UniChrom security API implementation

The UniChrom security API (SECAPI) is the set of DLL function exported by service module (module implemented according UniChrom service specifications). Only the UniChrom services are analysed for presence of SECAPI entry points.
UniChrom security model is simplified UNIX-way approach.
Each persona is described by entities like

- Login name
- Full User name
- User ID
- Group ID
- User shell
- User password

Login name – unique string identifier for user among currently active security API. Login name can be similar to Full name (if it is acceptable)

Basic considerations:
All Pchar strings except crypts are UTF-8 encoded strings.
All Huser handles is the non-negative consequential number of user in current security context.
All UID and GID are non-negative numbers. UID's have to be unique in current security context.
User with UID =0 is the Super User (root), whose actions are not limited at all.
Users with same GID belong to the same security GROUP. There are three security groups:
GID_ROOT = 0, GID_BOSS = 1, GID_USER = 2.

During the initialisation UniChrom performs the following steps:

1. Initialises SECAPI
2. Tries to login with current system user login name and without password. This is simple mapping of local user to SECAPI – provided restrictions. It means the system already validated used right to login, so the service only determines the user access level for UniChrom. Login is performed by calling **seCheck**() function with current user name and empty password. Successful login on this stage bring the system to step 5.
3. If the step 2 failed the UniChrom displays login dialog. Dialog is filled with All possible user names when SECAPI module allows anonymous (without successful **seCheck**() call) enumeration of available user names. I.e. calling **seGetUser**() without successful login

should fail to prevent anonymous user enumeration.

4. UniChrom calls **seCheck**() with selected (typed-in) login name and password. Successful login starts the system, failed login asks again for username and password with notification of wrong credentials supplied to dialog. Cancel button in login dialog breaks UniChrom system startup.
5. UniChrom asks the SECAPI module for basic user information – UID, GID, Full Name and SHELL. UID, and GID are used for determination of user rights, FullName is attached to every GLP record, SHELL modified the user interface according to recommendations of System Administrator.
6. System Starts.

UniChrom SECAPI reference

```
function seInit():integer;stdcall;
```

Initialise SECAPI for specified module.

```
function seDone():integer;stdcall;
```

Finalisation of the SECAPI for specified module is performed upon UniChrom system shutdown.

```
function seFindUser(const pcUser:PChar;var
     ndxUser:HUser):integer;stdcall;
```

Find the system user with specified name and return its sequential number (ndxUser)

```
function seCheck(ndxUser:HUser;const pcPwd:PChar):integer;stdcall;
```

Check the password of specified user with the number ndxUser.

```
function seGetUserInfo(ndxUser:HUser;nEnt:integer;var
     nfo:variant):integer;stdcall;
```

Get the information about user with the specified number (ndxUser). Function should work only after successful call of **seCheck()**

Information type:

ENT_UID – UID, unique user identifier

ENT_GID – GID, group identifier

ENT_LOGIN – login name

ENT_NAME – Full user name

ENT_SHELL – User shell (text string)

ENT_CRYPT – returns nothing

```
function seGetUser(ndxUser:HUser;var buf;var
    buflen:integer):integer;stdcall;
```

Get the name of user specified by number (ndxuser). Buflen should contain the length of receiving buffer (buf). Function used for user enumeration. If anonymous enumeration is not acceptable, the function should fail until successful call of **seCheck().**

```
function seAddUser(const pcUser:PChar;var
    ndxUser:HUser):integer;stdcall;
```

Add the user with the specified login name (pcUser) and return its index (ndxUser). The function should work only after successful call of **seCheck()** for users with GID allowed to perform such operations.

```
function seDelUser(ndxUser:HUser):integer;stdcall;
```

Delete the user specified by number (ndxUser). The function should work only after successful call of **seCheck()** only for users with GID allowed to perform such operations.

```
function seSetUserInfo(ndxUser:HUser;nEnt:integer;const
    nfo:variant):integer;stdcall;
```

Update specified user specified by ndxUser with new entities. Function should work only after successful call of **seCheck()** only for users with GID allowed to perform such operations.
Information type:

ENT_UID – UID, unique user identifier

ENT_GID – GID, group identifier

ENT_LOGIN – login name

ENT_NAME – Full user name

ENT_SHELL – User shell (text string)

ENT_CRYPT – set the user encrypted password. UniChrom supplies the function with its own crypt, but the supplied string can be stored as-is or used for internal (in-module) calculation. The main rule – crypt string passed to **seSetUserInfo()** and then passed to **seCheck()** have to be accepted for the same user. Also during login **seCheck**(ndxUser,'cryprstring') UniChrom passes the same crypt string as one passed during **seSetUserInfo**(ndxUser,ENT_CRYPT,'crypstring').