http://www.coincoin.fr.eu.org/?Your-SSH-keys-are-a-potential

Your SSH keys are a (potential) information leak

- 6- Webographie -

Date de mise en ligne : dimanche 7 février 2016

One of the things I've decided I want to do to improve my SSH security is to stop offering my keys to basically everything. Right now, I have a general keypair that I use on most machines ; as a result of using it so generally, I have it set up as my default identity and I offer it to everything I connect to. There's no particular reason for this, it's just the most convenient way to configure OpenSSH.

Some people will ask what the harm is in offering my public key to everything ; after all, it is a *public* key. Some services even publish the public key you've registered with them (Github is one example). You can certainly cite CVE-2016-0777 here, but there's a broader issue. Because of how the SSH protocol works, giving your SSH public key to someone is a potential information leak that they can use to conduct reconnaissance against your hosts.

As we've seen , when a SSH client connects to a server it sends the target username and then offers a series of public keys. If the current public key can be used to authenticate the username, the server will send back a challenge (to prove that you control the key) ; otherwise, it will send back a 'try the next one' message. So once you have some candidate usernames and some harvested public keys, you can probe other servers to see if the username and public key are valid. If they are valid, the server will send you a challenge (which you will have to fail, since you don't have the private key) ; if they are not, you will get a 'try the next one' message. When you get a challenge response from the server, you've learned both a valid username on the server and a potential key to target. In some situations, both of these are useful information.

(If the server rejects all your keys, it could be either that none of them are authorized keys for the account (at least from your IP) or that the username doesn't even exist.)

How do people get your SSH public keys if you offer them widely ? Well, by getting you to connect to a SSH server that has been altered to collect and log all of them. This server could be set up in the hopes that you'll accidentally connect to it through a name typo, or it could simply be set up to do something attractive ('ssh to my demo server to see ...') and then advertised.

(People have even set up demonstration servers specifically to show that keys leak. I believe this is usually done by looking up your Github username based on your public key.)

(Is this a big risk to me ? No, not particularly. But I like to make little security improvements every so often, partly just to gain experience with them. And I won't deny that CVE-2016-0777 has made me jumpy about this area.) (2 comments .)