Pepito

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
860	Tue Apr 3 16:16:32 CEST 2012		M

Contents

1 Discovered Vulnerabilities

1.1 Segfault on *NULL* password

When a NULL pointer is given to checkPassword, a segfault happens, which kill the server and therefore makes regular users unable to use it anymore.

See attached file "dos_on_password_null.patch" for patching

1.2 Stack Buffer Overflow on checkPassword

When the user provides a password longer than savePassword local var (currently 64), it will overwrite the program's memory, which allow various exploits such as modifiing the return value of checkPassword function. See "change_admin_password.py" script for proof of concept and live use, which change the admin password without any authentication.

This vulnerability is solved by using strncpy instead of strcpy, which will shrink down every submitted password to the max buffer lenght if exceeded.

See attached file "stackoverflow_on_checkPassword.patch" for patching

1.3 Stack Buffer Overflow on password change

When the user provides a new password longer than the hardcoded length (currently *512*), the current change*Password function will overwrite the memory just after the buffer, which can lead to modifiing the code execution and thus allow remote client to gain access to the server's memory and execution stack.

This vulnerability is solved by using strncpy instead of strcpy, which will shrink down every submitted password to the max buffer lenght if exceeded.

See attached file "stackoverflow_on_change_User-Admin_password.patch" for patching

1.4 Remote Code Execution (it is right?) on user command

If the remote user provides a command number below zero, the current server will go backward in memory and redirect the execution flow to the given pointer, which allow remote client to execute code located in the memory without any control.

For example, providing "-1" to the server as a command will force it to read the pointer located at *handlerTab[0] - (sizeof(handlerTab[0]) * 1) and execute the function located at this pointer like it was a regular Handler.

See attached file "remote_code_execution_on_cmd.patch" for patching

1.5 Format string attack with user entry

When the user send a command the sent password is checked by checkPassword and if it's a wrong password, the password is wrote on the error output and in the log file. But the fprintf function used to write on the error output is wrong formatted. The first argument can contain a printf-like string as "%s". For example, if you put a command as is "0 %s%s%s%s%s..." the printf will access the memory until the segfault. You can use "%n" command to write into the memory randomly...

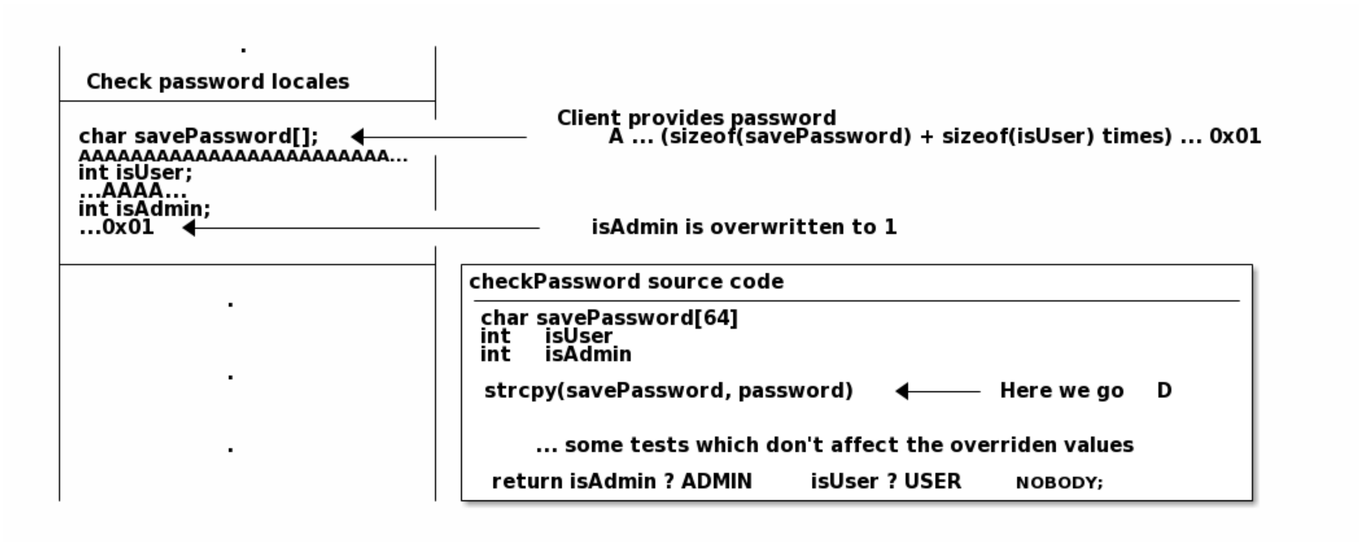
This vulnerablility is solved by use "%s" as first argument and message as second arguement witch remove printf-like strings.

See attached file "format_string_attack.patch" for patching

2 Exploits

2.1 Changing Admin Password without authentication

Using the SBO found on password checking function, we were able to overwrite the isAdmin local var and force returning ISADMIN so we are able to change the admin password to one of our choice.


Check password locales
char savePassword[];
AAAAAAAAAAAAAAAAAAAAAAAA...
int isUser;
...AAAA...
int isAdmin;
...0x01
Client provides password
A ... (sizeof(savePassword) + sizeof(isUser) times) ... 0x01
isAdmin is overwritten to 1
checkPassword source code
char savePassword[64]
int isUser
int isAdmin
strcpy(savePassword, password)
Here we go D
... some tests which don't affect the overriden values
return isAdmin ? ADMIN isUser ? USER NOBODY;