

## Re: Basics of key authentication

**Source:** <http://www.derkeiler.com/Newsgroups/comp.security.ssh/2004-07/0164.html>

---

**From:** Anne & Lynn Wheeler ([lynn\\_at\\_garlic.com](mailto:lynn_at_garlic.com))

**Date:** 07/25/04

Date: Sun, 25 Jul 2004 02:46:30 -0600

"OpticTygre" <[optictygre@adelphia.net](mailto:optictygre@adelphia.net)> writes:

- > *Ok, so everything I've read basically tells me the client creates a public*
- > *and private key. The public key gets copied to the server, and when the*
- > *client wants to log in, the server encrypts some message with the public*
- > *key, and the client decrypts it with its private key to prove he is who he*
- > *says he is. Is that right so far?*
- >
- > *Alright, if that's ok, then I have a few questions.*
- >
- > *1. A server can have tons of public keys stored on it. How does he know*
- > *which public key to encrypt the message with for the client?*
- >
- > *2. In the process of public / private key authentication for logins, what*
- > *is the order things are typically done? IE:*
- > *a. client says "hey, I want to connect"*
- > *b. client sends a message encrypted with private key*
- > *c. server decrypts through list of public keys*
- > *etc..... (I'm sure the above isn't right)*
- >
- > *In other words, what's the step-by-step process used for authenticating via*
- > *public/private keys between client and server? Thanks for helping to clear*
- > *things up.*

a radius scenario .... large percentage of ISPs around the world use radius as the standard mode of login by clients ... with userids and passwords. In the public key scenario ... the client registers a public key (in lieu of a password) and selects digital signature challenge/response authentication.

the client ppp connection code sends "login <their userid>" ... the server sends back some random challenge. the client combines the random challenge with some additional data ... and digitally signs it with their public key. the client returns the client contributed data and the digital signature to the server. the server takes the original random challenge, the client contributed data and uses the public key on file to validate the digital signature.

## comp.security.ssh: Re: Basics of key authentication

another is a kerberos scenario ... the dominant enterprise/campus authentication mechanism for windows and most open system platforms; again predominantly userid/password. the kerberos pk-init specification has a public key registered in lieu of a password and a digital signature challenge/response process used (process similar to the radius scenario).

part of the issue in the challenge/response authentication scenario ... is countermeasure against replay attacks ... where eversdropper records client's transmission and replays them at a later time as an impersonation attempt (i.e. the server always sends different challenge every time .... so the correct client responses would always be different & unique).

... basically, the client doesn't just say that they want to connect ... they client says that they want to connect as a specific entity/userid. the server then chooses the correct public key based on who the client is attempting to connect as. in the ssh case, it is found in the .ssh directory off the home directory of the userid (at the server) that the client is attempting to connect as. in the radius and kerberos scenarios ... it is a specialized database employed by those services.

... digital signature is a stylized process for using the private key for "encoding" a hash of the data ... with a corresponding digital signature verification process that uses the public key for "decoding" and checking the results (i.e. the server recalculates the hash of the same data and compares it against the result of "decoding" the digital signature).

in all the scenarios ... the connection is NOT being made as a non-differentiated, anonymous entity .... but as some specific entity known to the server. the server uses the entity specification in the connection authentication to select the appropriate public key.

for some additional discussion of digital signature authentication see the FIPS186-2 standard at the NIST site:

<http://csrc.nist.gov/cryptval/dss.htm>

--

Anne & Lynn Wheeler | <http://www.garlic.com/~lynn/>