1 A Newbie's Guide to Unix Security

So, you've just installed a brand spanking new Linux machine, and gotten it onto the internet, either via modem or some other form. You've heard things about security but haven't really had time to play with it. If you're here reading this article, its time to start securing your box - lets have some fun!

Some of you might be thinking, "Oh, but I only have a modem dialup to the internet, and besides, its got a dynamic IP address.". This is a bad attitude to have - with the current amount of automated scripts scanning networks, its only a matter of time.

It also isn't enough to say you have no important data - having your computer trashed and needing to reinstall isn't fun when a few simple steps might have stopped it. It's also possible that your computer could be used to launch attacks against others, and the authorities might come back to you and ask you why.

Note that this article will use examples from Debian, but most of the concepts will be the same across linux distributions, and a lot of it across unicies.

2 Evaluating Services

The first thing to do is to figure out what services you want to provide to people - in some cases, you might want to provide a webserver, get some email, or if its simply a desktop, nothing. Once you've decided what you want the box to do, its much easier to secure it than if you weren't sure what it should be doing.

Now its time to figure out what you currently have running on the box. There are several ways to do this.

- ps
- netstat
- strobe

To see a list of what processes are running, do the following. It is a good idea to become familiar with what is running on your computer, so you can notice any changes.

```
$ ps waux
```

To see what ports have something listening on, do the following.

```
$ sudo netstat --inet --listening --program
Password:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address     Foreign Address    State  PID/Program name
tcp        0      0 *:32768           *:*                LISTEN 326/rpc.statd
tcp        0      0 *:printer         *:*                LISTEN 393/lpd
tcp        0      0 *:dict            *:*                LISTEN 352/0
```

```
tcp        0      0 *:587               *:*                 LISTEN 559/sendmail: MTA:
tcp        0      0 *:netbios-ssn       *:*                 LISTEN 374/inetd
tcp        0      0 *:pop3              *:*                 LISTEN 374/inetd
tcp        0      0 *:imap2             *:*                 LISTEN 374/inetd
tcp        0      0 *:sunrpc            *:*                 LISTEN 161/portmap
tcp        0      0 *:www               *:*                 LISTEN 582/apache
```

The local address column lists the ip address and port that are being listened on. The state column lists what state the connection is in - the examples above are listening for connections. The PID/Program name column lists the process that is bound to the port.

To see what ports you can access over the network, you can use a port scanner. A popular one, called nmap, is easy to use and also has many options.

```
$ nmap localhost

Starting nmap V. 2.54BETA28 ( www.insecure.org/nmap/ )
Interesting ports on muon (127.0.0.1):
(The 1535 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
110/tcp    open        pop-3
```

Now that you know what ports are open, now you need to decide what you want to leave open. One important rule is that if you don't understand it, you probably don't need it. Ask around on mailing lists, IRC or your favourite support channels and find out if you require it. If you do not, disable the service, and then uninstall it.

To stop a service, find out which init script started it, and stop it via that. For example, if you have apache running you'd run a command like:

```
$ sudo /etc/init.d/apache stop
```

The other alternative is that the service is running from inetd, a program that spawns other programs. To remove services from there, edit the config file (/etc/inetd.conf) and comment out the offending service. Then restart inetd via your distributions means.

Now you should remove the package that contains the service - for details about how to do this, see your distributions documentation.

3 Seeing whats happening

It's also important to be able to tell what connections are currently open to your computer. This is yet another useful use of netstat. This is done by the following.

```
$ netstat --inet
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address       Foreign Address    State
tcp        0      0 192.168.1.1:32775   192.168.1.2:ssh    ESTABLISHED
tcp        0      0 192.168.1.1:32771   192.168.1.2:ssh    ESTABLISHED
tcp        0      0 192.168.1.1:ssh     192.168.1.2:1847   ESTABLISHED
tcp        0      0 192.168.1.1:www     192.168.1.2:1854   TIME_WAIT
```

4 Security Updates

Keeping track of your distribution's security updates is important - it will let you know if you're running vulnerable versions, and provide either fixes or workarounds. As these vulnerabilities have been published, it is important to fix them as soon as possible.

For Debian, there is a security-announce mailing list available at http://lists.debian.org/. You should really subscribe to it - it is not a high volume list, and contains important information. You can also look at previous security alerts at http://www.debian.org/security/.

To obtain updates via apt, add the following line to /etc/apt/sources.list:

```
deb http://security.debian.org/ stable/updates main
```

Then run the following commands - they will update the list of available packages, then show you what needs to be upgraded.

```
$ sudo apt-get update
$ sudo apt-get -u upgrade
```

Note that security updates for Debian are only guarenteed to be made for the stable distribution. Sometimes you see some for testing or unstable, but there are no guarentees.

For other distributions, please see their documentation or websites.

5 Conclusion

There are a few things to remember - securing a system isn't something thats absolute. There's no point where you can say that things are 100% secure - there's always bugs being found, and patches or configuration changes needed.

However, there becomes a point where any further effort put into securing a system costs more than it gains you - its simply a question of risks.

As you've seen, a basic review of what is installed and running, keeping track of your distributions security updates and watching logs goes a long way to securing your system.