

# Securing FreeBSD

Presented by: Adam Wien

# Main Topics

- `hosts.allow`
- `sshd`
- `login.conf`
- `cvsup`
- `IPFW`



**cPanel**

Training Seminar 2006

# hosts.allow

- Location:
  - /etc/hosts.allow
- Layout:
  - service : FROM : allow/deny
- Default is to allow all services
  - ALL : ALL : allow



# Editing hosts.allow

- We want to deny all and allow some.
- ftpd: ALL : allow
- sshd: ALL : allow
- exim : ALL : allow
- ALL : ALL : deny

# sshd

- Improving on default security.
- Location:
  - `/etc/ssh/sshd_config`

# Secure Shell(SSH) Public Key Authentication

- Only someone with the public and private key may access your server.
- More administrator involvement in who is provided shell access.
- Disables the need to authenticate against the servers password file only.

# Enabling Public Key Authentication

- Edit `/etc/ssh/sshd_config`
- Uncomment `PublicKeyAuthentication` and change it to “yes”
- Uncomment `AuthorizedKeysFile`
- Restart `sshd`
  - `/etc/rc.d/sshd restart`

# Generating Public and Private Keys

- `ssh-keygen -t dsa -f server.key`
- This will create two files:
  - `server.key`
    - private key
  - `server.key.pub`
    - public key
- Install your public key into your, or the user's home directory under `.ssh/authorized_keys`.

# Accepting only SSH2 Connections

- We want to enable only ssh2 compliant connections.
- Edit `/etc/ssh/sshd_config`
- Uncomment “Protocol 2”
- Restart sshd
  - `/etc/rc.d/sshd restart`

# Disabling Local PAM Authentication in SSH

- We want to disable local password authentication completely so a user must have a valid key pair to gain shell access to the server.
- Edit `/etc/ssh/sshd_config`
- Change both `PasswordAuthentication` and `UsePAM` to “no”.
- Restart `sshd`(`/etc/rc.d/sshd restart`)
- Be sure you have generated a public key pair and are able to access your server using them or this step will lock you out!

# Setting User Limits with login.conf

- Location:
  - /etc/login.conf
- You can either make changes in /etc/login.conf or create a file in a user's home directory called .login\_conf.

# Setting User Limits with login.conf

Username:\

:maxproc=50:\

:memoryuse=50M:\

openfiles=20:

- Run `cap_mkdb` to build the `login.conf` database file.

# cvsup

- cvsup is a program used for downloading sources using a cvsup server.
- Installation: 'pkg\_add -r cvsup-without-gui'
- We need to copy our cvsup configuration file. We'll just use an existing example and edit it.
- `cp /usr/share/examples/cvsup/stable-supfile /root`
- Edit `/root/stable-supfile`

# stable-supfile

- \*default host=cvsup11.freebsd.org
- \*default base=/var/db
- \*default prefix=/usr
- \*default release=cv tag=RELENG\_5
- \*default delete use-rel-suffix
- \*default compress
- src-all
- ports-all tag=.

# stable-supfile Explained

- The only line you should really be concerned with here is the release line. Specifying `RELENG_5` will give you the latest version of FreeBSD 5 STABLE. Specifying `RELENG_5_4` will provide you with the latest RELEASE of FreeBSD 5.4.

# Downloading Sources

- Next we need to suck down our sources by running 'cvsup -g -L 2 stable-supfile'.
- This will take a while depending on your connection speed.

# Adding IPFW and QUOTA support to your kernel

- Enter the directory `/usr/src/sys/`uname -p`/conf/` copy the GENERIC kernel to CPANEL.

# Kernel Options

- options IPFIREWALL
- options  
IPFIREWALL\_DEFAULT\_TO\_ACCEPT
- options IPFIREWALL\_VERBOSE
- options DUMMYNET
- options QUOTA

# Recompiling your Kernel

- Enter the `/usr/src` directory and issue the following command
  - `make buildkernel KERNCONF=CPANEL`
- Next we need to install the new kernel by issuing
  - `make installkernel KERNCONF=CPANEL`

# Rebuilding System Binaries

- Enter the `/usr/src` directory and issue
  - `make buildworld`
- When this finishes we need to install those binaries by running
  - `make installworld`
- When this finishes we need to run
  - `mergemaster`

# Enabling IPFW

- First we need to edit
  - /etc/rc.conf
- firewall\_enable="YES"
- firewall\_script="/etc/rc.firewall"
- firewall\_type="/etc/ipfw.rules"
- firewall\_quiet="NO"
- firewall\_logging="YES"
- firewall\_flags=""

# Basic Rule

- This rule will allow all traffic coming in from the lo0 interface(loopback). There should be no need to filter this.
- add allow ip from any to any in via lo0
- add, add a rule
- allow, allow this connection
- ip, all ip protocols
- from, source
- any, anywhere
- to, destination
- any, anywhere

# Basic Rule(continued)

- in, inbound packets
- via, via an interface
- lo0, loopback interface

# Allowing Resolving Nameserver

- We need to allow the nameservers listed in
  - /etc/resolv.conf
- Here's an example using cPanel's primary nameserver.

add allow udp from any to 69.90.250.18 out via  
xl0

add allow udp from 69.90.250.18 in via xl0

# Default Deny Rule

- Deny and log everything.

```
add deny log logamount 1000 all from any to  
any in via xl0
```

# Conclusion

- FreeBSD is a very secure and capable operating system.
- It is the operating system of choice for large scale Company's such as Yahoo and is widely used at cPanel.
- Included on the CD's provided is the complete list of firewall rules for a default cPanel<sup>®</sup> installation.
- Questions and Answers