The GPWSS Product Planning Document

This document is created for the United Computer Networking Professionals (UCNP) web site. Please see <http://www.ucnp.info> for more information.

Version history

Version	Date	Author	Comments
0.1	11.10.2010	mpan	First version. A GPWSS is based on the initial work done by mpan in 2009.
0.2	11.10.2010	mpan	Translation from Finnish to the English started.
0.9	11.10.2010	mpan	Translation from Finnish to the English ended. Some features clarified.
1.0	12.10.2010	mpan	Pseudocoding and defining GUI.
1.4	12.10.2010	mpan	Pseudocoding and defining GUI ended.
1.6	13.10.2010	mpan	Reviewed and approved version.
1.7	19.10.2010	mpan	The UCNP contact information updated. Notes concerning Random Password Generator (RPG) added.
1.9	19.10.2010	mpan	Reviewed and approved version.
2.0	20.10.2010	mpan	Final version. No changes to planning made. Notes clarified separately.
2.1	30.10.2010	mpan	Final and combined version. Notes with clarifications embedded.
2.2	30.10.2010	mpan	Reviewed and approved version.

Author(s): mpan.
Approved by GPWSS-info 13.10.2010, 19.10.2010, 30.10.2010 ucnp.info.
Contact information: ucnp@ucnp.info.

Table of Contents

1. A Generic Password Storage System (GPWSS)

1.1. Purpose

A Generic Password Storage System (GPWSS) can be build up for maintaining passwords. It is possible to use external equipment like USB memory sticks or mobile phones for this purpose.

1.2. Summary

The GPWSS will be implemented as a generic Java application. SHA-512 will be used as a cryptographic hash function (HASH). AES-256-CBC will be used as a encryption / decryption algorithm (Encrypt/Decrypt). A Random Password Generator (RPG) can be used to generate a password automatically. The Random Password Generator (RPG) can be based on for example genfipw.c copyright © by mpan in 1997 onwards. Passwords are handled via clipboard. Passwords will not be visible on display in cleartext format.

It is possible to use external equipment like USB memory sticks or mobile phones to store the GPWSS Java application and maintained passwords.

The "Generate Password" and the "Copy Password to Clipboard" features of the GPWSS product are used as protection against keystroke loggers (keyloggers) and scattered electromagnetic radiation. The keyboard accessibility features of the operating system i.e. the on-screen keyboard of the operating system may be used as protection against certain types of keystroke loggers but not against all types of scattered electromagnetic radiation. The implementation of an extra embedded software keyboard to the GPWSS product may be needed. In this case the pseudocode needs to be modified so that the software keyboard is added to every "Read from the keyboard" sentence. The Graphical User Interface needs to be modified so that there exists a Software Keyboard functionality for every "Give" and "Retype" field.

2. Pseudocode

2.1. Data Structures

1) Initialize global variables: PIN_OK = no, PIN=""
2) File handles: hash_file contains the SHA-512 cryptographic hash value of the PIN code and password_file contains the list of (PASSWORD_DESCRIPTION, PASSWORD)s encrypted with AES-256-CBC encryption / decryption algorithm and a PIN code
3) PASSWORDLIST is a list of (PASSWORD_DESCRIPTION, PASSWORD)s
4) A Random Password Generator (RPG) has two arguments: PASSWORD_LENGTH and PASSWORD_TYPE

5) The PASSWORD_LENGTH is a number of characters in a password. The PASSWORD_TYPE is a character set of a password. Following PASSWORD_TYPEs must be supported: numeric, alphanumeric and the printable characters of ASCII character set i.e. from 0x21 to 0x7E (33 to 126 decimal)

2.2. Procedure Main_Menu

1) Print a Main Menu:

- Log In
- Change PIN
- Maintain Passwords
- Copy Password to Clipboard
- Empty Clipboard
- Exit

2.3. Procedure Log_In

1) If (Read_PIN_HASH(hash_file) == error) then {Print_Warning(log_in) and PIN = Create_PIN} else PIN = Read_PIN
2) End of Procedure; Main_Menu

2.4. Function Create_PIN

1) Read PIN1 and PIN2 from the keyboard and print *-characters to the display instead
2) If (PIN1 != PIN2) then {Error(errors_in_PIN1_and_PIN2) and PIN_OK = no} else if (Check(PIN1) == error) then {Error(check_PIN_error) and PIN_OK = no} else {Save_PIN_HASH(PIN1, hash_file) and PIN_OK = yes}
3) End of Function; Return PIN1

2.5. Function Read_PIN

1) Read PIN from the keyboard and print *-characters to the display instead
2) If (HASH(PIN) != Read_PIN_HASH(hash_file)) then {Error(errors_in_PIN) and PIN_OK = no} else if (Check(PIN) == error) then {Error(check_PIN_error) and PIN_OK = no} else PIN_OK = yes
3) End of Function; Return PIN

2.6. Procedure Change_PIN

1) If (PIN_OK == no) then {Print_Warning(change_PIN) and Log_In}
2) Read PIN1_NEW and PIN2_NEW from the keyboard and print *-characters to the display instead

3) If (PIN1_NEW != PIN2_NEW) then Error(errors_in_PIN1_and_PIN2) else if (Check(PIN1_NEW) == error) then Error(check_PIN_error) else if (Save_Passwords(Encrypt(Decrypt(Read_Passwords(password_file), PIN), PIN1_NEW), password_file) != error) then {Save_PIN_HASH(PIN1_NEW, hash_file) and PIN = PIN1_NEW} else Error(save_password_file_error)
4) End of Procedure; Main_Menu

2.7. Procedure Maintain_Passwords

1) If (PIN_OK == no) then {Print_Warning(maintain_passwords) and Log_In}
2) PASSWORDLIST = Decrypt(Read_Passwords(password_file), PIN)
3) Print the values of PASSWORDLIST's PASSWORD_DESCRIPTION fields to the display
4) Read the PASSWORD_DESCRIPTION from the input field
5) If (Check(PASSWORD_DESCRIPTION) == error) then {Error(check_PASSWORD_DESCRIPTION_error) and goto 13}
6) if (the PASSWORD_DESCRIPTION is wanted to delete from the PASSWORDLIST) then {Delete(PASSWORD_DESCRIPTION, PASSWORDLIST) and goto 12}
7) if (the new or existing PASSWORD_DESCRIPTION is not wanted to save to the PASSWORDLIST) then goto 13
8) If (the PASSWORD_DESCRIPTION is in the PASSWORDLIST and a new PASSWORD_DESCRIPTION is wanted to save) then {Error(existing_PASSWORD_DESCRIPTION_error) and goto 13}
9) if (the password is wanted to generate automatically) then {Choose PASSWORD_LENGTH and PASSWORD_TYPE from the lists and PASSWORD1=RPG(PASSWORD_LENGTH, PASSWORD_TYPE) and PASSWORD2=PASSWORD1 and goto 11}
10) Read PASSWORD1 and PASSWORD2 from the keyboard and print *-characters to the display instead
11) if (PASSWORD1 != PASSWORD2) then Error(errors_in_PASSWORD1_and_PASSWORD2) else if (Check(PASSWORD1) == error) then Error(check_PASSWORD_error) else Save(PASSWORD_DESCRIPTION + PASSWORD1, PASSWORDLIST)
12) if (Save_Passwords(Encrypt(PASSWORDLIST, PIN), password_file) == error) then Error(save_password_file_error)
13) End of Procedure; Main_Menu

2.8. Procedure Copy_Password_to_Clipboard

1) If (PIN_OK == no) then {Print_Warning(copy_password_to_clipboard) and Log_In}
2) PASSWORDLIST = Decrypt(Read_Passwords(password_file), PIN)
3) Print the values of PASSWORDLIST's PASSWORD_DESCRIPTION fields to the display
4) Read the PASSWORD_DESCRIPTION from the input field
5) If (Check(PASSWORD_DESCRIPTION) == error) then {Error(check_PASSWORD_DESCRIPTION_error) and goto 7}
6) if (the password is wanted to copy to the Clipboard) then if (the

PASSWORD_DESCRIPTION is not in the PASSWORDLIST) then
Error(PASSWORD_DESCRIPTION_not_found) else Copy_to_Clipboard(PASSWORD)
7) End of Procedure; Main_Menu

2.9. Procedure Empty_Clipboard

1) If (PIN_OK == no) then {Print_Warning(empty_clipboard) and Log_In}
2) if (the Clipboard is wanted to empty) then Empty_Clipboard
3) End of Procedure; Main_Menu

2.10. Procedure Exit

1) if (the application is wanted to exit) then Exit

3. Graphical User Interface

3.1. Main Menu

Window title: Generic Password Storage System
Subtitle: Main

Please choose one of the following choices or press exit to quit:

+ Log In
+ Change PIN
+ Maintain Passwords
+ Copy Password to Clipboard
+ Empty Clipboard

Buttons: Exit (to Exit window)

3.2. Log In -> Create PIN

Window title: Generic Password Storage System
Subtitle: Main -> Log In

<Warning: log_in!>

Please create a Personal Identification Number for maintaining passwords:

Give a PIN: ********
Retype a PIN: ********

[Error: errors_in_PIN1_and_PIN2 or check_PIN_error or save_hash_file_error!]

Buttons: OK (do action, if success then to Main Menu window) and Cancel (to Main Menu window)

3.3. Log In -> Read PIN

Window title: Generic Password Storage System
Subtitle: Main -> Log In

Please give your Personal Identification Number for maintaining passwords:

Give a PIN: ********

[Error: errors_in_PIN or check_PIN_error!]

Buttons: OK (do action, if success then to Main Menu window) and Cancel (to Main Menu window)

3.4. Change PIN

Window title: Generic Password Storage System
Subtitle: Main -> Change PIN

<Warning: change_PIN!>

Buttons: OK (to Log In window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Change PIN

Please give a new Personal Identification Number for maintaining passwords:

Give a PIN: ********
Retype a PIN: ********

[Error: errors_in_PIN1_and_PIN2 or check_PIN_error or save_password_file_error or save_hash_file_error!]

Buttons: OK (do action, if success then to Main Menu window) and Cancel (to Main Menu window)

3.5. Maintain Passwords

Window title: Generic Password Storage System
Subtitle: Main -> Maintain Passwords

<Warning: maintain_passwords!>

Buttons: OK (to Log In window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Maintain Passwords

Browsable list: Values of PASSWORDLIST's PASSWORD_DESCRIPTION fields

Input field: Value of PASSWORDLIST's PASSWORD_DESCRIPTION field

[Error: check_PASSWORD_DESCRIPTION_error or
existing_PASSWORD_DESCRIPTION_error!]

Buttons: New (to Read Password window), Delete (to Delete Password window), Modify (to Read Password window) and Cancel (to Main Menu window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Maintain Passwords -> Read Password

Please give a new Password for the PASSWORD_DESCRIPTION:

Give a Password: ********
Retype a Password: ********

[Error: errors_in_PASSWORD1_and_PASSWORD2 or check_PASSWORD_error or save_password_file_error!]

Buttons: Generate Password (to Generate Password window), OK (do action, if success then to Maintain Passwords window) and Cancel (to Maintain Passwords window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Maintain Passwords -> Read Password -> Generate Password

Choose a Password length from a list: **
Choose a Password type from a list: **

[Error: errors_in_PASSWORD1_and_PASSWORD2 or check_PASSWORD_error or save_password_file_error!]

Buttons: OK (do action, if success then to Maintain Passwords window) and Cancel (to

Read Password window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Maintain Passwords -> Delete Password

Do you want to delete the PASSWORD_DESCRIPTION?

[Error: save_password_file_error!]

Buttons: Yes (do action, if success then to Maintain Passwords window) and No (to Maintain Passwords window)

3.6. Copy Password to Clipboard

Window title: Generic Password Storage System
Subtitle: Main -> Copy Password to Clipboard

<Warning: copy_password_to_clipboard!>

Buttons: OK (to Log In window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Copy Password to Clipboard

Browsable list: Values of PASSWORDLIST's PASSWORD_DESCRIPTION fields

Input field: Value of PASSWORDLIST's PASSWORD_DESCRIPTION field

[Error: check_PASSWORD_DESCRIPTION_error or PASSWORD_DESCRIPTION_not_found!]

Buttons: Copy to Clipboard (do action, if success then to Main Menu window) and Cancel (to Main Menu window)

3.7. Empty Clipboard

Window title: Generic Password Storage System
Subtitle: Main -> Empty Clipboard

<Warning: empty_clipboard!>

Buttons: OK (to Log In window)

***

Window title: Generic Password Storage System
Subtitle: Main -> Empty Clipboard

Do you want to empty the Clipboard?

[Error: empty_clipboard_error!]

Buttons: Yes (do action, if success then to Main Menu window) and No (to Main Menu window)

3.8. Exit

Window title: Generic Password Storage System
Subtitle: Main -> Exit

Do you want to exit the application?

[Error: exit_error!]

Buttons: Yes (do action, if success then exit) and No (to Main Menu window)

4. Implementation Issues

All major personal computer operating systems must be supported. These include Microsoft Windows, Mac OS X, GNU/Linux and FreeBSD. Because Java is a generic programming language this should not be a problem. Other operating system support is optional. In Future the support for smartphones etc. is needed.