http://www.coincoin.fr.eu.org/?Some-notes-on-OpenSSH-s-optional

Some notes on OpenSSH's optional hostname canonicalization

- 6- Webographie -

Date de mise en ligne : mercredi 2 mars 2016

As I mentioned in my entry on how your SSH keys are a potential information leak , I want to stop offering my ssh public keys to all hosts and instead only offer them to our hosts. The fundamental reason that I wasn't doing this already is that I make heavy use of short hostnames, either entirely without a domain or with only our local subdomain (ie, hostnames like apps0 or comps0.cs). When you use short hostnames, OpenSSH's relatively limited 'Host ...' matching power means that it's easiest to just say : Host * IdentityFile ....

This has the effect that you offer your public keys to everything.

There are two ways to deal with this. First, you can use relatively complex Host matching. Second, you can punt by telling ssh to canonicalize the hostnames you typed on the command line to their full form and then matching on the full domain name. This has a number of side effects, of course ; for instance, you'll always record the full hostnames in your known_hosts file.

Hostname canonicalization is enabled with 'CanonicalizeHostname yes'. This can be in a selective stanza in your .ssh/config, so you can disallow it for certain hostname patterns ; for instance, you might want to do this for a few crucial hosts so that you aren't dependent on ssh's canonicalization process working right in order to talk to them. CanonicalDomains and CanonicalizeMaxDots are well documented in the ssh_config manpage ; the only tricky bit is that the former is space-separated, eg : CanonicalDomains sub.your.domain your.domain

The CanonicalizePermittedCNAMEs setting made me scratch my head initially, but it has to do with (internal) hostname aliases set up via DNS CNAMEs. We have some purely internal 'sandbox' networks in a .sandbox DNS namespace, and we have a number of CNAMEs for hosts in them in the internal DNS view of our normal subdomain, for both convenience and uniformity with their external names. In this situation, if I did 'ssh acname', OpenSSH would normally fail to canonicalize acname as a safety measure. By setting CanonicalizePermittedCNAMEs, I can tell OpenSSH that hosts in our subdomain pointing to .sandbox names is legitimate and expected. So I set up : CanonicalizePermittedCNAMEs *.sub.our.dom :*.sandbox,*.sub.our.dom

I don't know if explicitly specifying our normal subdomain as a valid CNAME target is required. I threw it in as a precaution and haven't tested it (partly because I didn't feel like fiddling with our DNS data just to find out).

Although it's not documented, OpenSSH appears to do its hostname canonicalization by doing direct DNS queries itself. This will presumably bypass any special nsswitch.conf settings you have for hostname lookups. Note that although OpenSSH is using DNS here, it only cares about the forward lookup (of name to IP), not what the reverse lookup of the eventual host's IP is.

I've been experimenting with having OpenSSH do this hostname canonicalization for a few weeks now. So far everything seems to have worked fine, and I haven't noticed any delays or hiccups in making new SSH connections (which was one of the things I was worried about). Of course we haven't had any DNS glitches or failures over that time, either (at least none we know about).

Sidebar : Why OpenSSH cares about CNAMEs during canonicalization (I think)

I assume that this is because if OpenSSH was willing to follow CNAMEs wherever they went, an attacker with a certain amount of access to your DNS zone could more or less silently redirect existing or new names in your domain

off to outside hosts. You would see the reassuring message of, say : Warning : Permanently added 'somehost.sub.your.domain' (RSA) to the list of known hosts.

but your connection would actually be going to otherhost.attacker.com because that's where the CNAME points.

You still get sort of the same issue if you don't have hostname canonicalization turned on (because then the system resolver will presumably be following that CNAME too), but then at least the message about adding keys doesn't explicitly claim that the hostname is in your domain. (3 comments .)