

Re: sshd brute force attempts?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/questions/2006-09/msg01645.html>

- *From:* "Dan Mahoney, System Admin" <danm@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 20 Sep 2006 12:35:48 -0400 (EDT)
-

On Wed, 20 Sep 2006, Erik Norgaard wrote:

Dan Mahoney, System Admin wrote:

On Tue, 19 Sep 2006, Erik Norgaard wrote:

Along with some good advice. First of all: ssh is not a public service like http or smtp where you need anyone to be able to connect. So don't let them in the first place.

It is in this case. It's a web server that allows shell usage (and encourages it, as I actually advocate the power that comes with a shell as opposed to the primitive (and less secure) interface you may get with crap utilities like cpanel, or FTP (where you're at the mercy of the featureset of your particular app).

I think you misunderstood what I meant by public service, or maybe it wasn't clear: By a public service I mean a service available for anyone, even anonymously: You're not going to register the world to let people send mail to your server, (while you may (recommended) require authentication to send mail from your server).

Your ssh service should only be available to your users.

True enough, but so is/should pop3, and we're not having this problem there. Nor is there even an option for publickey auth (even though it uses PAM).

People can always manage access badly. Yes, you may not be sure of password protection on the keys, but the intruder first needs to get a copy of the key. If this is stored on a usb-stick the user carries with him, or only on systems that require local authentication first, then I think you're better off than password based ssh.

I think that people can better understand and manage a physical thing like a usb-stick and use that as their key. If the capacity is small enough, it is unlikely that people will use it for other

Re: sshd brute force attempts?

stuff and accidentally delete the key.

Yes, and then if/WHEN they do lose it, it's all the much MORE trouble to regenerate it and walk them through the motions of re-uploading it.

You may still find sshd login denied entries in your log – so what? it was denied! This is really only a problem if the traffics saturates your connection, or your log files grow so much that you run out of disk space.

It was denied, yes...but when it's denied for 200 different users from the same IP, it only takes one user with a weak password (and as much as I like keys, I personally prefer the passwords). I also find that since I have a nice web-enabled SSH app (as part of usermin), the key becomes sorta useless in that case.

As you read the article they had a password logger to see what passwords were attempted, quite interesting very very weak passwords. You can easily weed out bad password by running a cracker and forcing your users to change.

This is definitely "in the plan" — password crackers eat CPU like nobody's business so it would have to run "off site" but I've done this before with crack. I may try John this time.

I would like to find an alternative to passwd that can enforce a password policy, like min. 8 chars, upper AND lower case chars and numbers.

I've managed to very easily compile passwd against cracklib. Cracklib is in ports and easy to build — FreeBSD could use (but I haven't filed the requests) a) an option in make.conf to prevent passwd from getting built on a buildworld and b) the patched passwd/yppasswd tree in ports. If you want a few easy ports to maintain, these could be it :)

The article also comments on moving ssh to a different port, but this causes confusion and annoyance if you have many users and is non-standard. Doing the other things works great, an ssh-key on a usb-keyring is great.

For anyone savvy, yes. I don't assume that level of savvy.

Well, then – can't you also assume that people can use keys and understand that these should

Re: sshd brute force attempts?

be protected by passwords?

No, my assumption for the sake of simplicity has been to tell people "use this hostname for everything, and this ONE method of logging in should work for everything".

Yes, some of my more savvy users CAN set up keys. But for someone who wants the quick method to fix a few broken files, bad permissions, etc, it's far easier to tell them "get putty, log in..., and then cd to your homedir and type...".

I've been through this dance. "Get putty. Get puttygen. Now make a keyfile with options you really don't understand. Now find a way that, in the spirit of SSH you can upload that keyfile without using your password since I was told to disallow it...now password protect your key with something LONG and COMPLICATED when you can't even remember a password that you were emailed, and trusted your FTP app to remember...okay, now have that key with you everywhere you go (and you can't cheat and upload it to someplace like your xdrive.com or other service, you have to carry physical media. You understand all that? Okay, now cd to your homedir and type...

Personally, I created a script for parsing the delegated files from the different regional registries such as only to allow connection from EU countries.

Sounds interesting, is it public?

<http://www.daemonsecurity.com/pub/src/tools/cc-cidr.pl>

Thanks.

The output is just a list of cidr addresses that can be used in tables with packet filter. Or edit to create the output you want.

Thanks, will have a look.

-Dan

"We are basically...'Bandwidth Pimps'...Hrmmm...But that's cool man! You see these gold chains? It's all good!"

-Ali Dhoon
03/03/2003, 7PM

-----Dan Mahoney-----
Techie, Sysadmin, WebGeek

Re: sshd brute force attempts?

Re: sshd brute force attempts?

Gushi on efnet/undernet IRC
ICQ: 13735144 AIM: LarpGM
Site: <http://www.gushi.org>

freebsd-questions@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-questions>
To unsubscribe, send any mail to "freebsd-questions-unsubscribe@xxxxxxxxxxx"

Re: sshd brute force attempts?