Setting Up etded on Linux

Overview

Wolfenstein Enemy Territory is a free standalone game by Splash Damage, id Software, and Activision.

This document describes how to set up the dedicated ET server (etded) on Linux in a secure and stable way. By that I mean the following:

etded is run by a non-priveleged user (not root)
etded is run from inside of a chroot envornment
etded is invoked in such a way that will will start up again when it crashes.

This document does not cover the actual server configuration since that information can be found in a variety of other places and is always a hotly debated topic.

STEP #1: chroot setup

The /sbin/chroot command changes the root directory in the environment of a process (and thereby all of its children) to whatever you specify. Therefore, when chrooting the etded process, you are restricting that process to a portion of the filesystem. This is useful in case there is an exploit found for the etded server that would allow the attacker to drop to a shell in which case the attacker would not have access to the entire filesystem on your server.

Create a basic directory structure:

```
mkdir -p /usr/local/enemy-territory/home/et
mkdir -p /usr/local/enemy-territory/etc
mkdir -p /usr/local/enemy-territory/lib
mkdir -p /usr/local/enemy-territory/bin
```

In order to run etded, you will need to copy all of the shared object files that etded.x86 depends on to your chroot directory.

```
cd /lib
cp libc.so.6 libm.so.6 ld-linux.so.2 libcrypt.so.1 libdl.so.2 libnsl.so.1 \
libnss_compat.so.2 libnss_dns.so.2 libnss_files.so.2 libresolv.so.2 \
libtermcap.so.2 /usr/local/enemy-territory/lib/
```

Next, you will need to copy a few system binaries in order to use a user account other than root inside of the chroot:

```
cp /bin/bash /bin/su /usr/local/enemy-territory/bin/
```

NOTE: If you are using RedHat, SuSE, or some other distribution that uses pam, you will probably have to build a new su binary instead of using the one that came with your system (or try to copy all of the pam bullshit to your chroot too). The source code for su can be found in the GNU coreutils package. Here's some instuctions for building a new su binary. Please do not email me for help with this, as I don't know, nor do I want to know about pam.

Now you will have to set up the etc directory in your chroot directory. The only file that really matters is the passwd. For example:

/usr/local/enemy-territory/etc/passwd:

```
root:x:0:0::/:/bin/bash
et:x:603:603:enemy territory server:/enemy-territory:/enemy-territory/etded.x86
```

OK, this deserves some explaination: The user's SHELL inside of the chroot environment is actually the etded.x86 binary. The user's home directory is set to /enemy-territory instead of /home/et because etded.x86 needs to be invoked from inside of the enemy-territory directory. We'll create a symbolic link later on.

It may also be necessary for etded to be able to resolve hostnames. You will already have the libraries copied, but you'll need configuration files for it to work:

```
cd /etc/
cp nsswitch.conf resolv.conf /usr/local/enemy-territory/etc/
```

That should do it! Now you are ready to install Enemy Territory. When running the installer, you should be prompted for which directory you want to install et into (default is /usr/local/games/enemy-territory I think). Change this to "/usr/local/enemy-territory/enemy-territory_2.55" (or whatever version number it is). When the installer prompts for the location to install links, use "/usr/local/enemy-territory/bin".

After installation, you will need to set up a couple of symbolic links:

```
cd /usr/local/enemy-territory
ln -s enemy-territory_2.55 enemy-territory
cd /usr/local/enemy-territory/enemy-territory
ln -s ../home/et/.etwolf .etwolf
```

By linking this way, it becomes very easy to install new versions of enemy-territory since you can just install to enemy-territory_VERSION and replace the two links. Note that the ../home/et/.etwolf directory doesn't exist yet, it will be added in the next step (you can make the symlink even though it doesn't exist yet).

STEP #2: The user account

Most Linux distributions come with some utility to add users to the system. You can use whatver tool you like to do this, but make sure the account has no logon privelege and has a home directory of /usr/local/enemy-territory (or whatever the chroot directory is from STEP #1).

Personally, I just edit add lines to my /etc/passwd and /etc/group files with a text editor like the following:

/etc/passwd:

```
et:x:603:603:enemy territory server:/usr/local/enemy-territory:
```

/etc/group:

```
et::603:
```

Now, you'll need to give the user access to the only directory they should be allowed to write into:

```
mkdir -p /usr/local/enemy-territory/home/et/.etwolf
chown et /usr/local/enemy-territory/home/et/
```

STEP #3: Startup Scripts

As of the time of writing, the current relase of etded (2.55) is very unstable and crashes at least daily for me. Splash Damage is working on a fix, but in the mean time, you practically NEED some type of process to keep starting etded.

Every Linux distribution seems to handle init scripts differently. One thing that most (if not all) have in common is that they have a /etc/rc.d/rc.local file that is executed last on startup, so I'll explain how to use that, but you'll probably want to stick to the conventions of your distribution if you can.

First, place the following bash script in your PATH (e.g. /usr/local/bin/etded.sh) [download]:

```
#!/bin/sh

# the chroot 'root' directory where your game server is installed
SERVER_ROOT=/usr/local/enemy-territory

# additional options that should be passed to the game server command line
# for example, "+exec server.cfg +set net_ip 127.0.0.1"
SERVER_OPTIONS="+exec server.cfg"

# the location of a file that the PID of the running server should be written
# to by this script
SERVER_PIDFILE=/var/run/etded.pid

# the FULL PATH to the server binary. this is used by /sbin/pidof in
# determining if the server is running.
SERVER_BIN=/usr/local/enemy-territory/enemy-territory/etded.x86

# the username for the account that the server should be run as
SERVER_USER=et

# the location of a file that the PID of this script should be stored in
PIDFILE=/var/run/etded.sh.pid;

echo $$ > $PIDFILE;
while [ 1 ]; do
if [ ! $(pidof $SERVER_BIN) ]; then
```

```
screen -d -m chroot $SERVER_ROOT su - \
$SERVER_USER $SERVER_OPTIONS
sleep 1
if [ ! $(pidof $SERVER_BIN) ]; then
echo "\
Could not find a PID for ${SERVER_BIN}!
If etded.x86 is actually running, update ${0}
so that it can properly detect the PID of ${SERVER_BIN}."
rm $PIDFILE
exit;
fi
echo `pidof $SERVER_BIN` > $SERVER_PIDFILE
fi
sleep 30;
done;
```

And edit it for your configuration if necessary. You will also have to make this script executable.

Example:

```
cd /usr/local/bin/ \
&& wget -O etded.sh http://tjw.org/etded/etded.sh.txt \
&& chmod a+x etded.sh
```

Next, place the following bash script where your rc files go (e.g. /etc/rc.d/init.d/rc.etded) [download]:

```
#!/bin/sh

SERVER_BIN=/usr/local/enemy-territory/enemy-territory/etded.x86
SERVER_PIDFILE=/var/run/etded.pid
SCRIPT_PIDFILE=/var/run/etded.sh.pid
SCRIPT_PATH=/usr/local/bin/etded.sh

etded_start() {
$SCRIPT_PATH &
}

etded_stop() {
if [ -r $SCRIPT_PIDFILE ]; then
kill `cat $SCRIPT_PIDFILE`
rm $SCRIPT_PIDFILE
fi
if [ -r $SERVER_PIDFILE ]; then
kill `cat $SERVER_PIDFILE`
rm $SERVER_PIDFILE
else
killall $SERVER_BIN
fi
}

etded_restart() {
etded_stop
sleep 3
etded_start
}

case "$1" in
'start')
etded_start
;;
'stop')
etded_stop
;;
'restart')
etded_restart
;;
*)
echo "usage $0 start|stop|restart"
esac
```

Finally, ensure that the command '/etc/rc.d/init.d/rc.etded start' is run when your machine is started by either placing the command in /etc/rc.d/rc.local, or by some other method your distribution uses.

Example:

```
cd /etc/rc.d/init.d \
&& wget -O rc.etded http://tjw.org/etded/rc.etded.txt \
&& chmod a+x rc.etded \
&& echo "/etc/rc.d/init.d/rc.etded start" >> /etc/rc.d/rc.local
```

You're done. To start etded run '/etc/rc.d/init.d/rc.etded start', to stop it, type '/etc/rc.d/init.d/rc.etded stop', or to restart it, type '/etc/rc.d/init.d/rc.etded restart'.

You can use screen to view the console of the running etded by running 'screen -r'. To get out of the screen type Ctrl-A-D, which will not kill the etded process.

FAQ

Note that the etded.sh file above makes reference to a server.cfg file. It will not start etded unless you create a server.cfg file or edit the etded.sh script. There are two places that etded.x86 will look for the server.cfg file: /usr/local/enemy-territory/enemy-territory/etmain/ and /usr/local/enemy-territory/home/et/.etwolf/etmain/. I recommend using the latter so it doesn't get lost if you change et versions.

```
> [root@maelstrom init.d]# /etc/rc.d/init.d/rc.etded start
> [root@maelstrom init.d]# Could not find a PID for
> /usr/local/enemy-territory/enemy-territory/etded.x86!
> If etded.x86 is actually running, update /usr/local/bin/etded.sh
> so that it can properly detect the PID of
> /usr/local/enemy-territory/enemy-territory/etded.x86.
```

This error is generated by my etded.sh script if it cannot detect the process
ID of the etded.x86 proccess it just attempted to start.

There's two possibilites for this error:
1) etded.x86 wouldn't start or shut itself down shortly after starting.
This could be because you have an error in your config file, or
possibly a permission error. You can tell if this is the case if
there is no etded.x86 process running on your system after you
run the "rc.etded start" command. To find out why etded.x86 won't
start, skip the init script and just try running the server
directly with the command
'chroot /usr/local/enemy-territory su - et +exec server.cfg'

2) There was a problem finding or writing the process ID, but the
etded.x86 process was actually started. If this is the case
make sure you have a /var/run directory on your system and also
make sure your /sbin/pidof executable works and is in root's PATH.

```
linux:/etc/init.d # chroot /usr/local/enemy-territory su - et +exec server.cfg
su: incorrect password
```

This is what happens when the su binary that you used at:
/usr/local/enemy-territory/bin/su
requires pam (Pluggable Authentication Modules) support. It seems that most
distributions now build su and the other coreutils with pam support. You can
try to figure out which files are needed by pam and copy them to the chroot,
or just take the easy way out and build your own su binary to use here.