BetaApple

February 25

CAC Login

by
on 2/25/2011 12:52 PM
Category: CAC

This document relies heavily on content already published by Apple, Inc, located at http://support.apple.com/kb/TA24244?viewlocale=en_US

This is specifically about enabling CAC login to your computer. The scripts and steps below will 'bind' your CAC to an existing Active Directory user account.

Part of the login process is to do a lookup for the expected user in a directory service such as Open Directory, LDAP, or Active Directory. The first and recommended method to link a smart card user with a record in a directory service is to add the hash of the public key to the user's directory record. This is the most convenient and most secure way of identifying a smart card user. The second method is to lookup the user based on values drawn from the email signing certificate as required for the US Federal Government smart card use.

Binding a CAC to a local directory domain record

A script is preinstalled to assist you in binding a smart card to a user's local directory domain record. This is /usr/sbin/sc_auth:

myhostname# /usr/sbin/sc_auth -h
Usage: sc_auth accept [-v] [-u user] [-k keyname] # by key on inserted card(s)

sc_auth accept [-v] [-u user] -h hash # by known pubkey hash
sc_auth remove [-v] [-u user] # remove all public keys for this user
sc_auth hash [-k keyname] # print hashes for keys on inserted card(s)

An example of the output from this for a US Department of Defense Common Access Card (CAC) is:

myhostname% sc_auth hash
01C2F20D8964BE7701B57B63B0A1795B8F2604C1 Identity Private Key
443F30C356E676F447CD4DA89F46CC0CCED19737 Email Signing Private Key
4845564C1F8C6B378C19B8F262CE422933CF1FD1 Email Encryption Private Key

To add a user to the local directory

myhostname% sudo sc_auth accept -u myuser -h 01C2F20D8964BE7701B57B63B0A1795B8F2604C1

...where "01C2F20D8964BE7701B57B63B0A1795B8F2604C1" is the hash for the key associated with the Identity Private Key. Refer to the script for further usage instructions. You will need to run this as a user authorized to modify the directory. In this example, any of the hash entries listed could have been used for associating the card to the account. If desired, more than one smart card can be associated with a single user account by running the script again with the hash from the additional card(s).

The script adds a field to the user's authentication_authority property. For example, after executing the command above, the authentication_authority property for the user looks like:

myhostname% dscl . -read /Users/myuser
...
"authentication_authority" = (";ShadowHash;", ";pubkeyhash;
01C2F20D8964BE7701B57B63B0A1795B8F2604C1");
...

One can immediately log in to a new session using the smart card.

Smart card login uses Open Directory for all of its user lookups, so any supported directory structure will function properly.

| | 0 Comment(s)

Comments

There are no comments for this post.