

challenge, response, ssh, and a proposal

Source: <http://www.derkeiler.com/Mailing-Lists/FreeBSD-Security/2001-07/4111.html>

From: Dan Mahoney, System Admin (danm@prime.gushi.org)

Date: 07/12/01

Date: Thu, 12 Jul 2001 16:46:42 -0400 (EDT)
From: "Dan Mahoney, System Admin" <danm@prime.gushi.org>
To: security@freebsd.org, hackers@freebsd.org

Hey all...

I have a quick question that I've scoured the net for and really can't find the answer to.

It goes like this:

When using a challenge/response based auth scheme, like s/key, opie, or a cryptocard, one first gets the challenge, then enters the response as their password. The issue with this is that some of the protocols and clients don't seem to have a great facility for backpassing of the challenge to the user.

For example, with SecureCRT, which I've used, I cannot seem to find a place where it would hand you the challenge.

When doing ssh with opie, does it "let you in" enough to prompt for the response onscreen (rather than a dialog box?), or is this simply not supported.

On the same note, if one were to use OTP with ppp, is there a ppp stack out there that supports challenge/response, or must this be done via a terminal window?

What about pop3? ftp?

I mean, in THEORY, mail and ftp are less of a concern, as they can (SOMEWHAT) be limited to the running of arbitrary code as the user, but my situation is not one where one can set up ssh tunneling on five or six ports with non-standard clients. Enterprise networks have this option. I don't. My users should be free to use their choice of software. Their responsibilities are (at the moment) not to use weak passwords (I run crack regularly).

I'm working on implementing cryptocards on my system, and have a rather simple proposal, that avoids the problems I've outlined.

FreeBSD–Security: challenge, response, ssh, and a proposal

It goes a little something like this:

You load a webpage (http or https) (or ssh or telnet in, it doesn't matter). At this point you're presented with your challenge, you compute the response (or enter your s/key response). After that, you are handed a session password. The session password effectively becomes your system password, for any access FROM THAT IP for (A) Any access until a specified timeout (inactivity timeout) or (b) a hard–timeout or (c) until the user closes the webpage or initial auth session. Naturally, the webpage (or console app) would have the option to kill off the session key as well.

Of course, for any protocol that supports it (some ftpds can, su and telnet can, ssh, i'd love to know how it handles it from someone who's done it), you can still use your standard challenge/response system.

Given, this MIGHT require users to use user@realm (user@plaintext, user@opie, user@skey, user@cryptocard, user@session) authentication or something like that, but this is all pam–able, or should be.

Since cryptocards work via pam–radius it's easy enough to either add attributes to the radius server to handle this or to write a modified PAM, but in theory this should also be extensible to a pure–pam solution like opie and s/key.

This solves the problem of "no support in the client", and should be fairly easy to implement via PAM. It's also scalable to other things, like ibuttons and securid cards, and even biometrics (which are just really neat to make your geek friends all mouth–watery.)

I've BCCed my cryptocard sales rep on this email, but I'm interested in hearing any ideas anyone on this list has about (A) The possible problems and considerations (B) If there would be interest in such a project and (C) If there's been anything started related that I should know about. This SOUNDS like what Kerberos does, in a sense, but kerberos support isn't found much in clients either (the last client I used that supported it was NCSA telnet for the mac).

I look forward to hearing from anyone.

–Dan Mahoney

--

Pika Pika Pika!

–Pikachu, of Pokemon fame.

-----Dan Mahoney-----

Techie, Sysadmin, WebGeek

Gushi on efnet/undernet IRC

ICQ: 13735144 AIM: LarpGM

Web: <http://prime.qushi.org>

finger danm@prime.qushi.org

for pgp public key and tel#

To Unsubscribe: send mail to majordomo@FreeBSD.org

FreeBSD–Security: challenge, response, ssh, and a proposal

with "unsubscribe freebsd-security" in the body of the message