Securing trixbox CE

By
Tim Yardley
AKA Engineer Tim

Preface: This document is solely intended as a guideline or starting point. In no way is it written that everyone should do exactly what I put in here. These are just good general purpose tweaks to help people sleep at night. By no means is this the only way to secure a trixbox install. Again, this is just a starting point and I know it works. If you have ideas or recommendations then feel free to pass them along to the trixbox team via the forums.

Services

The first steps I normally take to secure any server that is my responsibility is to ensure all services that are not needed are disabled. This can be done with the *chkconfig* command. To list what services are starting at boot issue the following command. I have highlighted the services that are started on boot.

```
[trixbox1.localdomain ~]# chkconfig --list
anacron         0:off   1:off   2:on    3:on    4:on    5:on    6:off
asterisk        0:off   1:off   2:off   3:off   4:off   5:off   6:off
avahi-daemon    0:off   1:off   2:off   3:off   4:off   5:off   6:off
avahi-dnsconfd  0:off   1:off   2:off   3:off   4:off   5:off   6:off
bgpd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
capi            0:off   1:off   2:off   3:off   4:off   5:off   6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
dc_client       0:off   1:off   2:off   3:off   4:off   5:off   6:off
dc_server       0:off   1:off   2:off   3:off   4:off   5:off   6:off
dhcpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
dhcrelay        0:off   1:off   2:off   3:off   4:off   5:off   6:off
ez-ipupdate     0:off   1:off   2:off   3:off   4:off   5:off   6:off
haldaemon       0:off   1:off   2:off   3:on    4:on    5:on    6:off
httpd           0:off   1:off   2:off   3:on    4:on    5:on    6:off
ip6tables       0:off   1:off   2:off   3:off   4:off   5:off   6:off
ipmi            0:off   1:off   2:off   3:off   4:off   5:off   6:off
iptables        0:off   1:off   2:off   3:off   4:off   5:off   6:off
ircd            0:off   1:off   2:off   3:on    4:on    5:on    6:off
isdn            0:off   1:off   2:off   3:off   4:off   5:off   6:off
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
lm_sensors      0:off   1:off   2:on    3:on    4:on    5:on    6:off
lvm2-monitor    0:off   1:on    2:on    3:on    4:on    5:on    6:off
mDNSResponder   0:off   1:off   2:off   3:on    4:on    5:on    6:off
mcstrans        0:off   1:off   2:off   3:off   4:off   5:off   6:off
mdmonitor       0:off   1:off   2:on    3:on    4:on    5:on    6:off
mdmpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
memcached       0:off   1:off   2:on    3:on    4:on    5:on    6:off
messagebus      0:off   1:off   2:off   3:on    4:on    5:on    6:off
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
mysqld          0:off   1:off   2:off   3:on    4:on    5:on    6:off
named           0:off   1:off   2:off   3:off   4:off   5:off   6:off
netconsole      0:off   1:off   2:off   3:off   4:off   5:off   6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
netplugd        0:off   1:off   2:off   3:off   4:off   5:off   6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:on    4:on    5:on    6:off
openibd         0:off   1:off   2:on    3:on    4:on    5:on    6:off
ospf6d          0:off   1:off   2:off   3:off   4:off   5:off   6:off
ospfd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
postfix         0:off   1:off   2:on    3:on    4:on    5:on    6:off
rdisc           0:off   1:off   2:off   3:off   4:off   5:off   6:off
restorecond     0:off   1:off   2:on    3:on    4:on    5:on    6:off
ripd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
ripngd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
rpcgssd         0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

```
rpcidmapd       0:off   1:off   2:off   3:on    4:on    5:on    6:off
rpcsvcgssd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
smb             0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmptrapd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
vsftpd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
winbind         0:off   1:off   2:off   3:off   4:off   5:off   6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
zaptel          0:off   1:off   2:on    3:on    4:on    5:on    6:off
zebra           0:off   1:off   2:off   3:off   4:off   5:off   6:off
xinetd based services:
        chargen-dgram:  off
        chargen-stream: off
        daytime-dgram:  off
        daytime-stream: off
        discard-dgram:  off
        discard-stream: off
        echo-dgram:     off
        echo-stream:    off
        rsync:          off
        tcpmux-server:  off
        tftp:           on
        time-dgram:     off
        time-stream:    off
```

Now we can see that several services have started that we may or may not want on. Anacron, crond, haldaemon, httpd, kudzu, lm_sensors, lvm2-monitor, mDNSResponder, mdmonitor, memcached, messagebus, mysqld, network, ntpd, postfix, sshd, syslog, xinetd, and zaptel are all required to be on for a functioning trixbox system. The rest of the services can be disabled. If you plan to use ftp on the system, then leave vsftpd on. You can use *chkconfig <service name> off*.

```
chkconfig ircd off
chkconfig netfs off
chkconfig nfslock off
chkconfig openibd off
chkconfig portmap off
chkconfig restorecond off
chkconfig rpcgssd off
chkconfig rpcidmapd off
chkconfig vsftpd off
```

Now that we have the services disabled from starting at boot, lets go ahead and stop them from running right now.

```
service ircd stop
service netfs stop
service nfslock stop
service openibd stop
service portmap stop
service restorecond stop
service rpcgssd stop
service rpcidmapd stop
service vsftpd stop
```

Securing SSH

Securing sshd is critical to ensuring a somewhat worry free system. There are several ways to make it more difficult to get into a server over ssh.

Step 1. Create a user on the system to only allow ssh connections from. The username should be something that you only know and is not easily guessed. Here we will create a user called *trixuser* and assign a password to it. The password should be something with letters, numbers, symbols and not based off a dictionary word. For good passwords, try to use something that is both obscene and vulgar, this ensures that you will never repeat it out loud or want to tell anyone what it is. Also try and string it into a sentence making sure to use the letters, numbers, and symbols. Spaces in passwords work good too and are hard to add in scripts that might try to break into your server.

```
[trixbox1.localdomain init.d]# useradd trixuser
[trixbox1.localdomain init.d]# passwd trixuser
```

Now ensure that the new account works by using ssh to login to the trixbox CE server with this new account. If it does not let you in, make sure the password is correct or try to reset it. If it works continue on.

Only allowing one account access to the system over ssh is a great way to lock out most brute force attacks. To do this we need to edit the file in /etc/ssh/sshd_config and add the following to the file.

```
AllowUsers trixuser
```

I like to edit the *PermitRootLogin* setting so that root can't login over ssh. Remove the # from in front of the setting and change the yes to no.

```
PermitRootLogin no
```

Finally, I would recommend changing the Port setting from the standard 22, which everyone knows as ssh, to something else. Be careful what you change it to, you don't want the port to conflict with a port in use or that might become in use. You can also attract more attention to the server if you put it on another known port than if you left it at 22. In this example we will use 2222. Please decide on your own port number to use on your system. The setting we edit is *Port 22* in /etc/ssh/sshd_config. Remove the # from in front of the setting and change 22 to 2222.

```
Port 2222
```

We need to restart sshd for the changes to take affect. Please use caution when changing these settings on a remote system that you can't easily get too. If there is a error in the config it could cause sshd to not restart.

```
service sshd restart
```

Now test to make sure that you can get into the server over ssh. The root user should be denied access and only the user we created should be allowed to get in. Don't forget to change your ssh port to 2222 when connecting. In putty it is listed next to the IP address, on the command line the flag is *-p port*.

Firewall APF and BFD

I like to use rfxnetworks apf and bfd for fire walling all my systems. Links to their software can be found here.

apf - http://rfxnetworks.com/apf.php
bfd - http://rfxnetworks.com/bfd.php

rfxnetworks has many other great projects that are worth looking at. But this document will only cover apf/bfd.

APF stands for Advanced Policy Firewall. This is used to control iptables on the system to allow or disallow ports to be open. APF has additional features that make it stand out above the rest. Reactive address blocking (RAB), QoS (TOS), direct integration with BFD, many many more, see site for full details.

BFD stands for Brute Force Detection. This is used to monitor any failed logins and block IP addresses from getting in. This runs as a cron daemon and works perfectly with APF.

To install both of these applications is very simple. You can download them both from the rfxnetworks links, uncompress them, and then run the install.sh script. I have also created a installer script that can be downloaded to your machine and ran. This will install the latest and greatest apf/bfd. To get this script you will need to use wget or other method to pull it off a web server. You will also want to do this as root.

```
wget http://engineertim.com/install_apf_bfd.sh
chmod 755 install_apf_bfd.sh
./install_apf_bfd.sh
```

This will start the install process for both apf and bfd. Once completed you will be returned to a command prompt.

APF

To configure APF is pretty easy and I will touch on a few of the config file options in this document. All of the options are covered in great detail on their website and well commented in the conf.apf file.

The config file for APF lives in /etc/apf and is called conf.apf

We will need to edit the conf.apf file, I like to use vi but any command line editor will work.

If you have multiple interfaces on your trixbox setup, you will want to set the IFACE_IN and IFACE_OUT to your external interface. This is the, untrusted network interface that is connected to the internet. If you have a second card eth1, that is used for internal network, trusted network, you can set the IFACE_TRUSTED to this interface. Please see the comments in the conf.apf if you are uncertain.

The setup script will try to properly determine which interface is used for untrusted network and place it in the appropriate field.

I like to change the value of the SET_TRIM to 0. This value sets the total amount of rules allowed inside of the deny trust system. It is designed to save memory and start time. With the default value of 50, the system will start to purge old rules once this number is met. With the inclusion of BFD, this number will generally climb past 50. Setting this value to 0 will disable this feature.

```
SET_TRIM=”0”
```

APF has the ability to do QoS on packets, this is defined with the TOS values in the conf.apf. For SIP and IAX, I set the following.

```
TOS_8="21,20,80,4569,5060,10000_20000"
```

This also requires a small tweak to one of the config files that I will discuss later in the document in order to tag UDP packets.

Since we changed the SSH port to a different number, we have to tweak the conf.apf to match this new port.

```
HELPER_SSH_PORT="2222"
```

Make sure to place the correct port number that you decided to run SSH on.

Ingress filtering is used to open inbound ports for access, both TCP and UDP have separate settings. For a trixbox setup, the following ports should be open, both TCP and UDP are listed. If you are not using tftp, then do not have port 69 open. Do not forget to change the SSH port from 22, to the port you choose to run SSH on. Otherwise you will be locked out, here we are using port 2222 from our example above. I have not included IAX ports in this setup. There is a easy way to ensure that only specific hosts can use IAX that I will cover later. This is handy if you use IAX to do interoffice trunks like I do, but don't want IAX ports open for the world to see.

```
IG_TCP_CPORTS="2222,69,80,5060,6600,10000_20000"
IG_UDP_CPORTS="69,5060,10000_20000"
```

Egress filtering is used to allow outbound filtering. I don't use egress filtering and it will not be covered in this document. It is set to EGF=”0” , or disabled by default.

In the section of the conf.apf file called Imported Rules, there are settings for various feeds. Some of these feeds are very handy and I use them all. I have even setup my own custom feed that allows me to tweak all of my servers with global deny rules. You can disable or enable this feature with the USE_DS setting. A “1” is enabled, a “0” is disabled.

We are now ready to start APF for the first time. Double check that the SSH port is set correctly to the one you are using. If you start APF right now and something is wrong, it will disable itself in 5 minutes. This is called DEVEL_MODE and is the first setting in the conf.apf file. Leave this set to “1” until you are certain you can get in via ssh and things are working.

To see a list of command line options run *apf* without any flags.

```
[trixbox1.localdomain apf]# apf
apf(3402): {glob} status log not found, created
APF version 9.6 <apf@r-fx.org>
Copyright (C) 1999-2007, R-fx Networks <proj@r-fx.org>
Copyright (C) 2007, Ryan MacDonald <ryan@r-fx.org>
This program may be freely redistributed under the terms of the GNU GPL
usage /usr/local/sbin/apf [OPTION]
-s|--start ......................... load all firewall rules
-r|--restart ....................... stop (flush) & reload firewall rules
-f|--stop........ .................. stop (flush) all firewall rules
-l|--list .......................... list all firewall rules
-t|--status ........................ output firewall status log
-e|--refresh ....................... refresh & resolve dns names in trust rules
-a HOST CMT|--allow HOST COMMENT ... add host (IP/FQDN) to allow_hosts.rules and
                                     immediately load new rule into firewall
-d HOST CMT|--deny HOST COMMENT .... add host (IP/FQDN) to deny_hosts.rules and
                                     immediately load new rule into firewall
-u|--remove HOST ................... remove host from [glob]*_hosts.rules
                                     and immediately remove rule from firewall
-o|--ovars ......................... output all configuration options
```

To start APF we issue the following command.

```
[trixbox1.localdomain apf]# apf -s
apf(3445): {glob} activating firewall
apf(3489): {glob} determined (IFACE_IN) eth0 has address 192.168.1.31
apf(3489): {glob} determined (IFACE_OUT) eth0 has address 192.168.1.31
apf(3489): {glob} loading preroute.rules
apf(3489): {resnet} downloading http://r-fx.ca/downloads/reserved.networks
apf(3489): {resnet} parsing reserved.networks into
/etc/apf/internals/reserved.networks
apf(3489): {glob} loading reserved.networks
apf(3489): {glob} SET_REFRESH is set to 10 minutes
apf(3489): {glob} loading bt.rules
apf(3489): {dshield} downloading http://feeds.dshield.org/top10-2.txt
apf(3489): {dshield} parsing top10-2.txt into /etc/apf/ds_hosts.rules
apf(3489): {dshield} loading ds_hosts.rules
apf(3489): {sdrop} downloading http://www.spamhaus.org/drop/drop.lasso
apf(3489): {sdrop} parsing drop.lasso into /etc/apf/sdrop_hosts.rules
apf(3489): {sdrop} loading sdrop_hosts.rules
apf(3489): {glob} loading common drop ports
..........trimmed for this document.........
apf(3489): {glob} default (ingress) input drop
apf(3445): {glob} firewall initalized
apf(3445): {glob} !!DEVELOPMENT MODE ENABLED!! - firewall will flush every 5
minutes.
```

We can see that APF has started, downloaded some rules from dshield.org and spamhaus.org and then told us it is in DEVELOPMENT MODE. Now test connecting to your server over SSH to ensure that you have setup the correct port number ingress. If you can't connect, you will have to wait 5 minutes and then APF will shutdown. Once you are sure you can get in with SSH we can change the conf.apf file from DEVEL_MODE="1" to DEVEL_MODE="0" and restart/start APF. APF will start and not warn you about being in DEVELOPMENT MODE, your firewall should be good to go.

APF additional tweaks. This setup might not be ideal for everyone. If you connect to your provider over IAX then you will definitely want to add the IAX ports to the conf.apf. However if you have two or more systems that you connect to each other over IAX for interoffice connections, then this is the way to go. This will work with static IP addresses and DYNDNS setups alike. You can use a fully qualified DNS hostname or IP address. One of the flags for the apf command is -a, which is allow. This will globally allow a host to connect to this system, bypassing the firewall rules. I can't stress how handy this is. Some examples are allowing a SNMP query, IAX connections, or other ports that you do not want open, but need to allow specific hosts to connect to. To do this just issue the following command, substitute your remote system IP address with the one I have here.

```
apf -a 192.168.1.216
```

This will allow the system 192.168.1.216 to connect to any port on the firewalled server, thereby bypassing the firewall rules. If you are running APF on both systems, be sure to do the same thing on the other host using the correct IP address.

APF also allows a system admin to block a host or a complete subnet. This is handy if you see someone attempting to connect to your machine over ftp, telnet, ssh, etc.. To block a specific host use the following, be sure to use the IP you want to block.

```
apf -d 192.168.1.216
```

To block a complete subnet (CIDR) the command is very similar.

```
apf -d 202.86.128.0/24
```

This will block the entire subnet. You can sometimes get the subnet (CIDR) listing using a whois on the IP address. You can also lookup a CIDR by ip on google or ripe.net. Be sure that the subnet is not one you are in or you could lock yourself out.

TOS for UDP packets are not defined for APF. Only TCP packets have the TOS bit set. There is a easy way to fix this. In the /etc/apf/internals folder is a file called, functions.apf. We need to edit this file manually. It is pretty straight forward as to what we need to changed, so don't worry.

There are several places we have to add a single line. Look for the TOS_ section in the functions.apf. It will look like this.

```
if [ ! "$TOS_0" == "" ]; then
    for i in `echo $TOS_0 | tr ',' ' '`; do
        i=`echo $i | tr '_' ':'`
        $IPT -t mangle  -A PREROUTING -p tcp --sport $i -j TOS --set-tos 0
    done
fi
```

We have to add the settings for UDP. We copy one line and change tcp to upd. A sample is below, highlighted in red.

```
if [ ! "$TOS_0" == "" ]; then
    for i in `echo $TOS_0 | tr ',' ' '`; do
        i=`echo $i | tr '_' ':'`
        $IPT -t mangle  -A PREROUTING -p tcp --sport $i -j TOS --set-tos 0
        $IPT -t mangle  -A PREROUTING -p udp --sport $i -j TOS --set-tos 0
    done
fi
```

This additional line has to be done for all the TOS bits you are using. If you are only using TOS_8 , then only worry about doing it for those. Make sure you do the tospostroute and tospreroute sections.

BFD

Brute force detection is used to capture illegitimate login attempts for services on the system. I see quite often a large number of ssh attempts into several servers that have not had the ssh port changed. These attempts are often a outside attempt to gain access by running dictionary attacks against common user names. These can now easily be stopped by using BFD. If you ran the install_apf_bfd.sh then bfd should be installed.

The configuration file for bfd is located in /usr/local/bfd and is called conf.bfd. This file, like the one for apf, is heavely commented and covered in great detail on the rfxnetworks website. I will just be covering some of the settings.

I should first premise this by stating that you can become locked out of your own server if you fail to type your own password correctly. This is another good reason to add a trusted system using the *apf -a* command. You can also add a host to ignore by adding the IP address to the /usr/local/bfd/ignore.hosts file. The ban command that BFD uses is tied directly to APF. The command is *apf -d*, which is the same way I showed you to manually ban addresses and subnets.

The first configuration variable we will look at is TRIG, this is the number of failed attempts before becoming banned. The default is 15 and is pretty good. Keep in mind this is per IP address connections, not account. So if 1 IP address fails 15 times using multiple accounts, it will be banned. Feel free to change this value if you want, don't make it to high of a number as this will allow more attempts to be made.

BFD has the ability to send emails out to alert of brute force attempts. This is a good idea, but it also requires that your trixbox setup can properly send emails. I will not be covering how to setup the server for mail in this document. To enable email alerts set the value of EMAIL_ALERTS to 1, then set the address you want emails to be sent to using EMAIL_ADRESS. You can define the subject for the email as well. This makes for easy flagging/filtering in email applications.

BFD runs from cron and places a cron entry in /etc/cron.d called bfd. This runs bfd every 3 minutes. This should be acceptable for almost anyone.

You can get a list of offending IP addresses using BFD on the command line. This is useful for looking at specific IP subnets that you might want to start blocking, if you see a pattern starting. To get this list the following command is used.

```
bfd – a
```

Securing HTTP

I like to secure my entire http access on trixbox to only allow a select few users to gain access. So when a person goes to http://ip_address ; they are prompted right away for a login. This can be a global login, or a per user account login. This login can also only allow /user/ access while a administrator account can be setup to allow access to the entire system. This setup is easy to enable and provides very fine grained access control. It will also keep unwanted people (hackers) from gaining access to the web interface.

I like to use mod_auth_mysql to lock down http. You can use plain mod_auth if you don't want to install new software. I will cover configuring both setups.

mod_auth_mysql can be used to limit access to documents served by apache by checking data in a MySQL database. All of my accounts are stored in MySQL along with groups, passwords, etc. To install mod_auth_mysql we will use yum, this will resolve any dependencies that might pop up.

yum install mod_auth_mysql

When it asks “Is this ok [y/N]:” just put “y” and hit enter. This will install the module in the appropriate location.

We now need to edit the configuration file. The configuration is located in /etc/httpd/conf.d and it is called auth_mysql.conf. Edit this file with your favorite editor with the following, or something similar.

```
LoadModule mysql_auth_module modules/mod_auth_mysql.so
<Directory /var/www/html>

AuthName "Authentication Required"

AuthType Basic

AuthMYSQLEnable on

AuthMySQLUser root

AuthMySQLPassword passw0rd

AuthMySQLDB userauth

AuthMySQLUserTable users

AuthMySQLNameField user_name

AuthMySQLPasswordField user_passwd

AuthMySQLGroupTable groups

AuthMySQLGroupField user_group

require group user
```

```
 require valid-user
 </Directory>

 <Directory /var/www/html/maint>
 AuthName "Authentication Required"
 AuthType Basic
 AuthMYSQLEnable on
 AuthMySQLUser root
 AuthMySQLPassword passw0rd
 AuthMySQLDB userauth
 AuthMySQLUserTable users
 AuthMySQLNameField user_name
 AuthMySQLPasswordField user_passwd
 AuthMySQLGroupTable groups
 AuthMySQLGroupField user_group
 require group admin
 </Directory>
```

If you have changed your root mysql password from the default, you will have to update the configuration above to reflect this new password. You can also change the groups if you feel you need to. I use admin and user, to reflect the level of access.

We now need to create a new database, I call mine userauth. To do this I use mysqladmin, you can use phpmyadmin if you would like.

```
mysqladmin -uroot -p create userauth
```

We now need to populate our database with our tables. Below is a schema that is used to do just that. Copy the following to a new file on the system and call it userauth.schema.sql , watch for formatting when copying this. You can alternatively use phpmyadmin to create the tables.

```
-- MySQL dump 10.10
--
-- Host: localhost    Database: userauth
-- ------------------------------------------------------
-- Server version5.0.22

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Current Database: `userauth`
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ `userauth` /*!40100 DEFAULT CHARACTER SET
latin1 */;

USE `userauth`;

--
-- Table structure for table `groups`
--

DROP TABLE IF EXISTS `groups`;
CREATE TABLE `groups` (
  `user_name` char(30) NOT NULL,
  `user_group` char(20) NOT NULL,
  PRIMARY KEY  (`user_name`,`user_group`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
```

```
CREATE TABLE `users` (
  `user_name` char(30) NOT NULL,
  `user_passwd` char(20) NOT NULL,
  `extension` int(10) NOT NULL,
  `email` char(50) NOT NULL,
  PRIMARY KEY  (`user_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

We now need to import the schema into the database. We do this with the mysql command.

mysql -uroot -p userauth < userauth.schema.sql

This will import the sql schema into mysql for use. We now need to populate our database with the users. This can be done with phpmyadmin easily, I will show you how to do it using mysql.

```
mysql -uroot -p

mysql> use userauth;

INSERT INTO `userauth`.`users` (`user_name` ,`user_passwd` ,`extension` ,`email`)
VALUES ('engineertim', ENCRYPT('letmein'),'','' );

INSERT INTO `userauth`.`users` (`user_name` ,`user_passwd` ,`extension` ,`email`)
VALUES ('kerry', ENCRYPT('letmein'),'','' );

INSERT INTO `groups` VALUES ('engineertim','user'),('engineertim','admin'),
('kerry','user')


mysql> exit
```

The first INSERT INTO, adds the user engineertim with a encrypted password of letmein. The second INSERT INTO, adds another user kerry with a encrypted password of letmein. The final INSERT INTO, puts the created users in their groups for permissions. The engineertim user is added to the user and admin group, while the kerry user is added to the user group. Feel free to change the names and passwords to whatever you want. If you copy and paste this, watch the formatting. Those are all suppose to be one long line of text for each INSERT INTO.

We now need to edit the /etc/httpd/conf/httpd.conf file. You will want to comment out the last line. It should look like this.

```
#Include /etc/trixbox/httpdconf/*
```

Now it is time to restart httpd and test our results. Don't worry if it does not work the only change you need to make to revert back is to remove the comment line from /etc/httpd/conf/httpd.conf and move the /etc/httpd/conf.d/auth_mysql.conf file somewher and restart httpd. To restart httpd issue the following command.

```
service httpd restart
```

Now goto your trixbox server from a web browser. Right away you should be presented with a login window, you can provide the credentials to any of the users you created and get in. Depending on whether that login is part of the admin group will determine if they can get into the maint section of the trixbox web page. If the login is in both groups, then that login can view both.

This setup will prevent a lot of unwanted scraping and hacking on the system and just generally protect the user interface from traffic. If your trixbox is forward facing on the internet, then locking down the user portal is, in my opinion, critical to the proper function of the system.

You might have noticed that in the database schema I included two tables that were not used. These are email and extension. These tables will be used in a upcoming document. If you do not want to use them, feel free to exclude them. However, I will be showing you how to integrate postfix and mysql at a later date that will tie it all together. It is also handy to put in the extension and email of the account created for tracking.

A little easier approach.

If you do not want to use mod_auth_mysql and want to stick with the current htpasswd setup then I will show you a few tweaks to lock down the user panel. The file in /etc/trixbox/httpdconf/trixbox.conf has a few settings for /admin/ and /maint/. We want to edit this file to include /user/. Open the file in your favorite linux editor and add the following.

```
#Password protect /var/www/html
<Directory /var/www/html>
AuthType Basic
AuthName "Restricted Area"
AuthUserFile /usr/local/apache/passwd/wwwpasswd
Require user maint engineertim
</Directory>
```

We now need to add the user engineertim to the AuthUserFile. Issue the following command on the command line.

```
htpasswd /usr/local/apache/passwd/wwwpasswd engineertim
```

It will ask for a password for this user and then confirm it by typing it again. You will now need to restart httpd for this change to take affect.

```
service httpd restart
```

Now try out your trixbox login. Be sure to close any existing browsers that is logged into the trixbox server. It might be required to clear cookies and cache as well. It should ask you right away for a username/password. Feel free to use a different user other than engineertim.

Thus concludes this guide. I hope to continue to add to this on a regular basis and hopefully with user input this doc will grow. This is just the start or the seed. I have also started a second document on monitoring and alerting that should be available soon. In it you will find out how to setup and install OSSEC, Nagios alerting and integration into trixbox, CACTI integration and alerting, and hopefully much more.

Thank you for taking the time to read this,
Engineer Tim