

## Re: SSH as root

**Source:** [http://www.derkeiler.com/Mailing-Lists/securityfocus/Secure\\_Shell/2003-07/0024.html](http://www.derkeiler.com/Mailing-Lists/securityfocus/Secure_Shell/2003-07/0024.html)

---

**From:** Tim Greer (*chatmaster\_at\_charter.net*)

**Date:** 07/04/03

To: "Michael Coulter" <mjc@bitz.ca>  
Date: Thu, 3 Jul 2003 19:04:21 -0700

----- Original Message -----

From: "Michael Coulter" <mjc@bitz.ca>

To: "Tim Greer" <chatmaster@charter.net>

Cc: "Paul Bauer" <paul@shorttermwhat.com>; <secureshell@securityfocus.com>

Sent: Thursday, July 03, 2003 5:45 PM

Subject: Re: SSH as root

> *On Thu, Jul 03, 2003 at 05:31:17PM -0700, Tim Greer wrote:*  
> > *SSH keys can be a bad thing... But I suppose so could plain text*  
*passwords*  
> > *on a system if someone compromises it.*  
>  
> *Passwords are inferior to keys in at least 3 regards:*  
>  
> *- in the case of a MITM attack a password is compromised, a key is not*

Yes, but it doesn't require having a key on the server that could be compromised that allows someone to ssh into another server without any effort. Preventing a MITM attack is a good thing, for sure, and this is true anyway, if you the RSA key fingerprint to tell you it's changed. I guess I thought you were saying you could store the keys for logging right into the other server without prompting for a password. Yes? No? If no, (i.e., you didn't mean setting a password to not prompt for a password for logging into another server), then what I was commenting on my my reply doesn't relate to what you said and you may disregard it. You may know better what I had meant not, with that said. However, if it was what you meant and you don't agree, read on.

> *- in the case of the server being compromised the password is compromised,*  
a key is not

If they compromise a server, and the passphrase, etc. is there, they only need to get that data, or use it from those files and that server

> *- keys can be stored with a passphrase making it necessary to steal the*  
file

## SecurityFocus Secure Shell: Re: SSH as root

- > *itself as well as somehow obtain/bruteforce the passphrase, such as trojan'ing*
- > *the ssh client or keylogging*

If they compromise a server, and the passphrase, etc. is there, they only need to get that data, or use it from those files and that server.

- >
- > > *I don't recommend allowing for such ease, if someone manages to compromise*
- > > *one system and grab the file.*
- >
- > *Passphrases are a very good idea. However, if the client computer is compromised*
- > *you are in the same boat if you choose passwords, or keys with passphrases.*

Not if you don't have a password stored, they'd have to log what you type on the TTY you are logged in on. You have to be there to make it happen, with keys, you can have server after server compromised.

- > *The attacker needs to steal and file and capture the passphrase in the case of keys.*
- > *In the case of passwords they can just capture the password itself.*

Yes, if the server is compromised, but you'd also have to log into the compromised server and SSH into server B, for them to get the password for server B to access server B. You have a chance of noting something wrong on Server A before that happens. Had they stored the keys to log in without a password needed due to keys (this is what I'm speaking of, if you weren't, disregard my comments), then there's no need for them to wait, they just log into Server B and compromise it as well.

- > > *It's best to not use them from a security stand–point.*
- >
- > *I see nothing to support this, and several points to the contrary.*

You're entitled to your views on it, but for the reasons mentioned above, I don't agree—assuming you are saying what I thought. After all, once someone compromises a server, perhaps all bets are off for that server, including SSH'ing to another server, but unless you do, they log it and obtain that information before you find the server was compromised or know, then they aren't getting into Server B—that is a pretty good reason for my comment about it, I'd think.

Here's a page from a quick search on google to explain it (obviously I have failed to do this well):

[http://sourceforge.net/docman/display\\_doc.php?docid=761&group\\_id=1#securityprecaution](http://sourceforge.net/docman/display_doc.php?docid=761&group_id=1#securityprecaution)

## SecurityFocus Secure Shell: Re: SSH as root

"^ What security precautions need to be taken when using SSH keys? » |  
feedback | support

First and foremost, research the risks related to the use of SSH keys before you decide to make use of these mechanisms. Understanding how SSH works and the risks involved in key storage will help prepare you to take the proper security precautions.

You should never give out your SSH private key to anyone. SourceForge.net staff will never ask for your SSH private key data. It should be realized that only SSH public key data should be posted to remote hosts; private keys are for your local use alone.

SSH private keys may, at time of creation or after they have been created, have a password set on the key. This password will prevent use of the key until the password has been entered in to your SSH client. Some applications of SSH keys, such as automated host access by scripts, require the key holder to not set a password on that key. If you do not have a very good reason for doing this, you should set a password on your key.

The password you set on your SSH key should not match the password set on your SourceForge.net user account. Passwords should not be shared between different security systems or host networks. (i.e. you should pick a unique password for use on SourceForge.net and another unique password to protect your SSH key.)

Ensure that your private key has been placed only on machines over which you have direct control. Public, shared, and community machines are not suitable environments to store SSH private keys. Take action to help prevent theft of your SSH private key data. Setting a password on your SSH private key will help reduce the risks involved with private key theft.

## SecurityFocus Secure Shell: Re: SSH as root

Generate a key pair for each major location you work from. Keep the private key for that location only at that location. If your private key for a particular location is compromised, simply remove the corresponding public key from SourceForge.net, this preventing further use of that key pair to access SourceForge.net.

If you choose to make a backup of your SSH private key data, please ensure that any such backup is stored in a secure manner. This should be considered when guarding access to your private key data."

Again, if I misunderstood, disregard my response/comment and accept my apologies. If you did mean what I thought, perhaps this will illustrate what I meant. Obviously once someone gets a hold of the private key on the server compromised, they can just go and compromise each server there's a key for that has the password stored.

--  
Regards,  
Tim Greer chatmaster@charter.net  
Server administration, security, programming, consulting.