Programmed Insecurity – SySS Cracks Yet Another USB Flash Drive

***The SySS GmbH cracked the hardware-encrypted USB flash drive* ThumbDrive CRYPTO *from* Trek Technology.**

Dipl.-Inform. Matthias Deeg
Christian Eichelmann
Dipl.-Inform. Sebastian Schreiber

February 11, 2011

1 Introduction

At the end of 2009, the SySS GmbH found a severe security vulnerability in different USB flash drives with hardware-based AES encryption. By exploiting this security vulnerability, it was possible to gain unauthorized access to all protected data by just a few mouse clicks (see [1], [2] und [3]).

A recently performed security analysis of another USB flash drive with implemented hardware-based encryption showed that such critical security vulnerabilities are not at all a thing of the past.

2 Security Analysis

In the following section the example of a USB flash drive of the well-known manufacturer Trek Technology shows that programming errors can render an IT product that offers security by means of marketing actually insecure.

Concretely, the USB flash drive

- ThumbDrive CRYPTO [4]

was analyzed for security issues.

According to information provided by Trek Technology, the product version tested by the SySS GmbH is a customized version of the ThumbDrive CRYPTO USB flash drive which was customized for one special customer. The SySS GmbH could not verify this statement, as at the time this information was given to the SySS GmbH, there had already existed a product version of the USB flash drive in which the demonstrated security vulnerability had been fixed.

The following information can be found in the product description of this USB mass storage device:

> *ThumbDrive® CRYPTO ensures that 100% of the storage area is encrypted. With this 256-bit hardware AES engine, the ThumbDrive® CRYPTO offers one of the most advanced security solutions available today.*

In order to unlock the mass storage device and to access the protected data, the correct password for the user account `Administrator` has to be entered in the login dialog shown in figure 1.

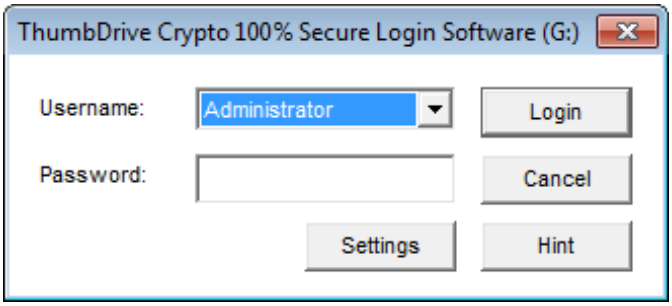


Figure 1: Password-based authentication

The administrative tools of the program `SecureLogin.exe`, which is stored on an emulated CDROM partition of the USB flash drive, can be used for setting the administrator's passwords, as figure 2 illustrates.

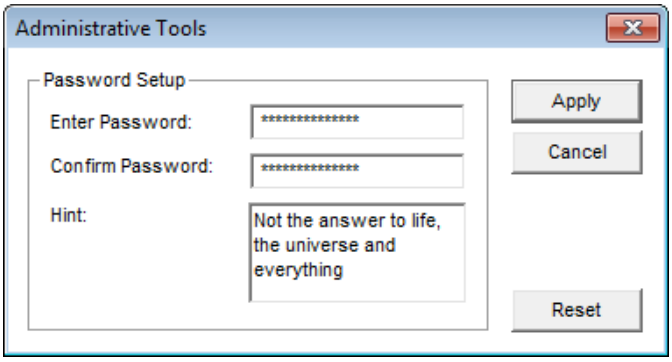


Figure 2: Administrative tools dialog

The used passwords have to meet the criteria of a hard-coded password policy and the maximum password length is restricted to 14 characters. Figure 3 shows the error message concerning weak passwords.

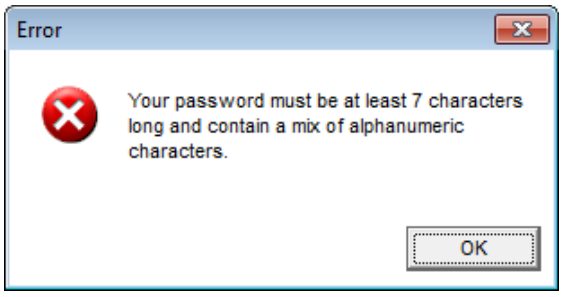


Figure 3: Error message concerning weak passwords

In the course of the performed security analysis, the SySS GmbH could find a severe security vulnerability in the password-based authentication of the TREK ThumbDrive CRYPTO USB flash drive.

The SySS GmbH found out that the program `SecureLogin.exe` encrypts the user input using the algorithm shown in figure 4.

```
.text:0042123A                jle     short copy_password
.text:0042123C                mov     al, [ebp+897Dh]             ; load encryption key (1 byte)
.text:00421242
.text:00421242 encrypt_password:                                  ; CODE XREF: sub_421170+E1↓j
.text:00421242                mov     cl, [esp+esi+120h+var_104]  ; load cleartext char
.text:00421246                add     cl, al                      ; add key value to char
.text:00421248                inc     esi                         ; point to next char
.text:00421249                not     cl                          ; generate bitwise complement (binary not)
.text:0042124B                mov     [esp+esi+120h+var_105], cl  ; store encrypted char
.text:0042124F                cmp     esi, ebx                    ; check if encryption is comleted
.text:00421251                jl      short encrypt_password      ; if not, encrypt next char
.text:00421253                jmp     short copy_password         ; else jump to copy routine
```

Figure 4: Annotated password encryption routine in the disassembler IDA Pro

The result of this encryption routine is then compared to a specific value, namely the correct encrypted password. Figure 5 shows this password comparison of 15 bytes (0Fh) at the address `0x40AAB8` during the runtime of the program `SecureLogin.exe` in the software debugger OllyDbg[1].

[1] `http://www.ollydbg.de/`

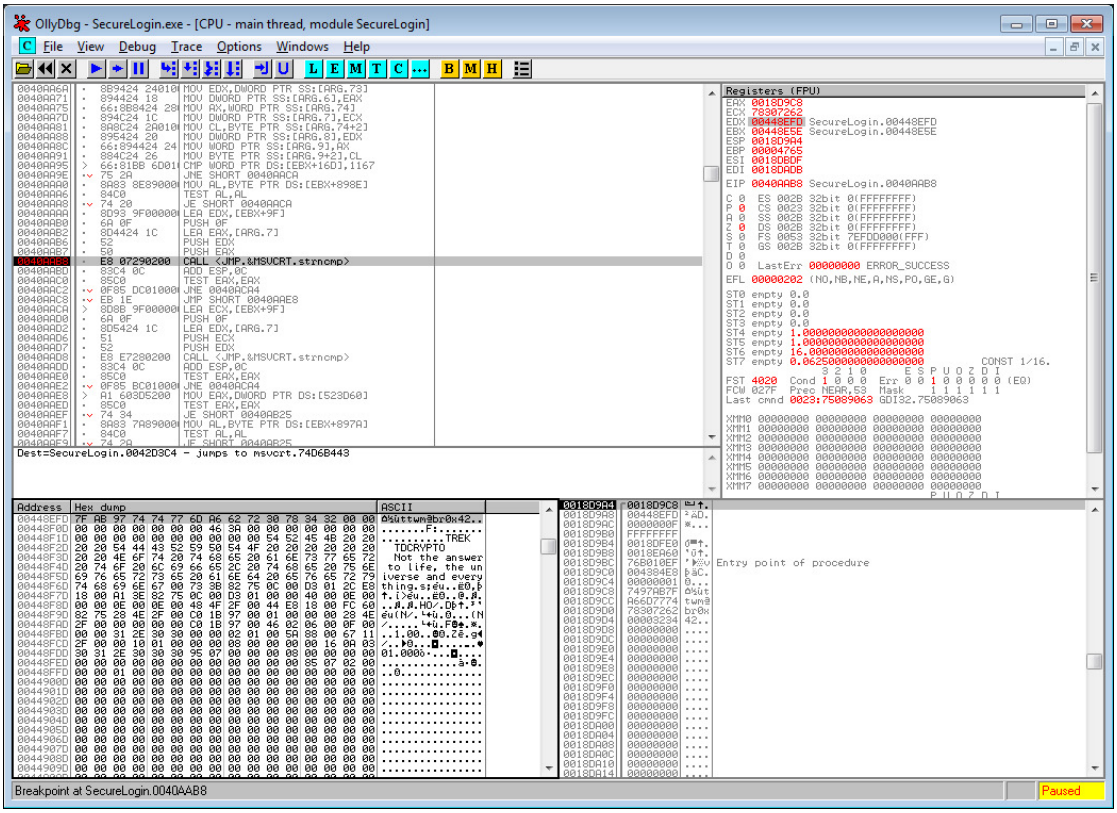


Figure 5: Password comparison in OllyDbg

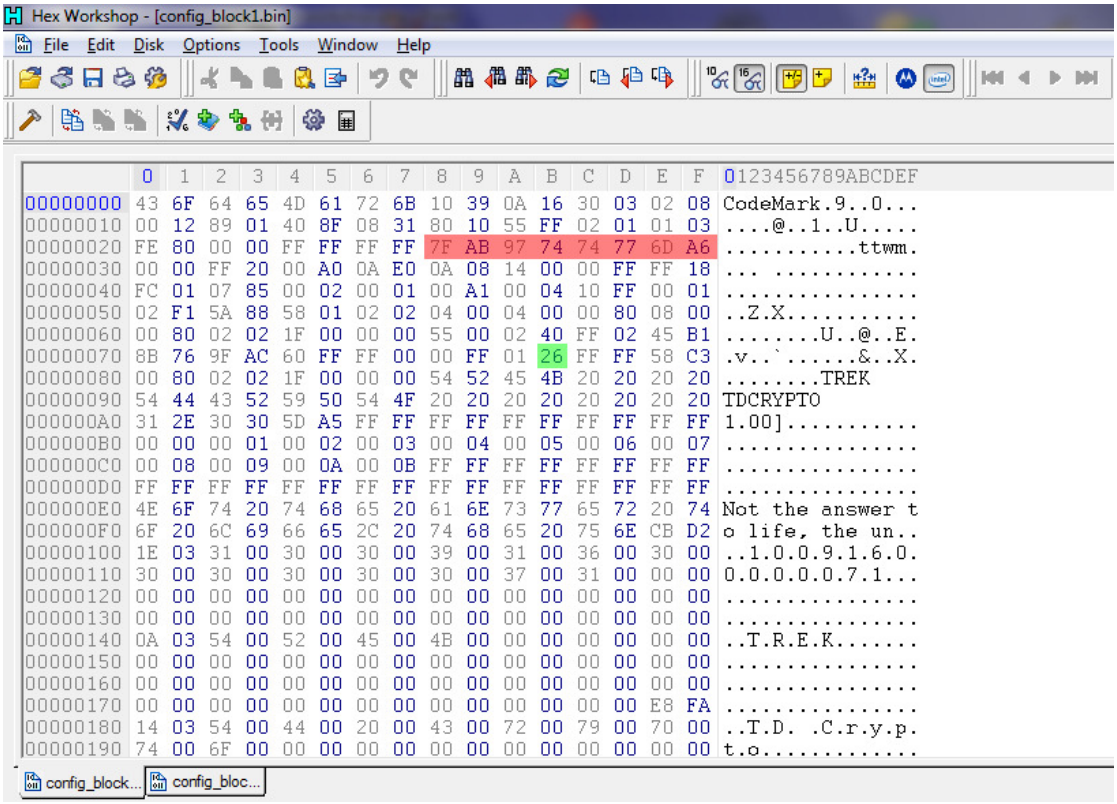
The annotated code of the password comparison is illustrated in figure 6.

```
.text:0040AAAA    lea     edx, [ebx+9Fh]          ; load address of correct encrypted password
.text:0040AAB0    push    0Fh                     ; MaxCount
.text:0040AAB2    lea     eax, [esp+62Ch+user_input] ; load address of user input
.text:0040AAB6    push    edx                     ; correct encrypted password
.text:0040AAB7    push    eax                     ; encrypted user input
.text:0040AAB8    call    strncmp                 ; compare strings
.text:0040AABD    add     esp, 0Ch
.text:0040AAC0    test    eax, eax
.text:0040AAC2    jnz     loc_40ACA4              ; bad guy jump
.text:0040AAC8    jmp     short loc_40AAE8        ; good guy jump
```

Figure 6: Annotated password comparison routine in the disassembler IDA Pro

A further analysis showed that the device configuration including the administrative password is stored in a special memory of the USB flash drive. When the program `SecureLogin.exe` is started, the device configuration is read from this memory using a controller-specific command. In each reading operation one 8K data block (8192 bytes) is copied from the USB flash drive to the host PC.

Figures 7 and 8 show the first few bytes of the two identified configuration blocks in which the administrative password can be found.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
00000000	43	6F	64	65	4D	61	72	6B	10	39	0A	16	30	03	02	08	CodeMark.9..0...
00000010	00	12	89	01	40	8F	08	31	80	10	55	FF	02	01	01	03	....@..1..U.....
00000020	FE	80	00	00	FF	FF	FF	FF	7F	AB	97	74	74	77	6D	A6	..........ttwm.
00000030	00	00	FF	20	00	A0	0A	E0	0A	08	14	00	00	FF	FF	18	... .
00000040	FC	01	07	85	00	02	00	01	00	A1	00	04	10	FF	00	01	....
00000050	02	F1	5A	88	58	01	02	02	04	00	04	00	00	80	08	00	..Z.X...........
00000060	00	80	02	02	1F	00	00	00	55	00	02	40	FF	02	45	B1	........U..@..E.
00000070	8B	76	9F	AC	60	FF	FF	00	00	FF	01	26	FF	FF	58	C3	.v..`.....&..X.
00000080	00	80	02	02	1F	00	00	00	54	52	45	4B	20	20	20	20	........TREK
00000090	54	44	43	52	59	50	54	4F	20	20	20	20	20	20	20	20	TDCRYPTO
000000A0	31	2E	30	30	5D	A5	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	1.00]...........
000000B0	00	00	00	01	00	02	00	03	00	04	00	05	00	06	00	07	....
000000C0	00	08	00	09	00	0A	00	0B	FF	FF	FF	FF	FF	FF	FF	FF	....
000000D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	....
000000E0	4E	6F	74	20	74	68	65	20	61	6E	73	77	65	72	20	74	Not the answer t
000000F0	6F	20	6C	69	66	65	2C	20	74	68	65	20	75	6E	CB	D2	o life, the un..
00000100	1E	03	31	00	30	00	30	00	39	00	31	00	36	00	30	00	..1.0.0.9.1.6.0.
00000110	30	00	30	00	30	00	30	00	30	00	37	00	31	00	00	00	0.0.0.0.0.7.1...
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000140	0A	03	54	00	52	00	45	00	4B	00	00	00	00	00	00	00	..T.R.E.K.......
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	E8	FA	....
00000180	14	03	54	00	44	00	20	00	43	00	72	00	79	00	70	00	..T.D. .C.r.y.p.
00000190	74	00	6F	00	00	00	00	00	00	00	00	00	00	00	00	00	t.o.............

Figure 7: Start of the configuration block 1 (8192 bytes)

The administrative password is stored in an encrypted manner (marked red) along with the used encryption key (marked green). To be precise, only the first eight characters of the password are encrypted (byte sequence `7FAB977474776DA6`), the remaining six characters are stored in plaintext (byte sequence `627230783432`, which is the ASCII string "`br0x42`").

As figure 4 illustrates, the used encryption algorithm is very simple and completely reversible in contrast to cryptographically secure one-way hash algorithms. The first 8 characters are encrypted by adding the value of the one byte long encryption key (`26h`) followed by a bitwise `not`-operation.

It is easy to see that the encrypted password can be decrypted by a bitwise `not`-operation followed by subtracting the value of the used encryption key as listing 1 shows.

Listing 1: Password decryption algorithm

```
// decrypt password
for (i = 0; i < 8; i++) {
    plaintext[i] = ~ciphertext[i] - key;
}
```

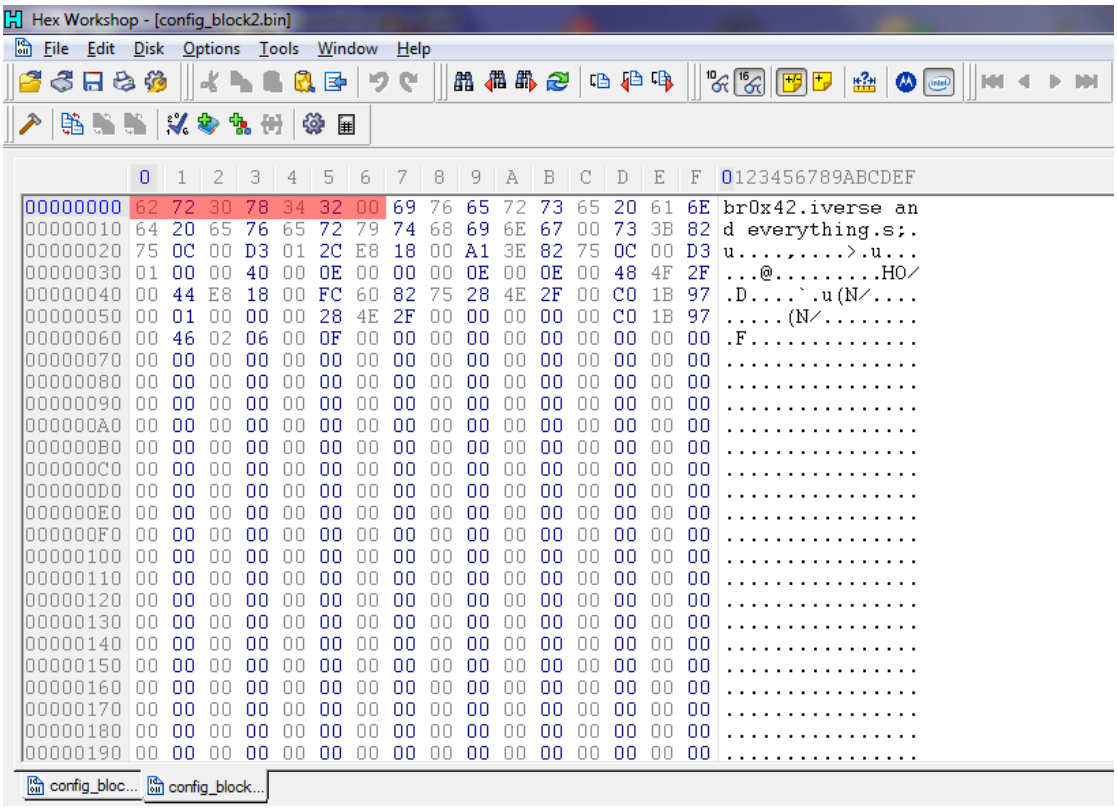


Figure 8: Start of the configuration block 2 (8192 bytes)

The encryption key is a random number between 1 and 254. A new encryption key is generated when a new password for the TREK ThumbDrive CRYPTO USB flash drive is set. The complete algorithm for the encryption key generation is shown in figure 9.

```
.text:00421201                push    0                       ; Time
.text:00421203                call    time                    ; get the current time
.text:00421208                push    eax                     ; Seed
.text:00421209                call    srand                   ; initialize PRNG
.text:0042120E                lea     eax, [esp+128h+var_110] ; load address of _ftime structure
.text:00421212                push    eax
.text:00421213                call    _ftime                  ; get the current time
.text:00421218                add     esp, 0Ch
.text:0042121B                xor     al, al                  ; set al to 0
.text:0042121D
.text:0042121D loc_42121D:                                    ; CODE XREF: sub_421170+BE↓j
.text:0042121D                test    al, al                  ; check if al is 0
.text:0042121F                jz      short loc_421225        ; jump, if it is
.text:00421221                cmp     al, 0FFh                ; check if al is 255 (0xff)
.text:00421223                jnz     short loc_421230        ; jump, if it's not
.text:00421225
.text:00421225 loc_421225:                                    ; CODE XREF: sub_421170+AF↑j
.text:00421225                call    rand                    ; call PRNG
.text:0042122A                add     al, [esp+120h+var_10C]  ; add value of _ftime structure to random number
.text:0042122E                jmp     short loc_42121D
.text:00421230 ; ---------------------------------------------------------------------------
.text:00421230
.text:00421230 loc_421230:                                    ; CODE XREF: sub_421170+B3↑j
.text:00421230                mov     [ebp+897Dh], al         ; store random number (= encryption key)
```

Figure 9: Annotated encryption key generation routine in IDA Pro

In the course of the security analysis, the SySS GmbH developed a *proof-of-concept* software tool for demonstration purposes. This software tool named ThumbDrive CRYPTO Unlocker extracts the correct administrative password and automatically unlocks the protected mass storage device of a TREK ThumbDrive CRYPTO USB flash drive with a single mouse click. Figure 10 shows this *proof-of-concept* software tool in action.

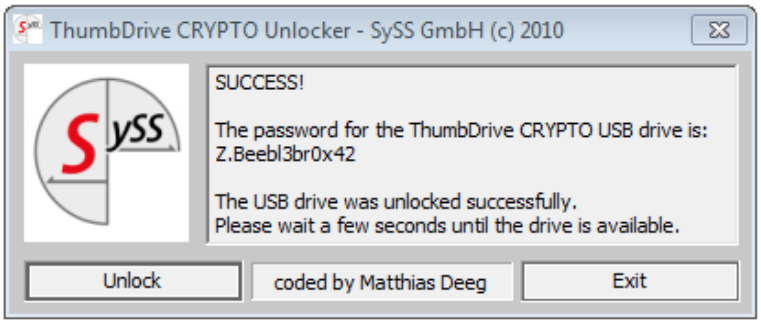


Figure 10: *Proof-of-concept* software tool ThumbDrive CRYPTO Unlocker

3 Conclusion

The SySS GmbH could once again demonstrate – using the example of the USB flash drive TREK ThumbDrive CRYPTO – that programming errors in the password-based authentication make it possible to gain access to all stored data by just a few mouse clicks fairly easily. If an appropriate software tool was available on the Internet, even technically inexperienced attackers could pose a security risk when getting hold of such a tool.

By exploiting the shown software vulnerability, implemented security features like the hardware-based 256-bit AES encryption and the hard-coded password policy are effectively rendered useless as they do not prevent the attack.

This test result shows that especially in the development of complex IT security products manufacturers have to exercise utmost care in high security standards in order to avoid critical security issues which lead the high security requirements ad absurdum.

The manufacturer Trek Technology was informed about the found security vulnerability by the SySS GmbH. Trek Technology responded quickly and fixed the demonstrated security flaw in an updated product version.

As mentioned before, according to information provided by Trek Technology, the product version tested by the SySS GmbH is a customized version of the ThumbDrive CRYPTO USB flash drive which was customized for one special customer. The SySS GmbH could not verify this statement as at the time this information was given to the

SySS GmbH, there had already existed a product version of the USB flash drive in which the demonstrated security vulnerability had been fixed.

References

[1] Jürgen Schmidt, *NIST-certified USB Flash drives with hardware encryption cracked*, http://www.h-online.com/security/news/item/NIST-certified-USB-Flash-drives-with-hardware-encryption-cracked-895308.html 2

[2] Matthias Deeg, Sebastian Schreiber, *SySS cracks* SanDisk *USB- Flash Drive* http://www.syss.de/fileadmin/ressources/040_veroeffentlichungen/dokumente/SySS_Cracks_SanDisk_USB_Flash_Drive.pdf 2

[3] Matthias Deeg, Sebastian Schreiber, *SySS cracks* Kingston *USB Flash Drive* http://www.syss.de/fileadmin/ressources/040_veroeffentlichungen/dokumente/SySS_Cracks_Kingston_USB_Flash_Drive.pdf 2

[4] Product information about TREK ThumbDrive CRYPTO, http://thumbdrive.com/cart/product.php?id_product=29 2