress content before it is delivered to the client
- `mod_dir` – Provides for *trailing slash* redirects and serving directory index files
- `mod_disk_cache` – Content cache storage manager keyed to URIs
- `mod_dumpio` – Dumps all I/O to error log as desired.
- `mod_echo` – A simple echo server to illustrate protocol modules
- `mod_env` – Modifies the environment which is passed to CGI scripts and SSI pages
- `mod_example` – Illustrates the Apache module API
- `mod_expires` – Generation of Expires and Cache-Control HTTP headers according to user-specified criteria
- `mod_ext_filter` – Pass the response body through an external program before delivery to the client
- `mod_file_cache` – Caches a static list of files in memory
- `mod_headers` – Customization of HTTP request and response headers
- `mod_imap` – Server-side imagemap processing
- `mod_include` – Server-parsed html documents (*Server Side Includes*)
- `mod_info` – Provides a comprehensive overview of the server configuration
- `mod_isapi` – ISAPI Extensions within Apache for Windows
- `mod_ldap` – LDAP connection pooling and result caching services for use by other LDAP modules
- `mod_log_config` – Logging of the requests made to the server
- `mod_log_forensic` – Forensic Logging of the requests made to the server
- `mod_logio` – Logging of input and output bytes per request
- `mod_mem_cache` – Content cache keyed to URIs
- `mod_mime` – Associates the requested filename's extensions with the file's behavior (handlers and filters) and content (mime-type, language, character set and encoding)
- `mod_mime_magic` – Determines the MIME type of a file by looking at a few bytes of its contents
- `mod_negotiation` – Provides for *content negotiation*
- `mod_nw_ssl` – Enable SSL encryption for NetWare
- `mod_proxy` – HTTP/1.1 proxy/gateway server
- `mod_proxy_connect`
- `mod_proxy` extension for CONNECT request handling
- `mod_proxy_ftp` – FTP support module for `mod_proxy`
- `mod_proxy_http` – HTTP support module for `mod_proxy`

```

==> License check disabled, port has not defined LICENSE
==> Found saved configuration for apache-2.2.19
-> httpd-2.2.19.tar.bz2 doesn't seem to exist in /usr/ports/distfiles/apache22.
-> Attempting to fetch http://www.apache.org/dist/httpd/httpd-2.2.19.tar.bz2
httpd-2.2.19.tar.bz2          100% of 5197 kB   645 kBps
==> Extracting for apache-2.2.19
-> SHA256 Checksum OK for apache22/httpd-2.2.19.tar.bz2.
==> apache-2.2.19 depends on file: /usr/local/bin/per15.12.3 - found

```

Figure 2. Apache Installation Procedure begins

- `mod_rewrite` – Provides a rule-based rewriting engine to rewrite requested URLs on the fly
- `mod_setenvif` – Allows the setting of environment variables based on characteristics of the request
- `mod_so` – Loading of executable code and modules into the server at start-up or restart time
- `mod_speling` – Attempts to correct mistaken URLs that users might have entered by ignoring capitalization and by allowing up to one misspelling
- `mod_ssl` – Strong cryptography using the *Secure Sockets Layer* (SSL) and *Transport Layer Security* (TLS) protocols
- `mod_status` – Provides information on server activity and performance
- `mod_suexec` – Allows CGI scripts to run as a specified user and Group
- `mod_unique_id` – Provides an environment variable with a unique identifier for each request

- `mod_userdir` – User-specific directories
- `mod_usertrack` – Clickstream logging of user activity on a site
- `mod_version` – Version dependent configuration
- `mod_vhost_alias` – Provides for dynamically confi-gured mass virtual hosting

More info for modules can be found in apache website <http://httpd.apache.org/docs/2.0/mod/>.

Then click tab to go to OK Button and click enter to continue (Figure 2). In the next screens that will appear (Figure 3) accept default values and click ok to continue installation when the installation finish you will se the Figure 4.

To make apache start at boot time edit `/etc/rc.conf` and add this line

```
#echo `apache22_enable="YES"` >> /etc/rc.conf
```

starting apache

```
#/usr/local/etc/rc.d/apache22 start
```

Disable Directory indexing. Change

```
Options Indexes FollowSymLinks
```

To

```
Options All -Indexes FollowSymLinks MultiViews
```

To check if module `mod_security` is loaded

```
#apachectl -t -D DUMP_MODULES
```

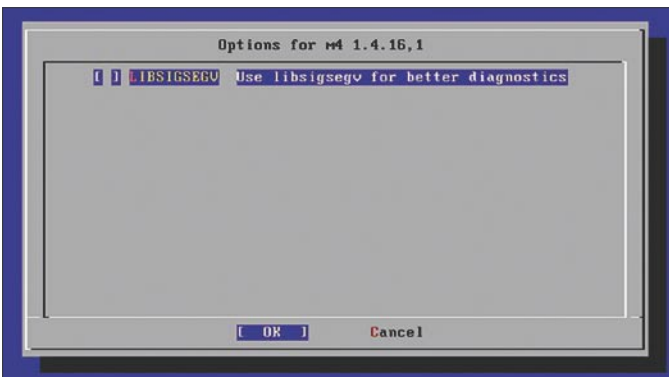


Figure 3. One of the many prompts you will get

```

==> Compressing manual pages for apache-2.2.19
==> Registering installation for apache-2.2.19
==> Cleaning for autoconf-2.68
==> Cleaning for libtool-2.4
==> Cleaning for expat-2.0.1_1
==> Cleaning for apr-ipv6-devrandom-gdbm-db42-1.4.5.1.3.12
==> Cleaning for pcre-8.12
==> Cleaning for libiconv-1.13.1_1
==> Cleaning for M4-1.4.16.1
==> Cleaning for help2man-1.48.4
==> Cleaning for gmake-3.82
==> Cleaning for autoconf-wrapper-20181119
==> Cleaning for python27-2.7.2_1
==> Cleaning for automake-1.11.1
==> Cleaning for gdbm-1.8.3_3
==> Cleaning for db42-4.2.52_5
==> Cleaning for p5-Locale-gettext-1.05_3
==> Cleaning for gettext-0.18.1.1
==> Cleaning for automake-wrapper-20181119
==> Cleaning for apache-2.2.19
#

```

Figure 4. Apache Installation Procedure Ends

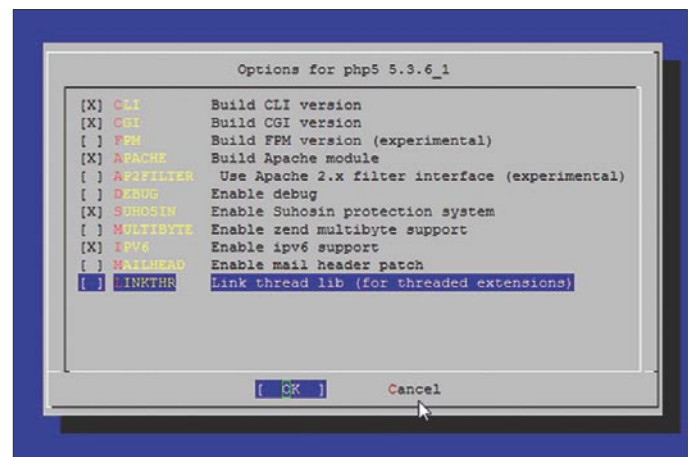


Figure 5. Configuring php

You should see in the list

```
unique_id_module (shared)
security2_module (shared)
```

Note

If you get warning

[warn] (2)No such file or directory: Failed to enable the *httpready* Accept Filter

add the following line into `/boot/loader.conf`:

```
echo 'accf_http_load="YES"' >> /boot/loader.conf
```

and restart system to load it.

Installing php

```
# cd /usr/ports/lang/php5
# make config
```

Check build apache module and click ok (Figure 5).

```
# make install clean
```

Check in `/usr/local/etc/apache22/httpd.conf` if there is line

```
LoadModule php5_module libexec/apache22/libphp5.so
```

Also modify this line

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

With this line

```
<IfModule dir_module>
    DirectoryIndex index.php index.htm index.html
</IfModule>
```

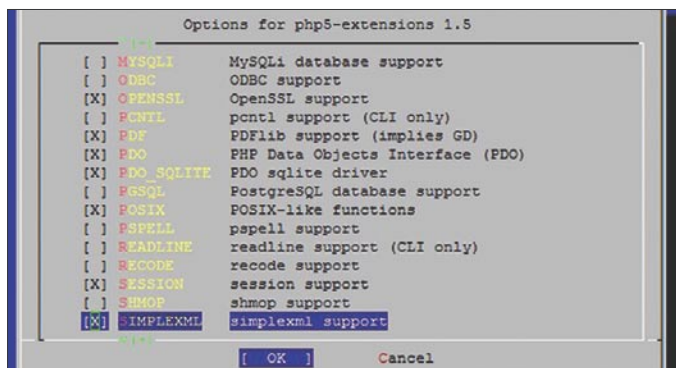


Figure 6. Choosing php extensions

And also add this line inside `httpd.conf`

```
<IfModule php5_module>
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
</IfModule>
```

Create `php.ini` config

```
#mv /usr/local/etc /php.ini-production /usr/local/etc /
php.ini
```

Now we will install php extension need for some cms like joomla

```
#cd /usr/ports/lang/php5-extensions
#make config
```

In the screen appears we choose except the defaults values also bz2, curl, exif, ftp, mysql, odf, pdf, session, gd, mcrypt, zip, zlib (Figure 6). Click ok.

Then start installation

```
#make install clean
```

And then click ok to continue installation. After installation finish restart apache


PHP Version 5.3.6 	
System	FreeBSD localhost.my.domain 8.2-RELEASE FreeBSD 8.2-RELEASE #0: Fri Feb 18 02:24:46 UTC 2011 root@almeida.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC i386
Build Date	Aug 9 2011 22:47:48
Configure Command	'./configure' '--with-layout=GNU' '--localstatedir=/var' '--with-config-file-scan-dir=/usr/local/etc/php' '--disable-all' '--enable-libxml' '--with-libxml-dir=/usr/local' '--with-pcre-regex=/usr/local' '--with-zlib-dir=/usr' '--program-prefix=' '--with-apxs2=/usr/local/bin/apxs' '--with-regex=php' '--with-zend-vm=CALL' '--prefix=/usr/local' '--mandir=/usr/local/man' '--infodir=/usr/local/info' '--build=i386-portbsd-freebsd8.2'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php
Additional .ini files parsed	/usr/local/etc/php/extensions.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626.NTS
PHP Extension Build	API20090626.NTS
Debug Build	no
Thread Safety	disabled

Figure 7. Phpinfo() website. Proof of php running on our apache

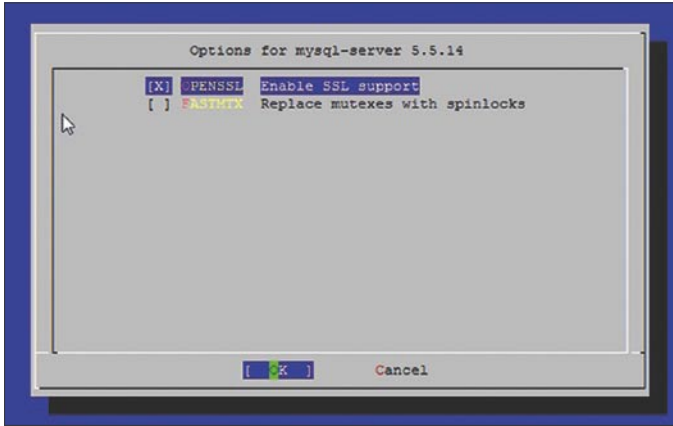


Figure 8. Configuring mysql Server

```
#/usr/local/etc/rc.d/apache22 restart
```

Create a test page to see if php is working. The best is using `phpinfo()` function

```
#mv /usr/local/www/apache22/data/index.html /usr/local
    /www/apache22/data/itworks.html

#echo „<? phpinfo(); ?>” >> /usr/local/www/apache22/data/
    index.php
```

And test your website in web browser

```
http://your domain name or your ip/
```

you should see a page like this see Figure 7.

Installing mysql

```
#cd /usr/ports/databases/mysql55-server
#make install clean
```

In the screen appears below keep default config and click ok (Figure 8).

Enable mysql server start at booting

```
#echo `mysql_enable="YES"` >> /etc/rc.conf
```

Start mysql server

```
#/usr/local/etc/rc.d/mysql-server start
```

Because mysql server by default is listening in all ip interface this is not secure. We want mysql server listen only on localhost because we are going to use the server for websites. So we need to add also in `rc.conf` `bind-address`. The command is

```
#echo `mysql_args="--bind-address=127.0.0.1"` >> /etc/
    rc.conf
```

And the we restart mysql to get the new settings

```
#/usr/local/etc/rc.d/mysql-server restart
```

If you want to manage mysql server instead of the command line you can install `phpmyadmin`. Is a nice web frontend that you can easily manage your databases.

Installation procedure is as follow Listing 1.

For security reasons we rename the default name of `phpmyadmin` folder and we add a random string like in the end like 5485

```
#mv phpMyAdmin-3.4.3.2-all-languages/ phpmyadmin_54td85
```

Now cd to directory

```
#cd /usr/local/www/apache22/data/phpmyadmin_54td85
#mv config.sample.inc.php config.inc.php
```

open `config.inc.php`

```
#vi config.inc.php
```

And find line

```
localhost# netstat -na
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp46 0 0 *.3306 *.* LISTEN
```

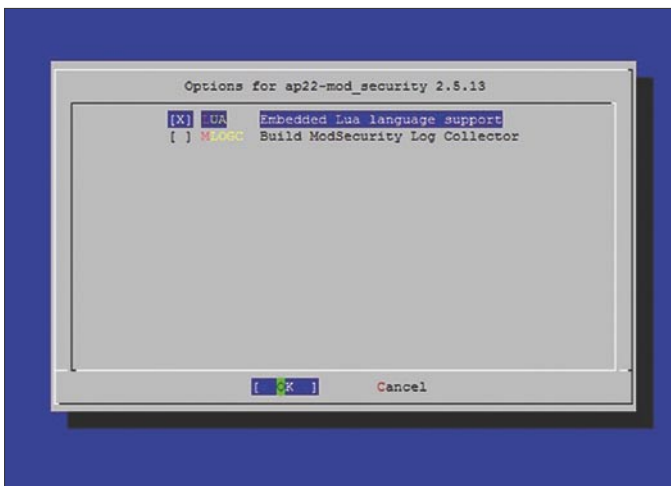
Figure 9. Netstat showing that mysql is listen to external interface

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp4 0 0 127.0.0.1.3306 *.* LISTEN
```

Figure 10. Netstat showing that mysql is listen to localhost and is more secure

Listing 1. Download and untar phpmyadmin, Mysql web frontend

```
#cd /usr/local/www/apache22/data/
#wget http://sourceforge.net/projects/phpmyadmin/files%2FphpMyAdmin%2F3.4.3.2%2FphpMyAdmin-3.4.3.2-all-
languages.tar.gz
#tar -zxvf files%2FphpMyAdmin%2F3.4.3.2%2FphpMyAdmin-3.4.3.2-all-languages.tar.gz && rm -rf files%2FphpMyAdmin%2F3.4.3
.2%2FphpMyAdmin-3.4.3.2-all-languages.tar.gz
```

**Figure 11.** Phpmyadmin web frontend**Figure 12.** Configure mod security screen before installation

```
====> License check disabled, port has not defined LICENSE
=> modsecurity-apache_2.5.13.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
=> Attempting to fetch http://heanet.dl.sourceforge.net/project/mod-security/modsecurity-apache/2.5.13/modsecurity-apache
_2.5.13.tar.gz
modsecurity-apache_2.5.13.tar.gz          100% of 1387 kB   600 kBps
====> Extracting for ap22-mod_security-2.5.13
=> SHA256 Checksum OK for modsecurity-apache_2.5.13.tar.gz.
```

Figure 13. Mod security installation finish

```
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

And replace it with

```
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

If the procedure is correct when you go to your browser and type the url

```
http://your domain name or your ip/phpmyadmin_54td85
```

you will see a picture like the one Figure 11.

If you see this page login to the system as root without password and then go to privileges and change all users password using Edit Privileges Password. You can use the same password for user root. But don't use the same password for other users you will create here.

Note

To increase security to this folder you can use apache htaccess to allow certain ips to access this folder.

Installing modsecurity

First we install LUA

```
#cd /usr/ports/lang/lua
#make install clean
```

and then mod security

```
#cd /usr/ports/www/mod_security
#make install clean
```

```
LoadModule unique_id_module libexec/apache22/mod_unique_id.so
LoadModule security2_module libexec/apache22/mod_security2.so
LoadModule setenvif_module libexec/apache22/mod_setenvif.so
LoadModule version_module libexec/apache22/mod_version.so
```

Figure 14. Apache `httpd.conf` with line for enable `mod security` check Embedded Lua language support and click ok (Figure 12). Process start (Figure 13).

When installation finish we have to enable module `unique_id` (if is not already enabled) in apache config and then `mod_security`

```
#cd /usr/local/etc/apache22
#vi httpd.conf
```

Below line

```
LoadModule unique_id_module libexec/apache22
/mod_unique_id.so
```

we add

```
LoadFile /usr/local/lib/libxml2.so
LoadFile /usr/local/lib/liblua-5.1.so
LoadModule security2_module libexec/apache22
/mod_security2.so
```

Restart apache

```
#!/usr/local/etc/rc.d/apache22 restart
```

Configure modsecurity

Change line

```
<IfModule security2_module>
    Include etc/apache22/Includes/mod_security2/*.conf
</IfModule>
```

To

```
<IfModule security2_module>
    Include etc/apache22/Includes/mod_security2/*.conf
    Include etc/apache22/Includes/mod_security2
    /base_rules/*.conf
    Include etc/apache22/Includes/mod_security2/asl/*.conf
</IfModule>
```

Now `modsecurity` config and rules files are in `/usr/local/etc/apache22/Includes/mod_security2`

```
#cd /usr/local/etc/apache22/Includes/mod_security2
```

Create a file name `modsecurity_crs_10_config.conf`

Listing 2. Modifying `modsecurity_crs_10_config.conf` to make `mod security` function

```
SecComponentSignature "core ruleset/2.0.10"

SecRuleEngine On
SecAuditEngine On

SecAuditEngine RelevantOnly
#SecAuditLogRelevantStatus "^(?:5|4(?:!04))"
#SecAuditLogType Serial
SecAuditLog /var/log/modsecurity_audit.log

SecDebugLogLevel 4
SecDebugLog /var/log/modsecurity_debug.log

SecRequestBodyAccess On
SecResponseBodyAccess On
SecResponseBodyMimeType (null) text/html text/plain
    text/xml
SecResponseBodyLimit 524288

# Server masking is optional
SecServerSignature "Microsoft-IIS/0.0"

SecDataDir /tmp

# Configures the directory where temporary files will be
    created.

SecTmpDir /tmp

# TODO Change the temporary folder setting to a path
    where only
#     the web server has access.
#
SecUploadDir /tmp

# Whether or not to keep the stored files.
#
# In most cases you don't want to keep the uploaded
    files (especially
# when there is a lot of them). It may be useful to
    change the setting
# to "RelevantOnly", in which case the files uploaded
    in suspicious
# requests will be stored.
#
SecUploadKeepFiles Off

SecDefaultAction "phase:2,deny,status:501,log"
```



```
#touch modsecurity_crs_10_config.conf
```

now we have to edit this file. Open it with your favourite editor e.x vi or pico

```
#vi modsecurity_crs_10_config.conf
```

and add the lines (Listing 2). If you don't have wget install it from ports because we will need it to download *tar.gz* files

```
#cd /usr/ports/ftp/wget
#make install clean
```

Download ASL rules (Listing 3) or just create a simuilink

```
#cd /etc
#ln -s /usr/local/etc/apache22/Includes/mod_security2/asl/ asl
```

We also zero domain-spam-whitelist.conf file because of an error in modsecurity

```
# cat /dev/null > /usr/local/etc/apache22/Includes
/mod_security2/ domain-spam-whitelist.conf
```

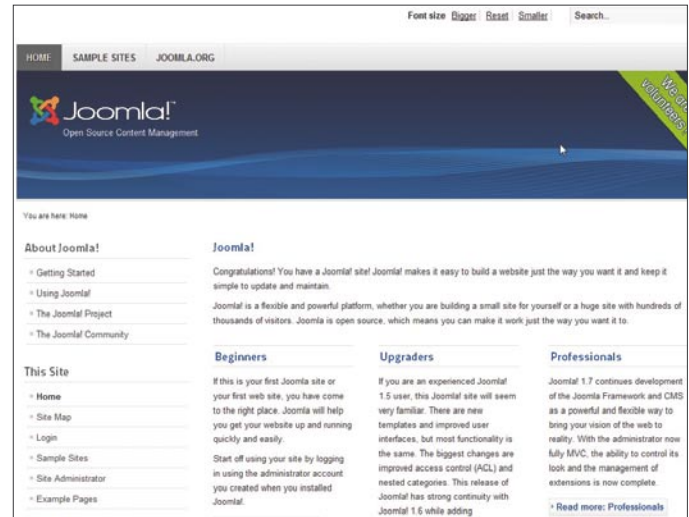


Figure 15. Joomla 1.7 CMS Frontend

Now lets configure some false positive entries of modsecurity to make our server functionable (see Listing 4).

Restart apache to take configurations

```
#/usr/local/etc/rc.d/apache22 restart
```

Listing 3. Downloading and Installing atomicorp mod security rules

```
#wget http://updates.atomicorp.com/channels/rules/delayed/modsec-2.5-free-latest.tar.gz
#tar -zxvf modsec-2.5-free-latest.tar.gz && mv modsec/ /usr/local/etc/apache22/Includes/mod_security2/asl
```

Now we modify asl rules to find our path:

```
#cd /usr/local/etc/apache22/Includes/mod_security2/asl
# find ./ -type f -name '*.conf' | xargs sed -i -e 's/etc\/asl\/usr\/local\/etc\/apache22\/Includes\/mod_security2\/asl/g'
```

Listing 4. Configuring mod security exclusions

```
#cd /usr/local/etc/apache22/Includes/mod_security2/asl
#cat > 99_asl_exclude.conf << EOF
<Directory /usr/local/www/apache22/data/>
SecRuleRemoveByID 960032
SecRuleRemoveByID 960034
SecRuleRemoveByID 960010
</Directory>

<Location /phpmyadmin_5485>
SecRuleRemoveByID 950001
SecRuleRemoveByID 959013
SecRuleRemoveByID 959009
SecRuleRemoveByID 959904
</Location>
EOF
```

Listing 5. Download and untar Joomla 1.7 CMS

```
#cd /usr/local/www/apache22/data
#wget http://joomlancode.org/gf/download/frsrelease/15278/66554/Joomla_1.7.0-Stable-Full_Package.tar.gz
#tar -zxvf Joomla_1.7.0-Stable-Full_Package.tar.gz
```

Listing 6. Testing mod security in Joomla 1.7 CMS using an sql injection

```
http://your domain name or your ip/index.php?action=&type=view&s=&id=-1'%20union%20select%200,concat(char(85),char(15),char(101),char(114),char(110),char(97),char(109),char(101),char(58),name,char(32),char(124),char(124),char(32),char(80),char(97),char(115),char(115),char(119),char(111),char(114),char(100),char(58),pass),0,0,0,0,0,0%20from%20phpdesk_admin/*
```

Note

In order to make your websites function correctly you have to monitor log files for false positive alerts and disable or fix this alerts. You can monitor alerts with command

```
# tail -f /var/log/modsecurity_audit.log | grep id
```

Also if your hardware is old is good to delete some rules or your apache web server will be slow. Example you can delete this files from /usr/local/etc/apache22/Includes/mod_security2/asl directory 10_asl_antimalware.conf

```
10_asl_antimalware_output.conf
11_asl_data_loss.conf
20_asl_useragents.conf
30_asl_antimalware.conf
30_asl_antispam.conf
30_asl_antispam_referrer.conf
```

Method Not Implemented

GET to /index.php not supported.

Figure 16. Error message after sql injection

```
--52a9f80f-F--
HTTP/1.1 501 Method Not Implemented
Allow: TRACE
Content-Length: 221
Connection: close
Content-Type: text/html; charset=iso-8859-1

--52a9f80f-H--
Message: Access denied with code 501 (phase 2). Operator EQ matched 0 at REQUEST_HEADERS. [file "/usr/local/etc/apache22/Includes/mod_security2/base_rules/modsecurity_crs_21_protocol_anomalies.conf"] [line "46"] [id "960015"] [rev "2.0.10"] [msg "Request Missing an Accept Header"] [severity "CRITICAL"] [tag "PROTOCOL_VIOLATION/MISSING_HEADER_ACCEPT"] [tag "WASCTC/WASC-21"] [tag "OWASP_TOP_10/A7"] [tag "PCI/6.5.10"]
Action: Intercepted (phase 2)
Stopwatch: 1313056432309617 418984 (416377 417424 -)
Producer: ModSecurity for Apache/2.5.13 (http://www.modsecurity.org/); core ruleset/2.0.10.
Server: Apache/2.2.19 (FreeBSD) mod_ssl/2.2.19 OpenSSL/0.9.8q PHP/5.3.6 with Suhosin-Patch

--52a9f80f-Z--
```

Figure 17. mod security audit log entry after the sql injection

or whatever you think is not necessary for your website protection.

Testing

Now lets test modsecurity if it is working. In order to test it in real website i am going to install joomla 1.7, a very popular opensource CMS. Installing Joomla CMS (Listing 5). Open web browser and type

```
http://your domain name or your ip/
```

it should open joomla installation follow on screen procedure and finish joomla installation if the dir is not writable by apache in the end it will not create configuration.php file. To do it manually

```
#touch /usr/local/www/apache22/data/configuration.php
#vi /usr/local/www/apache22/data/configuration.php
```

copy from web browser the configuration file and add them to *configuration.php* also click the remove installation folder. If it not succeeded remove from command line

```
#rm -rf /usr/local/www/apache22/data/installation
```

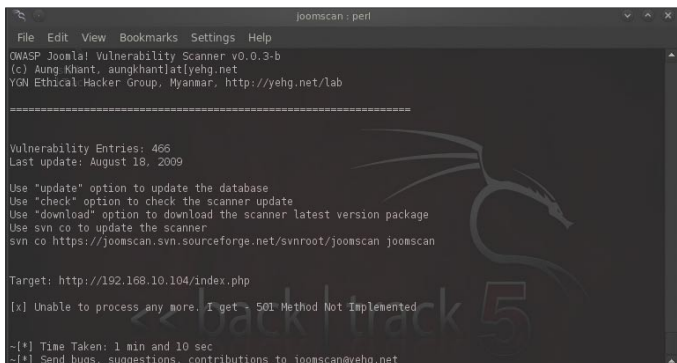


Figure 18. Backtrack joomscan penetration testing utility

If everything is working you will see the picture below if you open your web browser and type

`http://your domain name or your ip/`

Open web browser and type

`http://your domain name or your ip/index.php? login.php
?username=admin'; DROP%20TABLE%20users-`

or another exploit you can test is Listing 6.

If everything is working you will see (Figure 16). And in log file you will see the deny rule (Figure 17).

Also if you try scanning the server for security vulnerabilities using joomscan (can be downloaded from here <http://sourceforge.net/projects/joomscan/> or inside the backtrack dvd) it will return error that it can not process website (Figure 18).

STAVROS N. SHAELES

*Stavros N. Shaeles is a member of the IEEE and the IEEE Computer Society. He received his diploma in Electrical and Computer Engineering in Democritus University of Thrace in 2007. He is working with unix system for 8 years. Currently he is a phd student in research area of data mining with applications to computer security in *nix Systems, under the supervise of Associate Professor Alexandros S. Karakos, and also he is administrator of LPDP Lab in Democritus University of Thrace in Greece (DUTH).*

a d v e r t i s e m e n t

BSD - DAY (2011)

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology

5th November (Saturday)

www.bsdday.eu/2011



MySQL Unleashed!

We explore some tips and tricks that you can use to gain better performance with MySQL

What you will learn...

- How to fine tune and optimize MySQL databases for best performance.

What you should know...

- Working with MySQL, database administration.
-

Your database table seems to be well-indexed and error-proof, yet a simple query on it takes ages to complete. Or may be web apps look good in the dev environment, but become equally bad in the production environment.

If you are a database admin, chances are that you have already encountered above situations at some stage or the other. Therefore, in this article, we shall be looking at debugging, myth-busting and handling certain common (and uncommon) MySQL issues. In this first part, we begin with certain simple and easily implementable tips and tricks.

Storage Engine Woes

If your table uses transactions, you should consider using InnoDB as it comes with full ACID compliance. However, if you do not require transactions, it would be wiser to stick to MyISAM, the default storage engine.

Also, do not try to sail on two boats, er...sorry, storage engines. Consider this: in a transaction, some tables use InnoDB while the rest are on MyISAM. The outcome? The entire subject will be nullified, with only the ones in the transaction being brought back to original state, the rest dumped with committed data. Needless to say, this will lead to inconsistency across the database. However, there exists a simple way to enjoy both the flavours! Most MySQL distributions nowadays include InnoDB,

compiled and linked! But if you opt for MyISAM, you can still download InnoDB separately, and use it as a plugin! Simple, eh?

Counting Issues

If your table employs a storage engine that supports transactions (such as InnoDB), you shouldn't use `COUNT(*)` to find out the total number of rows in the table. The reason being that using `COUNT(*)` on a production class database will at the very most return an approximate value, as at any given time, some transactions will be running. Such incorrect result from `COUNT(*)` will obviously generate bugs if put to use.

The default storage engine for MySQL is MyISAM, which does not support transactions. However, engines such as InnoDB are favored over MyISAM as the latter has a (notorious) distinction of not being the best fault tolerant storage engine. This, in fact, beats the myth that MySQL is faster than PostgreSQL. `COUNT(*)` returns the results quickly in MySQL only when operating under MyISAM. If the storage engine is changed to InnoDB, `COUNT(*)` takes the same amount of time as PostgreSQL.

Test, Test, Test

The major headache with queries is not the fact that no matter how careful one is, something or the other is

bound to be left out and cause a bug later on. Rather, the problem is the timing at which the bug surfaces, which in most cases is after the application/database has gone live. There really exists no sure-shot strategy to counter it, except for the test samples that you must run on your application/database. Any database query cannot be approved unless it is subjected to chunks of thousands of record samples.

Countering Table Scans

More often than not, if MySQL (or any relational database model) has to search or scan for any particular record in a table, a full table scan is used. Again, more often than not, the easiest cure here is to use index tables to solve the problem as full table scans result in poor performance. However, as we shall see in subsequent issues, this does not come without its share of fallacies.

Using Explain

EXPLAIN is an excellent command when it comes to debugging, so let us explore it in depth.

First, let us create a sample table:

```
CREATE TABLE `awesome_bsd` (
  `emp_id` INT(10) NOT NULL
  DEFAULT '0' ,
  `full_name` VARCHAR(100) NOT NULL ,
  `email_id` VARCHAR(100) NOT NULL ,
  `password` VARCHAR(50) NOT NULL ,
  `deleted` TINYINT(4) NOT NULL ,
  PRIMARY KEY (`emp_id`)
)
COLLATE = `utf8_general_ci`
ENGINE = InnoDB
ROW_FORMAT = DEFAULT
```

The table is self-explanatory, with five columns, the last 'deleted' being a Boolean flag to check if an account is active or has been deleted. Next, you may populate this table with sample records (say, 100 employee records). As you can see, the Primary Key lies on 'emp_id'.

So, using the email address and password fields, we can easily create a query to validate or deny a login attempt, as follows:

```
SELECT COUNT (*) FROM awesome_bsd WHERE
email_id = 'blahblah' AND password = 'blahblah'
AND deleted = 0
```

Oops! I've already told you to avoid using COUNT(*). Let me rectify:

```
SELECT emp_id FROM awesome_bsd WHERE
email_id = 'blahblah' AND password = 'blahblah'
AND deleted = 0
```

Now, let us introspect. In the first instance, we queried to locate and return the number of rows where `email_id` and `password` were equal to the given values. In the second case, we did the same but instead decided to ask the value of `emp_id` for all the rows that satisfied the given criterion. What'd you say? Which query is the more expensive?

Apparently, both of them are equally expensive database killing queries because unintentionally, we are querying for a full table scan in each case. To understand better, execute this:

```
EXPLAIN SELECT emp_id FROM awesome_bsd WHERE
email_id = 'blahblah' AND password = 'blahblah'
AND deleted = 0
```

In the output, concentrate on the second-last column, `rows`. Assuming that we had populated the table with 100 records, it will show 100 in the first row, which is the number of rows that MySQL needs to scan in order to evaluate the result of this query. What does this show? Yes, a full table scan (read: memory hog).

To overcome this evil, we need to add indexes.

Indexes

First things first: it's a bad idea to create indexes to every second problem that you might encounter. Excessive indexing leads to slower performances and resource hog. Before going any further, let us create a sample index on our example:

```
ALTER TABLE `awesome_bsd` ADD
  INDEX `LoginValidate` (`email_id`)
```

Next, run the query again:

```
EXPLAIN SELECT emp_id FROM awesome_bsd WHERE
email_id = 'blahblah' AND password = 'blahblah'
AND deleted = 0
```

Now notice the value. Instead of 100, it should now say 1. Thus, MySQL is now scanning only 1 row in order to give you the output of this query, thanks to the earlier created index. You might notice, the index created is only for the email address field while the query searches for other fields too. This shows that MySQL first performs a cross-check to see if any of the values specified in the WHERE

clause has indexes defined for it, and if so, performs accordingly. However, it isn't that every iteration will be reduced to one. If, for instance, the indexed field is not unique (such as employee names, which can have identical values in two rows), there will be multiple records left even after indexing. Yet, it will still be better off than full table scan.

Also, the order of columns specified in the WHERE clause does not play a role in the process. If, for instance, in the above query, you reverse the order of fields such that email address comes last, MySQL will still iterate on the basis of the indexed column.

Now, with indexing at your finger tips, you've noticed how to avoid numerous full table scans and gain better results. Lets proceed further.

Full Table Scans Can Strike Back, Too

First up, coming to common MySQL errors or issues that are often ignored. Lets create a table along the lines of the following sample (this sample table has few flaws in it, as we shall see later on):

```
CREATE TABLE 'awesome_table' (
  'awe_a' INT(10) NOT NULL AUTO_INCREMENT,
  awe_date' DATE NOT NULL,
  PRIMARY KEY ('awe_a'),
  INDEX 'awe_date' ('awe_date')
)
```

Additionally, you may suffix the following to the above table too (it depends on the environment you have at your disposal, though the following code is a recommended addition, if possible):

```
COLLATE = 'utf8_general_ci'
ROW_FORMAT = DEFAULT
```

Populate the table with some sample records (say, 10 records). The Primary Key lies with the `awe_a` column, while the `awe_date` column is indexed as well. So, simply because indexing is on for `awe_date` column, we can assume that any queries done on the column will not run unoptimised, right? Apparently, not! Run the following sample query:

```
EXPLAIN SELECT * FROM awesome_table
WHERE awe_date < '1980'
```

What did you get? Correct! It runs a full table scan, yet again, in spite of the index placed on the `awe_date` column. Now, let us modify the above query slightly:

```
EXPLAIN SELECT * FROM awesome_table
WHERE awe_date < '1980'-01-02'
```

What did you see now? It no longer performs a full table scan, but instead, shows the scan type as `range` rather than `index`. The outcome? Faster processing! As you must have noticed, in the first query, '1980' is an ambiguous parameter but in the second query the entire date eliminates the possibility of a scan type such as `ALL`.

Another common scenario wherein an otherwise-not-required full table scan is called upon is one comprising of `UCASE` and `LCASE`. More often than not, applications perform case-insensitive searches. For example:

```
EXPLAIN SELECT * FROM table-name
WHERE UCASE (column-name) = 'THIS IS SO WONDERFUL' ;
```

In such searches, MySQL will ignore the indexes, convert the values held by the specified column in each row to `UCASE` and then perform the search for the given sample text. The easiest way out of such a situation is to store either `UCASE` or `LCASE` values (the requisite case conversion should ideally be performed when the record is inserted in the table). Following that, the case of the value under consideration can be automatically compared, as shown in the query below:

```
EXPLAIN SELECT * FROM table-name
WHERE column-name = UCASE ('this is so wonderful') ;
```

This shall compel MySQL to convert the given value into `UCASE` in order to match a rule that allows for storage of only `UCASE` values in the given column.

The Myisam Storage Engine – a Closer Look

As we covered earlier, MyISAM is MySQL's default storage engine. Now we shall take a closer look at it.

MyISAM by default stores a table in two files (one for the data, the other for indexes). For the data file, the extension is `.MYD` while for the index file, the extension is `.MYI`. You can also use the `DATA DIRECTORY` and `INDEX DIRECTORY` options along with the `CREATE TABLE` command to specify the location of each file of the given table. Since these files are platform independent, most databases support specifying of the directories.

Also, all readers having `SELECT` associated with queries need to obtain read locks and multiple users can do the same by means of shared locks. However, on the contrary, all writers need to have exclusive locks.

Visit our website

You will find here:

- ➡ materials for articles-listings, additional documentation, tools
- ➡ the most interesting articles to download
- ➡ current information on the upcoming issue

Thus, while two users can acquire read locks (SELECT) at the same time, they cannot perform write operations (INSERT, UPDATE or DELETE, etc.) simultaneously. This is the precise reason why indexing is crucial and needless to say, if you fail to handle your indexes well, write operations will become slow and time consuming resulting in heavy load on the system.

MyISAM supports Full Text Index (also known as Full Text Search) but doesn't yet support transactions. Table locks are a possibility, but row locks are not. Further more, MyISAM supports compressed tables too (read only). However, it must be noted that only individual rows are compressed and not the entire table as a whole.

MyISAM also has the advantage of specifying NULL values even in indexed fields, as well as providing a different character set for each CHAR or VARCHAR column type.

MyISAM and B-TREE – Spicing Up Your Indexes!

In a MyISAM powered table, the type of each index is B-tree. So before going any further, let us analyze what a B-tree is, and to do so, we shall turn to Wikipedia (<http://en.wikipedia.org/wiki/B-tree>):

“... a B-tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic amortized time. The B-tree is a generalization of a binary search tree in that a node can have more than two children. ...Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. It is commonly used in databases and filesystems.”

With the introduction out of the way, we now turn our attention once again to MyISAM and B-tree, with special focus on indexes. First, we can briefly sum up the theoretical aspect of the issue.

It can be said that the B-tree index has a root node on the top (since it is a tree, it has to have a root). In B-tree, any node that doesn't have a child attached to it is called a leaf node. Therefore, the root node is a non-leaf node while all the nodes that spring from it are called leaf nodes. The links between a node and its immediate children can be shown as *pointers*. Do not confuse the pointers to be C/C++ pointers.

Going below the leaf nodes (ones without children nodes), you'll find the actual table data. The data is linked to the leaf nodes on the basis of *key values*. Thus, it becomes quite obvious that effective and speedy searches depend on how the key values associate the data to the leaf nodes, or, in simple terms, how effectively a table is indexed.

At this junction, we can also tear apart the myth that MyISAM supports clustered indexes. Truth is, MyISAM does not store data in a sorted fashion, whereas for a

clustered index to work, data must be sorted. MyISAM, on the other hand, stores data as and when it is inserted into the table. It does sort the indexes, but as we have already covered, indexes are stored in a separate file (.MYI) than data itself (.MYD). MyISAM uses indexes to point to the exact location of unsorted data and as a result, removes the need of data storage in a sorted manner. Bottom line is that clustered indexes are not possible on MyISAM.

The most obvious benefit of employing a B-tree is that it considerably improves the search functionality (*SELECT* queries). However, on the down side, queries such as *INSERT* and *DELETE* tend to become slower as each time a record is either inserted or deleted, the indexes located in .MYI file also need to be modified. The cure in such a case is to index selectively.

Index selectivity implies the difference in values stored or recorded in the columns of a table. Selectivity is measured on a scale of 0 or 1, wherein 1 implies that each value in the selected column is *UNIQUE*. Generally, selectivity of 1 occurs with columns that are *UNIQUE* or *PRIMARY KEY*, though this isn't always the case and it varies with the nature of values stored in the given columns. For the sake of simplicity, we can stick to the following formula:

$$\text{SELECTIVITY} = \text{NO. OF DISTINCT RECORDS} / \text{TOTAL NO. OF RECORDS}$$

The above formula is a stripped down and simplified version for the purpose of understanding. If you so desire, you can use the alternate way to calculate selectivity by employing a production class database and finding the number of *DISTINCT* rows in it. Bear in mind though, that the number of *DISTINCT* values in a column may or may not always work perfectly.

Higher selectivity means the operations shall be of shorter duration and vice-versa. As a result, lower

selectivity is termed as an expensive operation while higher selectivity is an inexpensive operation.

Finally, coming back to the sample table that we created at the start of the article. The `awe_a` column is a *PRIMARY KEY*, and will thus have a selectivity of 1. the `awe_date` column is indexed as well, so lets focus on it. Quite obviously, all dates cannot be distinct or *UNIQUE* and this column is bound to have a low selectivity. In such a case, it will not serve as a good index and as a result, in spite of indexing the column, we got a full table scan in the first query that we ran earlier.

Before performing a query, MySQL calculates the *cost* of the different ways in which the query can be performed and then picks the cheapest or most effective way. So if a low selectivity column is used for an index, it will overload the system. To avoid such overloading, MySQL may choose not to use your index if the selectivity is low. This is precisely the reason why even after using multiple indexes, your queries may still result in full table scans (read: slower outputs) and burden the system resources. In simple terms, the entire input and output process depends on the appropriateness of the indexing and querying. Hence, it becomes vital that indexes are used judiciously and selectively.

In this article, we covered the myths and overlooked or relatively lesser known details about MySQL indexes, as well as the functioning of the MyISAM storage engine. I hope you enjoyed reading it. Happy querying!

SUFYAN BIN UZAYR

Sufyan bin Uzayr is a 20-year old freelance writer, graphic artist, programmer and photographer based in India. He writes for several print magazines as well as technology blogs. His prime areas of interest include open source, mobile development, web CMS and vector art. He is also the Founder and Editor-in-Chief of <http://www.bravenewworld.in> He can be reached at <http://www.sufyan.co.nr>

iXsystems Announces FreeNAS™ 8.0.1 RC1

The FreeNAS™ Team has announced the first Release Candidate for the FreeNAS™ 8.0.1 branch. FreeNAS™ 8.0.1 represents a major leap in functionality and stability for FreeNAS™ 8. Features added to FreeNAS™ in the 8.0.1 branch include S.M.A.R.T. and UPS services, USB 3.0 support, and OSX Lion AFP compatibility. In addition, cronjob support and rsync have been added to the GUI, and replication has been improved for increased data integrity.

Once FreeNAS™ 8.0.1 is officially released, development focus will shift to 8.1, which will add a plug-in system to support services such as BitTorrent and UPNP.



FreeBSD
Mall

**Your FreeBSD &
PC-BSD Resource**

www.FreeBSDMall.com



FreeBSD 8.2 Jewel Case CD/DVD

Set contains:

- **Disc 1:** Installation Boot (i386)
- **Disc 2:** LiveFS (i386)
- **Disc 3:** Essential Packages (i386)
- **Disc 4:** Essential Packages (i386)

FreeBSD 8.2 CD	\$39.95
FreeBSD 8.2 DVD	\$39.95
FreeBSD 7.4 CDROM	\$39.95
FreeBSD 7.4 DVD	\$39.95

FreeBSD Subscriptions

Save time and \$\$\$ by subscribing to regular updates of FreeBSD!

FreeBSD Subscription , start with CD 8.2	\$29.95
FreeBSD Subscription, start with DVD 8.2	\$29.95
FreeBSD Subscription, CD 7.4	\$29.95
FreeBSD Subscription, DVD 7.4	\$29.95

PC-BSD 8.2 DVD (Hubble Edition)

PC-BSD 8.2 DVD	\$29.95
PC-BSD Subscription	\$19.95

BSD Magazine

BSD Magazine	\$11.99
--------------------	----------------

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide)	\$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)	\$39.95
★ Special: The FreeBSD Handbook, Volume 2 (Both Volumes)	\$59.95
★ Special: The FreeBSD Handbook, Both Volumes, & FreeBSD 8.2	\$79.95

The FreeBSD Bundle

Inside the Bundle, you'll find:

- FreeBSD Handbook, 3rd Edition, Users Guide
- FreeBSD Handbook, 3rd Edition, Admin Guide
- FreeBSD 8.2 4-disc set
- FreeBSD Toolkit DVD

★ Special: The FreeBSD CD Bundle	\$99.95
★ Special: The FreeBSD DVD Bundle	\$99.95

The FreeBSD Toolkit DVD **\$39.95**

FreeBSD Mousepad **\$10.00**

FreeBSD Caps **\$20.00**

PC-BSD Caps **\$20.00**

For **MORE** FreeBSD & PC-BSD items, visit our website at FreeBSDMall.com!

CALL 925.240.6652 Ask about our software bundles!

**NEW
APPAREL!**



Terminal Descriptions

for OpenBSD AMD/Intel consoles

In this article I would like to describe the results of my work of tuning OpenBSD consoles for AMD/Intel PCs. These results are also applicable to computers with the same hardware architecture (amd64 or i386, see <http://www.openbsd.org/plat.html>): servers, workstations, notebooks, etc.

What you will learn...

- important facts about ASCII terminals
- how to tune OpenBSD AMD/Intel consoles for comfortable work with mail and Midnight Commander

What you should know...

- what is OpenBSD
- how to install OpenBSD operating system
- how to use OpenBSD packages and ports

I often worked on OpenBSD AMD/Intel PC consoles and really did not have good support of navigation and function keys of a typical PC keyboard. Also I had some problems with colors/attributes (maybe they

were videoadapter dependent). These issues exist long time and cause much inconveniences if you often work at console (not in a graphical environment!).

Listing 1. Cyrillic support for AMD/Intel consoles

```
# cat /etc/kbdtype
ru
# cat /etc/rc.local
...
if [ -x /usr/sbin/wsconscfg -a -x /usr/sbin/wsfontload ]; then
    /usr/sbin/wsfontload -h 16 /usr/share/misc/pcvtfonts/koi8-r-8x16

    for CONSOLE in 2 3 ; do
        /usr/sbin/wsconscfg -d -F ${CONSOLE}
        /usr/sbin/wsconscfg -t 80x25bf -e vt100 ${CONSOLE}
    done

    /sbin/wsconsctl -w keyboard.map+="keycode 184 = Mode_Lock" >/dev/null
fi
...
#
```

Listing 2. Terminal descriptions patch for AMD/Intel consoles

Apply this patch by doing:

```
cd /usr/src
patch -p0 <OpenBSD_PC_console.patch
```

And then rebuild and install the terminal description databases:

```
cd share/termtypes
make obj
make cleandir
make depend
make
make install
```

After that you can use these emulations **for** AMD/Intel PC consoles:

```
- pccon0-m
- pccon0
- pccon-m
- pccon
```

Also you can replace default "vt220" to "pccon" in /etc/ttys **for** "console" & "ttyC*" entries.

```
=====
--- ./share/termtypes/termtypes.master.orig      Mon
          Nov 29 23:55:07 2010
+++ ./share/termtypes/termtypes.master Sun Aug 14 18:33:
          19 2011
@@ -1649,6 +1649,55 @@
   qansi-w|QNX ansi for windows,
   xvpa, use=qansi-m,

+#### OpenBSD consoles
+#
+# From: Alexei Malinin <Alexei.Malinin@mail.ru>; July,
+      2011.
+#
+# The following terminal descriptions for the AMD/
+      Intel PC console
+# were prepared based on information contained in the
+      OpenBSD-4.9
+# termtypes.master and wscons(4) & vga(4) manuals
+      (2010, November).
+#
+pccon+keys|OpenBSD PC keyboard keys,
+  kbs=^H, kcan=^C, kclr=^L, kcub1=\E[D, kud1=\E[B,
```

```
      kcufl=\E[C,
+  kcuul=\E[A, kdchl=\E[3~, kend=\E[8~, kent=^M, kfl=\E[11~,
+  kf10=\E[21~, kf11=\E[23~, kf12=\E[24~, kf2=\E[12~,
+  kf3=\E[13~, kf4=\E[14~, kf5=\E[15~, kf6=\E[17~, kf7=\E[18~,
+  kf8=\E[19~, kf9=\E[20~, khome=\E[7~, kichl=\E[2~,
+  knp=\E[6~, kpp=\E[5~, krfr=^R, kspd=^Z,
+pccon+acs0|simple ASCII pseudographics for OpenBSD PC
      console,
+  acsc=+>\,<-\^.\v0#'a\:\f\h#j+k+l+m+n+o~p-q-r-s_
      t+u+v+w+x!!|#~o,
+pccon+acs|default ASCII pseudographics for OpenBSD PC
      console,
+  acsc=++\,<,\,--..00''aaffgghhijjkkllmmnooppqqrrssttuu
      vvwxxxyzzz{{{||}}~},
+pccon+colors|ANSI colors for OpenBSD PC console,
+  bce,

+  op=\E[m$<2>, setab=\E[4%p1%dm$<2>,
+  setaf=\E[3%p1%dm$<2>,
+pccon+base|base capabilities for OpenBSD PC console,
+  am, km, mc5i, npc, nxon, xenl, xon,
+  cols#80, it#8, lines#24,
+  bel=^G, clear=\E[H\E[2J$<50>, cr=^M,
+  cup=\E[%i%p1%d;%p2%dH$<5>, ed=\E[J$<50>, el=\E[K$<3>,
+  ell=\E[1K$<3>, enacs=\E(B\E)0, home=\E[H$<5>, ht=^I,
+  ind=^J, nel=\EE$<2>, rev=\E[7m$<2>, ri=\EM$<5>, rmacs=^O,
+  rmso=\E[m$<2>, rs2=\Ec$<50>,
+  sgr=\E[m$<2>?%p1%p3%|t\E[7m$<2>%?%p9%t\016%e\017%;,
+  sgr0=\E[m$<2>\017, smacs=^N, smso=\E[7m$<2>,
+pccon0-m|OpenBSD PC console without colors & with
      simple ASCII pseudographics,
+  use=pccon+base,
+  use=pccon+acs0,
+  use=pccon+keys,
+pccon0|OpenBSD PC console with simple ASCII pseudographics,
+  use=pccon0-m,
+  use=pccon+colors,
+pccon-m|OpenBSD PC console without colors,
+  use=pccon+base,
+  use=pccon+acs,
+  use=pccon+keys,
+pccon|OpenBSD PC console,
+  use=pccon-m,
+  use=pccon+colors,
+
+#### NetBSD consoles
+#
+# pcvt termcap database entries (corresponding to release 3.31)
```

Note

Commands and options discussed in this article refer to the latest version of OpenBSD – 4.9.

Let us look at my typical work environment:

- an AMD/Intel PC with VGA display,
- PC keyboard (usually 104-key with cyrillic letters),
- `vt220` default console terminal type,
- cyrillic support for the 2 and 3 consoles (`Ctrl+Alt+F3` and `Ctrl+Alt+F4`), fragments for configuration files (in my `/etc` catalog) which differ from defaults are on Listing 1.

Note

Useful links about OpenBSD cyrillization:

- <http://www.obsd.ru/8/?q=node/1172>
- <http://www.openbsd.ru/docs/howto-cyrillic.html>
- <http://www.openbsd.org/faq/faq7.html>

The console environment described above is suitable for mail and Midnight Commander but not all navigation and function keys work as expected, some color/attribute issues are annoying.

Note

Midnight Commander (<http://www.midnight-commander.org/>) is a handy full-screen file manager but it is not in the base OpenBSD distribution. It can be installed from packages or ports (`ports/misc/mc`), see <http://www.openbsd.org/faq/faq15.html>.

Before delving into details of tuning the console let us recall how full-screen applications interact with ASCII (or alphanumeric) terminals. These applications typically use high-level screen management library. In turn this library uses a terminal descriptions database for performing high-level screen management functions (cursor movement, setting colors, etc). The most famous screen management library for ASCII terminals is `curses` which uses one of the two terminal descriptions databases: `termcap` or `terminfo`. These terminal description databases make `curses` terminal independent, and the terminal independence is the foundation of `curses`. `termcap` and `terminfo` are the mechanisms by which UNIX systems support hundreds of varieties of ASCII terminals without the need for special drivers for each terminal. Most of the capabilities in `termcap` and `terminfo` are identical except in name.

Listing 3. Tuning display resolutions for AMD/Intel consoles

```
...
if [ -x /usr/sbin/wsconscfg -a -x /usr/sbin/wsfontload ]; then
    /usr/sbin/wsfontload -h 8 /usr/share/misc/pcvtfonts/koi8-r-8x08
...

and

...
if [ -x /usr/sbin/wsconscfg -a -x /usr/sbin/wsfontload ]; then
    /usr/sbin/wsfontload -h 10 /usr/share/misc/pcvtfonts/koi8-r-8x10
...

and

...
if [ -x /usr/sbin/wsconscfg -a -x /usr/sbin/wsfontload ]; then
    /usr/sbin/wsfontload -h 16 /usr/share/misc/pcvtfonts/koi8-r-8x16
...

```

Note

Important OpenBSD manual pages about ASCII terminals:

- `tty` (5) – terminal initialization information
- `wscnscfg` (8) – configure virtual terminals on a `wscn` display
- `wscn` (4) – console access
- `vga` (4) – VGA graphics driver for `wscn`
- `stty` (1) – set the options for a terminal device interface
- `tset` (1) – terminal initialization
- `tput` (1) – terminal capability interface
- `termcap` (5) – terminal capability database
- `terminfo` (5) – terminal capability database

So, the problem to be solved is that `vt220` terminal type is not well suited for the AMD/Intel PC console.

What could I do?.. In the OpenBSD terminal descriptions database (I used the text version of `termcap` – `/usr/share/misc/termcap`) I found descriptions for NetBSD, FreeBSD, Linux (and for many others operating systems) consoles but nothing suitable for the OpenBSD AMD/Intel PC console! So the only solution would be to prepare a complete and correct terminal description for this console... I read OpenBSD manual pages and many others information sources that might be relevant to ASCII terminals, `curses`, `vt100`, `vt220`, `xterm`, ANSI, etc...

Note

The best source of information I ever read is the book “termcap & terminfo” published by O’Reilly in 1988 (<http://oreilly.com/catalog/9780937175224/>).

At last I prepared several terminal descriptions for the AMD/Intel PC console. The patch against OpenBSD-4.9 sources is on Listing 2. Do not forget to read the comments at the beginning of the patch!

Note

This patch can be downloaded from here: http://am1225.narod.ru/software/OpenBSD_PC_console.patch.

Note

The OpenBSD FAQ describes how to build the operating system from sources: <http://www.openbsd.org/faq/faq5.html>.

After patching OpenBSD it will be possible to use several terminal types for AMD/Intel consoles:

- `pccon` is suitable for color display with 80x25 resolution,

- `pccon-m` is suitable for black and white display with 80x25 resolution,
- `pccon0` is suitable for color display with 80x40 and 80x50 resolutions,
- `pccon0-m` is suitable for black and white display with 80x40 and 80x50 resolutions.

There are no pseudographics for 80x40 and 80x50 display resolutions, so I prepared separate terminal descriptions `pccon0` and `pccon0-m` for these cases.

Note

To set up resolutions it is necessary to use the appropriate font:

- `/usr/share/misc/pcvtfonts/koi8-r-8x08` for 80x50 resolution,
- `/usr/share/misc/pcvtfonts/koi8-r-8x10` for 80x40 resolution,
- `/usr/share/misc/pcvtfonts/koi8-r-8x16` for 80x25 resolution.

The appropriate fragments of `/etc/rc.local` are on Listing 3.

Note

To eliminate some color/attribute issues I usually run Midnight Commander as follows:

```
# mc -c --colors errdhotnormal=black,lightgray:menuhotsel=
lightgray,black
```

That is all I have to tell about my work. Also I hope that the OpenBSD developers will find these terminal descriptions helpful and include them into the base OpenBSD distribution as the default configuration for the AMD/Intel console.

ALEXEI MALININ

Alexei graduated with a degree from applied mathematics. His adventure with UNIX started in 1990, and he works as a system/network administrator since 1991. He is an OpenBSD fan since version 2.2.

Alexei.Malinin@inetcomm.ru

(Ab)using VideoLAN

Learn what you can do with your video and audio using powerful VideoLAN command line interface

Dealing with video and audio data is part of our everyday life. Sometimes, though, we need to do things that fall into „advanced“ category. What tools should we use then?

What you will learn...

- That VideoLAN is a full-featured multimedia framework
- That you can combine VideoLAN's modules into powerful pipelines
- How to use VideoLAN in four real-life scenarios

What you should know...

- How to use command line
- Core networking concepts
- Core video/audio concepts

A number of multimedia-related solutions are present in open-source world right now. Among the most popular and ubiquitous are MPlayer and VideoLAN.

They share a fair amount of the codebase (both use ffmpeg), but have somewhat different design. MPlayer is famous for *having a command line option for everything*. It has rich functionality and you can enable or disable certain features using command line flags. Still, if you need to do something that MPlayer developers didn't expect you to need, you're in trouble.

VideoLAN's design (at least from user perspective) is quite different. It's not just a player – it's a full-featured multimedia framework, like GStreamer or DirectShow. Although it has rather simplistic user interface, you have a total control over VideoLAN via the command line. You can build pipelines of filters and pass them as command line arguments. Unlike MPlayer, which can only play (you have to use MEncoder to encode data), VideoLAN can do any crazy thing you want with your video or audio.

VideoLAN's problem though, is that this incredible flexibility isn't that well documented (though situation is improving continuously). Let me share some examples of what VideoLAN can do:

Scenario 1

My desktop FreeBSD machine is connected to my stereo and I use it for music. But it does not have a display, which

makes watching movies on it, er... problematic. So what I want is to be able to watch the movie on my laptop while redirecting the audio to my desktop machine.

Let's start a VideoLAN that will listen to the UDP socket on port 1234 and play everything that it receives.

```
vlc udp://
```

Command that looks like `vlc [smth]` tells VideoLAN to open something. In this case it's a UDP socket. VideoLAN uses port number 1234 by default.

Now what we need is to start playing video on the laptop. We don't need any sound there, instead we want audio to be streamed to a desktop machine. Also we don't want to stream video to desktop machine – all we care about there is sound. Let's try the following command:

```
vlc some_movie.avi --sout="#duplicate{dst=display{noaudio,
delay=1250},dst=duplicate{dst=std{mux=ts,access=udp,dst=
192.168.1.42:1234},select=\"novideo\"}}}"
```

Here we build a full fledged pipeline. `duplicate` module dispatches stream to a multitude of nested modules (modules' chains specified with `dst=`).

`display` is a module that, surprisingly enough, displays the stream on the current screen. It also plays sound on the local audio subsystem. But we disable it with `noaudio`

parameter, as we need only video on the laptop. Also we use `delay=1250` – this is the default buffering time used by VideoLAN when transmitting and receiving data over the network. In order for picture and sound to be in sync, we need to delay picture a bit – so that we have enough time to buffer sound.

Second destination point of `duplicate` module is another `duplicate` module. We need it to specify `select=novideo` option, which will prevent video from being sent to `std`. `std` stands for *standard* – it's a standard sink for the data. In this particular case it sends the audio via udp to 192.168.1.42:1234. As we need to send a stream in some format, we specify `mux=ts` which means MPEG-TS – MPEG container format specifically designed to be used in networking environment.

Now we have everything settled and you should hear the sound coming from the desktop machine and still see a perfectly synchronized video on your laptop.

Scenario 2

(a bit weird, but nice for demonstration). I have 2 laptops and I want to split the movie between them – i.e. to use their screens as one large screen. The laptops should stand next to each other, the left one should show the left half of the picture and the right one – the right part.

Not everything in VideoLAN can be tuned in the pipeline command line argument (`--sout=...`). It also has a number of general-purpose command line arguments. For example – `--crop`, which tells VideoLAN how to crop a picture that is displayed locally.

Let's assume that our movie's size is 720x304.

In order to fulfill the scenario, we need VideoLAN running as UDP server on one of the laptops. We'll receive the full picture here and will have to crop it in order to show only the right half.

```
vlc udp:// --crop='309x303+310+0'
```

`crop` argument tells VideoLAN to use picture of the width 309 and height 303 with the offset 310x0 pixels from the top left corner of the original picture.

Let's execute the following on the second laptop:

```
vlc some_movie.avi --sout="#duplicate{dst=display{delay=1250},dst=duplicate{dst=std{mux=ts,access=udp,dst=192.168.1.42:1234},select=\"noaudio\"}}\" --crop='309x303+0+0'
```

Similar to the previous scenario, we display the video on local display with a delay of 1250 milliseconds. `crop` argument tells VideoLAN to crop the picture to size 309x303, which effectively shows us only the left part of

If you wish to contribute to BSD magazine, share your knowledge and skills with other BSD users – do not hesitate – read the guidelines on our website and email us your idea for an article.

Join our team!



Become BSD magazine Author or Betatester

As a betatester you can decide on the contents and the form of our quarterly. It can be you who read the articles before everybody else and suggest the changes to the author.

Contact us:
editors@bsdmag.org
www.bsdmag.org

the picture. We also stream video stream (without audio data) to our UDP server host to display the second half of the picture.

Now, if we run VideoLAN using the commands above on 2 laptops, we'll see half of the picture on each laptop with audio being played only by the second one.

Scenario 3

I am not at home and I want to use my laptop to watch a DVD movie stored on the hard drive of my desktop machine.

The idea is that you may have a low-bandwidth connection that will make raw DVD data streaming impossible. Therefore what we need to do is to transcode datastream on the fly. It's really not that hard. Let's start with the VideoLAN on the home machine.

```
vlc dvd:///home/user/saved/dvd --sout=#transcode{vcodec=h264,
vb=1024,deinterlace,acodec=mp4a,ab=96,channels=2}:
std{access=http,mux=asf,dst=10.0.0.1:10005}
```

Couple of points here. First, we play DVD that is stored on disk – so we use a special syntax for that. Then, we use `--sout` argument to build our pipeline. `transcode` module is the VideoLAN's swiss army knife for all kinds of stream transformations.

Most of the options specified in the example are self-explanatory, so let's cover them just briefly:

- `vcodec` – what video codec to use for transcoding. VideoLAN has implementations of practically all codecs that exist at this moment. We use h264 as one of the most effective.
- `vb` – stands for video bitrate. As our bandwidth is limited, we limit the bitrate to 1MBit.
- `deinterlace` – means that we want the picture to be deinterlaced prior to transcoding.
- `acodec` – what audio codec to use. We use mp4a as one of the most effective ones.
- `ab` – audio bitrate
- `channels` – number of audio channels in audio streams that we want to have. It's reasonable to downmix audio to 2 channels when transmitting data over the network.

Another important moment is that we use `access=http` (instead of `access=udp` in previous examples). With `access=udp`, VideoLAN pushes the stream to the desired address. With `access=http` it acts as a server by itself.

That's why on our remote machine we'll have to use the following command line:

```
vlc http://10.0.2.1:10005
```

The above command will connect to the VideoLAN started on the home machine and will stream transcoded data from it.

Scenario 4

Transcode the movie to be played on iPhone.

This is fairly common. There are tons of tools that can do this. Still it's worth pointing out that you can also use VideoLAN for iPhone-targeted transcoding. here's the command line you need:

```
vlc in.avi --sout="#transcode{width=320,canvas-height=240,
vcodec=mp4v,vb=768,acodec=mp4a,ab=96,channels=1,audio-
sync}:std{access=file,mux=mp4,dst="out.mp4\"}"
```

This example is also fairly straight-forward. However, we use some new options here:

- `width` – resize the video to have a given width
- `canvas-height` – note that we use it and not just `height`. When you use `canvas-height`, If the video can't be resized to a given height without changing its aspect ratio, it will be padded with black stripes.
- `audio-sync` – it will insert additional frames or drop some frames in order for video and audio to be perfectly synced. Useful to avoid potential synchronization problems.

It's also worth noting that we use `access=file` as our output and MP4 as container format.

If we run the above command, VideoLAN will start converting the stream as fast as possible – this is because we haven't specified `display` in our pipeline – so VideoLAN can process video faster than in realtime.

Four scenarios described above show the power of VideoLAN's video and audio processing abilities. However, VideoLAN can do a lot more. For example, it has pluggable interfaces system, which allows you to control VideoLAN via text input, window UI, infrared remote controller, telnet, irc and so on. But this is probably a different topic that will be covered in one of the next issues.

MICHAEL BUSHKOV

Michael Bushkov is an active FreeBSD user and former committer. He is one of the main contributors of FreeBSD's nsswitch caching daemon (nscd) implementation.

EuroBSDcon

2011



The **Anniversary**

BSD Daemon © Marshall Kirk McKusick. Used with permission. <http://www.mckusick.com/copyright.html>

6 until 9 October, 2011
Meeting Plaza, Maarssen

Address: Planetenbaan 100
3606 AK Maarssen
The Netherlands

GPS: N52.12840, E5.0360

10th European BSD Conference

<http://2011.eurobsdcon.org/>



NetBSD Intrusion Detection Server

How can we describe the functions of such a server?

Sometimes special type of systems are needed to be running on the server. This server will serve different purposes, it will take care of the network security.

What you will learn...

- How to run snort Intrusion Detection System on your machine.
- If you have previously bad experience with hackers, intruders, now you have the opportunity to detect such intruders.
- What an IDS is and how it works.

What you should know...

- What a NetBSD is. A basic knowledge of BSD operating system is required.
- To have bad experience with hackers, intruders.
- A basic knowledge of networks.

Since most of the people around the world can not buy super-duper highly expensive IDS (Intrusion Detection System) machines, I will show you how to prepare such a custom made machine with a usual server. We all need IDS machines put in our networks. The world, and the internet, have become more hostile and sometimes the company's security depends highly on the IDS that is silently processing packets somewhere in the network.

The Intrusion Detection System shortly called IDS is a software system designed to help you to detect attempts of accessing computer systems through a network. The IDS can help us to detect any unusual network activity and can alert us about that. The system cannot directly detect attacks within properly encrypted traffic but with appropriate rules you can have a wider picture of what is going to happen in your network or machines. So, the better are the rules that the system use the better are the detection results. And let's do not forget that hackers become more innovative after every attempt.

An intrusion detection system is used to detect several types of malicious behaviors that can compromise the security and trust of a computer system. These types of behaviors include network attacks against vulnerable services or host based attacks that aims to take control of your machines. You and your machines as well as all your equipment are targets because most of the hackers want

to gain access to what you have. In order to achieve that they may try many ways, such as unauthorized logins and access to sensitive files, or using of viruses, trojan horses, and worms.

An IDS can be composed of several components: Sensors which generate security events, a Console to monitor events and alerts and control the sensors, and a central Engine that records events logged by the sensors. Also IDS can use several output engines like database, log files, pipes or network sockets. Everyone of the output engines is useful and has its own benefits. These output engines can also affect the performance of the system. Of course, it is not the same to log to a local file and to log to a central database server. And it is not the same to log to a structured local file and plain text file.

The Operating system of our choice – NetBSD

The NetBSD is primarily focused on high quality design, stability and performance of the system. I prefer to use NetBSD because at first: I am a fan and second: I am an enthusiast. But one of the main reasons is that I have some small experience with other types of operating systems and I know why to use NetBSD. NetBSD is very fast and does not need a machine for 100 000 euros just to make packet inspection. Some people probably prefer FreeBSD or OpenBSD, but I think that NetBSD is perfect for that kind of work.

The Intrusion Detection System of our choice

– Snort

Snort is a free and open source *network intrusion prevention system* (NIPS) and *network intrusion detection system* (NIDS) capable of performing packet logging and real-time traffic analysis on IP networks.

Snort can perform various ways to analyze and detect hacking activity. Some of these ways are protocol analysis, content searching/matching, also it is used to block and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, web application attacks, SMB probes, or OS fingerprinting attempts. The software is mostly used for intrusion prevention purposes, by dropping attacks as they are taking place.

There are several running methods that are available in Snort. It can be configured to run in the following modes:

Sniffer mode. In this mode, Snort simply reads the packets off of the network and displays them for you on the console.

Packet Logger mode, which logs the packets to disk.

Network Intrusion Detection System (NIDS) mode. You have complex configuration options, that allow Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

I do not have the intention to describe all the aspects of the network security and probably you do not wish that, the thing that is my intention to show you is how to implement Snort in you NetBSD system. So, I intend to show you the things as they are based on my experience. On every documentation in the internet you can find *dry* documentation how to use Snort, what its options are and what that options mean but there are rare information from the real life. And my efforts are mostly focused on this.

An intrusion detection system like Snort is a perfect tool to protect you but it should be used properly to take maximum effect. I would remark that such a system is especially in benefit when is used in combination with optimized and highly effective operating system like NetBSD. We all know that NetBSD is preferred choice for servers with requirement for high reliability. Especially in firewalls, gateways or border machines accessible by internet. I would like to say that I prefer to use Snort for one more thing. The case where I have to protect specific services against bug exploitation. Maybe for many people is strange how such a system could be used to protect services from their own bugs to be exploited, but it is possible. Let me show you a real life example from my personal experience.

Listing 1. Installation of Snort

```
# pkg_add snort
snort-2.8.5.1: Creating group 'snort'
snort-2.8.5.1: Creating user 'snort'
useradd: Warning: home directory '/nonexistent' doesn't exist, and -m was not specified
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/classification.config to /usr/pkg/etc/snort/classification.config
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/gen-msg.map to /usr/pkg/etc/snort/gen-msg.map
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/generators to /usr/pkg/etc/snort/generators
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/reference.config to /usr/pkg/etc/snort/reference.config
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/sid-msg.map to /usr/pkg/etc/snort/sid-msg.map
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/snort.conf.default to /usr/pkg/etc/snort/snort.conf
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/threshold.conf to /usr/pkg/etc/snort/threshold.conf
snort-2.8.5.1: copying /usr/pkg/share/examples/snort/unicode.map to /usr/pkg/etc/snort/unicode.map
```

=====

The following files should be created for snort-2.8.5.1:

```
/etc/rc.d/snort (m=0755)
  [/usr/pkg/share/examples/rc.d/snort]
```

```
=====
```

```
=====
```

```
$NetBSD: MESSAGE,v 1.5 2005/09/14 12:46:52 adrianp Exp $
```

Listing 2a. Output of running Snort, Initializing Snort

```

Output of running Snort
Running in IDS mode

---= Initializing Snort =---
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file /usr/local/etc/snort/snort.conf
PortVar 'HTTP_PORTS' defined : [ 80 ]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1521 ]
Frag3 global config:
Max frags: 65536
Fragment memory cap: 4194304 bytes
Frag3 engine config:
Target-based policy: FIRST

Fragment min_ttl: 1
Fragment ttl_limit (not used): 5
Fragment Problems: 1
Stream5 global config:
Track TCP sessions: ACTIVE
Max TCP sessions: 8192
Memcap (for reassembly packet storage): 8388608
Track UDP sessions: INACTIVE
Track ICMP sessions: INACTIVE
Stream5 TCP Policy config:
Reassembly Policy: FIRST
Timeout: 30 seconds
Min ttl: 1
Options:
Static Flushpoint Sizes: YES
Reassembly Ports:
21 client (Footprint)
23 client (Footprint)
25 client (Footprint)
42 client (Footprint)
53 client (Footprint)
80 client (Footprint)
110 client (Footprint)
111 client (Footprint)
135 client (Footprint)
136 client (Footprint)
137 client (Footprint)
139 client (Footprint)
143 client (Footprint)
445 client (Footprint)
513 client (Footprint)

514 client (Footprint)
1433 client (Footprint)
1521 client (Footprint)
2401 client (Footprint)
3306 client (Footprint)
HttpInspect Config:
GLOBAL CONFIG
Max Pipeline Requests: 0
Inspection Type: STATELESS
Detect Proxy Usage: NO
IIS Unicode Map Filename: /usr/local/etc/snort/
                                unicode.map
IIS Unicode Map Codepage: 1252
DEFAULT SERVER CONFIG:
Server profile: All
Ports: 80 8080 8180
Flow Depth: 300
Max Chunk Length: 500000
Max Header Field Length: 0
Inspect Pipeline Requests: YES
URI Discovery Strict Mode: NO

Disable Alerting: NO
Oversize Dir Length: 500
Only inspect URI: NO
Ascii: YES alert: NO
Double Decoding: YES alert: YES
%U Encoding: YES alert: YES
Bare Byte: YES alert: YES
Base36: OFF
UTF 8: OFF
IIS Unicode: YES alert: YES
Multiple Slash: YES alert: NO
IIS Backslash: YES alert: NO
Directory Traversal: YES alert: NO
Web Root Traversal: YES alert: YES
Apache WhiteSpace: YES alert: NO
IIS Delimiter: YES alert: NO
IIS Unicode Map: GLOBAL IIS UNICODE MAP CONFIG
Non-RFC Compliant Characters: NONE
Whitespace Characters: 0x09 0x0b 0x0c 0x0d
rpc_decode arguments:
Ports to decode RPC on: 111 32771
alert_fragments: INACTIVE
alert_large_fragments: ACTIVE
alert_incomplete: ACTIVE
alert_multiple_requests: ACTIVE
Portscan Detection Config:

```

Listing 2b. Showing configuration

```

Detect Protocols: TCP UDP ICMP IP
Detect Scan Type: portscan portsweep decoy_portscan
                  distributed_portscan
Sensitivity Level: Low
Memcap (in bytes): 10000000
Number of Nodes: 36900

Tagged Packet Limit: 256
Loading dynamic engine /usr/local/lib/snort/
                  dynamicengine/libsengine.so...
                  done
Loading all dynamic preprocessor libs from /usr/local/
                  lib/snort/dynamicpreprocessor/...
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//lib_
                  sfdynamic_preprocessor_example.so...
                  done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  dcerpc_preproc.so... done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  dns_preproc.so... done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  ftptelnet_preproc.so... done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  smtp_preproc.so... done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  ssh_preproc.so... done
Loading dynamic preprocessor library /usr/local/lib/
                  snort/dynamicpreprocessor//libs_
                  ssl_preproc.so... done
Finished Loading all dynamic preprocessor libs from /usr/
                  local/lib/snort/dynamicpreprocessor/
FTPTelnet Config:

Inspection Type: stateful
Check for Encrypted Traffic: YES alert: YES
Continue to check encrypted data: NO
TELNET CONFIG:
Ports: 23
Are You There Threshold: 200
Normalize: YES
Detect Anomalies: NO
FTP CONFIG:

FTP Server: default
Ports: 21
Check for Telnet Cmds: YES alert: YES
Identify open data channels: YES
FTP Client: default
Check for Bounce Attacks: YES alert: YES
Check for Telnet Cmds: YES alert: YES
Max Response Length: 256

SMTP Config:
Ports: 25 587 691
Inspection Type: Stateful
Normalize: EXPN RCPT VRFY
Ignore Data: No
Ignore TLS Data: No
Ignore SMTP Alerts: No
Max Command Line Length: Unlimited
Max Specific Command Line Length:
ETRN:500 EXPN:255 HELO:500 HELP:500 MAIL:260
RCPT:300 VRFY:255
Max Header Line Length: Unlimited
Max Response Line Length: Unlimited
X-Link2State Alert: Yes
Drop on X-Link2State Alert: No
Alert on commands: None

DCE/RPC Decoder config:
Autodetect ports ENABLED
SMB fragmentation ENABLED
DCE/RPC fragmentation ENABLED
Max Frag Size: 3000 bytes
Memcap: 100000 KB
Alert if memcap exceeded DISABLED

DNS config:
DNS Client rdata txt Overflow Alert: ACTIVE
Obsolete DNS RR Types Alert: INACTIVE
Experimental DNS RR Types Alert: INACTIVE
Ports: 53

SSLPP config:
Encrypted packets: not inspected
Ports:

992 993 994 995

+++++
Initializing rule chains...
1 Snort rules read

```

Listing 2c. Reading rule chains

```

1 detection rules
0 decoder rules
0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
+++++
+-----[Rule Port Counts]-----
| tcp udp icmp ip
| src 0 0 0 0
| dst 0 0 0 0
| any 1 0 0 0
| nc 1 0 0 0
| s+d 0 0 0 0
+-----

+-----[thresholding-config]-----
| memory-cap : 1048576 bytes
+-----[thresholding-global]-----
| none
+-----[thresholding-local]-----
| none
+-----[suppression]-----
| none
+-----

Rule application order: activation->dynamic->pass->drop-
                        >alert->log
Log directory = /var/log/snort/
Verifying Preprocessor Configurations!
0 out of 512 flowbits in use.
***
*** interface device lookup found: em0
***

Initializing Network Interface em0
Decoding Ethernet on interface em0

[ Port Based Pattern Matching Memory ]
+-[AC-BNFA Search Info Summary]-----
| Instances : 4
| Patterns : 69
| Pattern Chars : 297
| Num States : 225
| Num Match States : 69
| Memory : 10.83Kbytes
| Patterns : 1.63K
| Match Lists : 1.72K

+-----
--== Initialization Complete ==--

,,_ -*> Snort! <*-
o" )~ Version 2.8.2.1 (Build 16) NetBSD
''' By Martin Roesch & The Snort Team: http://
                               www.snort.org/team.html
(C) Copyright 1998-2008 Sourcefire Inc., et al.
Using PCRE version: 7.7 2008-05-07

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.8
                <Build 14>
Preprocessor Object: SF_SSLPP Version 1.0 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 7>
Preprocessor Object: SF_FTPTELNET Version 1.1 <Build 10>
Preprocessor Object: SF_DNS Version 1.1 <Build 2>
Preprocessor Object: SF_DCERPC Version 1.1 <Build 4>
Preprocessor Object: SF_Dynamic_Example_Preprocessor
                    Version 1.0 <Build 1>

Not Using PCAP_FRAMES
*** Caught Int-Signal
=====
Packet Wire Totals:
Received: 0
Analyzed: 0 (0.000%)
Dropped: 0 (0.000%)
Outstanding: 0 (0.000%)
=====
Breakdown by protocol (includes rebuilt packets):
ETH: 0 (0.000%)
ETHdisc: 0 (0.000%)
VLAN: 0 (0.000%)
IPV6: 0 (0.000%)
IP6 EXT: 0 (0.000%)
IP6opts: 0 (0.000%)
IP6disc: 0 (0.000%)
IP4: 0 (0.000%)
IP4disc: 0 (0.000%)
TCP 6: 0 (0.000%)
UDP 6: 0 (0.000%)
ICMP6: 0 (0.000%)
ICMP-IP: 0 (0.000%)
TCP: 0 (0.000%)
UDP: 0 (0.000%)
ICMP: 0 (0.000%)
TCPdisc: 0 (0.000%)

```

Listing 2d. Initialization Complete

```

UDPdisc: 0 (0.000%)
ICMPdis: 0 (0.000%)
FRAG: 0 (0.000%)
FRAG 6: 0 (0.000%)

EAPOL: 0 (0.000%)
ETHLOOP: 0 (0.000%)
IPX: 0 (0.000%)
OTHER: 0 (0.000%)
DISCARD: 0 (0.000%)
InvChkSum: 0 (0.000%)
S5 G 1: 0 (0.000%)
S5 G 2: 0 (0.000%)
Total: 0
=====
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
=====
Frag3 statistics:
Total Fragments: 0
Frag Reassembled: 0
Discards: 0
Memory Faults: 0
Timeouts: 0
Overlaps: 0
Anomalies: 0
Alerts: 0
FragTrackers Added: 0

FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
Frag Nodes Inserted: 0
Frag Nodes Deleted: 0
=====
Stream5 statistics:
Total sessions: 0
TCP sessions: 0
UDP sessions: 0
ICMP sessions: 0
TCP Prunes: 0
UDP Prunes: 0
ICMP Prunes: 0
TCP StreamTrackers Created: 0
TCP StreamTrackers Deleted: 0
TCP Timeouts: 0
TCP Overlaps: 0
TCP Segments Queued: 0
TCP Segments Released: 0
TCP Rebuilt Packets: 0
TCP Segments Used: 0
TCP Discards: 0
UDP Sessions Created: 0
UDP Timeouts: 0
UDP Discards: 0
Events: 0
=====
=====
=====

```

Recently I wrote a server application that receives and sends data through a port to other clients in the network. Nothing special, anybody can write such a daemon to do that. Yes, exactly, but I am not a perfect programmer and I usually have some bugs in my applications (like many developers). In fact, who does not make mistakes? Probably the one that does not work...

So, this server had some weak points and I needed to protect it from exploitation of these bugs. Of course I had the idea how to fix the bugs but some time to do that was needed and for that moment I had no time to fix any bug. Instead of that, I had to prepare some solution because I needed that server to work and I needed it to work correctly. Probably many people can say: *Of course, fixing the bug is the most appropriate solution, after that the server should be ok.* Yes, but all the applications have bugs and the bugs appear progressively. So, let me explain how I used Snort to fix

the server. Snort gives me the opportunity to sniff the traffic, so I could see the packets and log a message if there was an attempt for bug exploitation or I could even drop the packet. This is just a small area where Snort can be useful.

Also I would remark the perfect combination of NetBSD, its firewall, and Snort, that one can use. This combination allows one to use it for a border machines where the security is from high importance. I would prefer to use it to take the maximum possible protection for my network. Let me show you an example with the server mentioned before. 3 days after the server started I had to analyze the logs and I had totally shocked. The hack attempts were sooo many. Actually the server offers 2 services – SSH and the service of my daemon. There were dozens of attempts to login with some usernames like *melinda*, *jack*, and etc... also I had some attempts with the “root” user. Of course, I was prepared for this

and I configured Snort to inspect the incoming packets. Then I checked the log file from time to time to collect new information about the IP addresses that *breached the line*.

Let's have a look about another situation. I have connection to internet, I use pppoe and I see how difficult is the life of the ISP of my area. I would advice any internet service providers to use Snort. Basically, ISP should provides service to all of its customers but there are a lot of customers that do not want just to use that service but also want to use it for bad things like hacking, stealing passwords or some other illegal activity. So, in simple words, the ISP has very bad job... The provider also has to protect its customers from each other and protect their data. Let's do not forget the threats from the internet and if we summarize all these things together we have the real position of the ISP. And I would say that is not that good position. From one side the ISP should provide service and from the other side this provider should protect the customers.

This is the right place where Snort and NetBSD together can fight all of the problems of such ISPs.

Basically, Snort can be used to detect, stop, and report illegal activity and in that case it can make the ISP's life easier. This is just an example how the intrusion detection system like Snort can be useful.

Let's get Snort to work on our machine (see Listing 1).

That installed the snort on my system, you should check if you need some other packages to be installed, it is different for every system, so if the `pkg_add` program needs more packages you should install them as well.

Then you can focus on your work with snort. Actually the work with it is very simple. There is a configuration file called `snort.conf` and several rules files.

I have the configuration file in `/etc/snort.conf` and the rules are there also. So, all the files are available `/etc/snort/` directory. You can use them at any location that you want, this is not important.

To use snort, you will need to perform the following steps:

Step 1.

In case you don't have `PKG_RCD_SCRIPTS` set in your `/etc/mk.conf`, copy

Listing 3. Example detection log

```
[**] [1:999369:0] A Test log [**]
[Priority: 0]
02/28-18:53:59.755446 52:54:0:12:35:2 -> 8:0:27:BF:DA:3 type:0x800 len:0x3C
192.168.0.1:3128 -> 10.0.2.15:61247 TCP TTL:64 TOS:0x0 ID:27871 IpLen:20 DgmLen:44
***A**S* Seq: 0x1554F001 Ack: 0xA122F39E Win: 0x2000 TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:999369:0] A Test log [**]
[Priority: 0]
02/28-18:53:59.756180 52:54:0:12:35:2 -> 8:0:27:BF:DA:3 type:0x800 len:0x3C
192.168.0.1:3128 -> 10.0.2.15:61247 TCP TTL:64 TOS:0x0 ID:27872 IpLen:20 DgmLen:40
***A**** Seq: 0x1554F002 Ack: 0xA122F3F1 Win: 0x2238 TcpLen: 20

[**] [1:999369:0] A Test log [**]
[Priority: 0]
02/28-18:53:59.768326 52:54:0:12:35:2 -> 8:0:27:BF:DA:3 type:0x800 len:0x28E
192.168.0.1:3128 -> 10.0.2.15:61247 TCP TTL:64 TOS:0x0 ID:27873 IpLen:20 DgmLen:640
***AP*** Seq: 0x1554F002 Ack: 0xA122F3F1 Win: 0x2238 TcpLen: 20

[**] [1:999369:0] A Test log [**]
[Priority: 0]
02/28-18:53:59.769342 52:54:0:12:35:2 -> 8:0:27:BF:DA:3 type:0x800 len:0x3C
192.168.0.1:3128 -> 10.0.2.15:61247 TCP TTL:64 TOS:0x0 ID:27874 IpLen:20 DgmLen:40
***A***F Seq: 0x1554F25A Ack: 0xA122F3F1 Win: 0x2238 TcpLen: 20
```



```
/usr/pkg/share/examples/rc.d/snort to  
/etc/rc.d/snort and add
```

```
snort=YES
```

Step 2.

Now start snort by issuing the command

```
/etc/rc.d/snort start
```

We also can run snort on *dry* without to start it as a service.

Run Snort with the following command:

```
snort -c /path-to-your-config-file -de -l /path-to-your-log  
-directory
```

That will run snort with configuration file at your path-to-your-config-file and log directory at /path-to-your-log-directory.

This is some example output that you should see Listing 2.

Snort exiting

Run time prior to being shutdown was 3.14117 seconds.

Some example logs. As you can see the logs have information about source, destination ports, and other basic information about the packet (see Listing 3).

Summary

Any type of an operating system can be used for such a server but it should be fast, reliable and secure. The performance is very important because IDS is a network dependent system and as fast as our server process the packets as fast will detect an attack. And as fast it detects the attack as fast it will alert other systems about the situation. So, in order to achieve this goals, we need a big iron and a fast operating system.

SVETOSLAV CHUKOV

Svetoslav Chukov/Chukich is a system administrator with experience in BSD and Linux. Some of the primary interests for him are: system security, firewalls, improving performance of the servers, filesystem optimizations, benchmarks, high availability and some others... He enjoys benchmarking huge storage servers, or if they aren't available, he also likes to play with „more simple“ 2 nodes clusters.

If you wish to contribute to BSD magazine, share your knowledge and skills with other BSD users - do not hesitate - read the guidelines on our website and email us your idea for an article.

Join our team!

Become BSD magazine

Author or Betatester

As a betatester you can decide on the contents and the form of our quarterly. It can be you who read the articles before everybody else and suggest the changes to the author.

Contact us:

editors@bsdmag.org

www.bsdmag.org

MAGAZINE

BSD

In the next issue:

- LibGTop and OpenBSD**
- Dtrace on FreeBSD**
- Collectd part 2**
- and Other !**

**Next issue is coming in
October!**

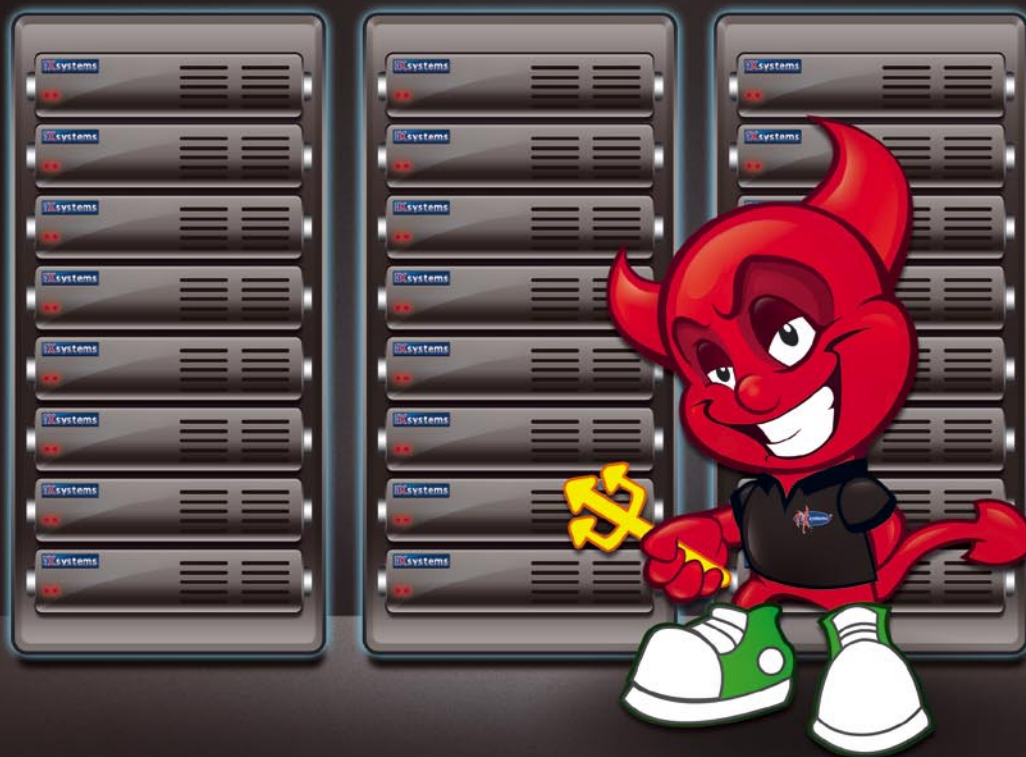
Looking for help, tip or advice?
Want to share your knowledge with others?

EMISADAMMAGAZINE

BSD

Give us your opinion about the magazine's content
and help us create the most useful source for you!

What has your server vendor done for BSD lately? Probably, not much.



Work with a vendor that **supports** the operating system you love!

iX is the corporate sponsor of the PC-BSD® Project, a major corporate donor to the FreeBSD Foundation, and leads the FreeNAS™ development team -- all while employing some of the most brilliant minds in the FreeBSD® community. For BSD hardware and software expertise, look no further.

1-855-GREP-4-IX

<http://www.iXsystems.com/community>

