HARDENING SSH - MAKING A GOOD THING BETTER

OpenSSH is every system administrators friend. It helps us to do so much, that we sometimes forget that it also needs to be configured securely. Assuming otherwise can lead to problems as people try to use it to gain access to your machine. Just because it is useful does not mean you can be lazy when you setup your SSH server. To help with that here are some guidelines for making it a bit more secure.

Where to Start?
As always, we start with the configuration file for the *sshd* process, it is called *sshd_config* and is generally found in */etc/ssh*;

Setting	*Set To*	*Reason*
Port	22	This is the port that the *sshd* server will listen on for connections
Protocol	2	This specifies that the server will only "*talk*" using ssh protocol 2
PermitRootLogin	no	Disallows root logins over ssh
IgnoreRhosts	yes	Disallows use of *.rhosts* or *.shosts* files
StrictModes	yes	This checks file modes and ownership of the user's files and home directory before accepting login
X11Forwarding	no	Disallows X11 forwarding
RhostsRSAAuthentication	no	Disallows use of *.rhosts* or */etc/hosts.equiv*
HostbasedAuthentication	no	Disallows use of *.rhosts* or */etc/hosts.equiv*
PermitEmptyPasswords	no	Does not allow users who have empty passwords to login
Ciphers	blowfish-cbc,aes256-cbc,aes256-ctr	Only use the strong encryption schemes
AllowUsers	bob jane andrew	You can use this option to explicitly specify which users can actually use the sshd server
DenyUsers	daemon bin sync adm lp shutdown halt mail news uucp nobody operator squid postgres gopher postfix xfs	Deny all system accounts and any other users who you may not want to have access
PasswordAuthentication	yes/no	If you set this option to "no" then the only way a user will be able to login will be via x509 certificates
Compression	yes/no	This might give a performance boost. Test and see.

What about Key Usage?
SSH has two options when you create a key - RSA and DSA. I personally prefer the RSA option as I believe it adds a bit more security then the DSA key. So when I create a key for my usage, I generally use the following command for a 2048 bit RSA key;

```
ssh-keygen -t rsa -b 2048
```

Also remember that if you use these keys for logging on to a server, then they need to be stored in the *authorized_keys2* file in the *.ssh* directory. Try to always use SSH protocol 2 when connecting to a server, if the

server offers protocol 1 and 2 you can force the use of protocol 2 by using the *-2* switch with your *ssh* command. For example;

```
ssh -2 bob@workserver.com
```

Final Words

Make sure SSH is properly configured and you will have no reason to regret using it. Remember that no matter how many features a piece of software has -even if they are security enhancing features-, all software needs to be securely setup. As always, have fun and learn.