A Security Aspect Of USB Flash Drives

Version 1.1 September 2007

1. Forewords

Table of Contents

- 1. TakeMS
- 2. PEAK III
- 3. PEAK II
- 4. AlcorMicro

Have you ever asked yourself these questions?

- How safe is the password function of my USB flash drive (UFD)?
- What if I loose or forget my password, then what?
- If my UFD is lost or stolen, can someone else access my data?

If so, continue reading this document and you will hopefully get some answers!

This paper will focus on *four* different USB flash drives and the different software that is distributed with the UFD in question. It also includes a brief analysis of how safe they are. Or should I say "how unsafe they are?!". As the software that I have tested does not use encryption a simple patch may sometimes do the trick and provide us with the real password.

Sometimes those handy devices, which we rely on so much to keep our work portable and safe, are NOT always as safe as you would wish them to be. Using a ring3 debugger (OllyDbg) the communication between the protection software and the flash drive is easily intercepted. If the data, sent between the UFD and the computer, is just plain text, security could be totally compromised when monitoring the data via the debugger. This is both positive AND negative; the upside is that if you really have lost/forgotten your password, it MIGHT be retrievable (if you have to knowledge). On the downside, if someone wants to snoop around on your "protected" section of your UFD you could be, depending on choice of software, VERY poorly protected. As a special bonus I have decided to bundle this paper with my password recovery tool, the "UFD Password Revealer v1.2".

Enjoy your read, potassium / ARTeam



Disclaimers

All code included with this tutorial is free to use and modify; we only ask that you mention where you found it. This tutorial is also free to distribute in its current unaltered form, with all the included supplements.

All the commercial programs used within this document have been used only for the purpose of demonstrating the theories and methods described. No distribution of patched applications has been done under any media or host. The applications used were most of the times already been patched, and cracked versions were available since a lot of time. ARTeam or the authors of the paper cannot be considered responsible damages the companies holding rights on those programs. The scope of this tutorial as well as any other ARTeam tutorial is of sharing knowledge and teaching how to patch applications, how to bypass protections and generally speaking how to improve the RCE art. We are not releasing any cracked application.

Verification

ARTeam.esfv can be opened in the ARTeamESFVChecker to verify all files have been released by ARTeam and are unaltered. The ARTeamESFVChecker can be obtained in the release section of the ARTeam site: http://arteam.accessroot.com/releases/

Table of Contents

Verification	2
1. TakeMS – Protection? Where?	3
1.1 Hardware	3
1.2 Methodology	3
1.3 Patching protection application to reveal the password	5
2. PEAK III – Not much better.	7
2.1 Hardware	7
2.2 Methodology	7
2.3 Patching protection application to reveal the password	9
3. PEAK II – A Harder Shell	11
3.1 Hardware	11
3.2 Methodology	11
4. AlcorMicro – XOR:ed partial password storage	16
4.1 Hardware	16
4.2 Methodology	16
5. What if I'm not a hardcore reverser then?	20
6. Conclusions	20
7. Greetings	20
8. References	20
Document History	20

1. TakeMS - Protection? Where?

1.1. Hardware

First victim is a USB flash drive (UFD) from TakeMS (1 Gb, fig 1.1), which supports a public and a "secure" partition that was setup with the software that came with the device.



Fig 1.1 TakeMS 1 Gb stick

1.2. Methodology

The first UFD to be examined was the TakeMS stick. So load up the protection software included with the stick (CarryItEasy from cososys.com) and assign a password (ARTeam) and a password reminder (Who owns?) to the protected partition. Unplug and re-plug the UFD and re-run CarryItEasy. This time you will be asked to enter a password. See figure 1.2.

Carry it Easy - Login	×
You need to enter your password to start Carry it Easy!	
Password	Login Cancel
Password hint:Who Rules?	
To change your password go to the	Security Settings

Figure 1.2 Password dialog of CarryItEasy. Yeah, Who Rules? BGates? Nah.

Since this application launches a copy of itself in a temporary folder and re-launches with CreateProcess, we need to attach OllyDbg to the newly created process. So launch our good friend Olly!

Select process to attach				
Process	Name	Window	Path	
00001060 00001730 00000680 00000640 00000728 00000630 00000644 000000644 000000644	CarryItE PLBkMon ashMaiSv ashServ ashWebSv aswUpdSv ISUSPM KHALMNPR LUBPASPU	Carry it Easy - Login PLBkMon aswServ helper window KHALHPP_MainWindow	C:\DOCUME"1\roe\LOCALS"1\Temp\CarryItEasy\CarryItEasy.exe C:\Program Files\Alwil SoftB Flash Disk Utility\PLBKMOn.exe C:\Program Files\Alwil Software\Avast4\ashMaiSv.exe C:\Program Files\Alwil Software\Avast4\ashMeiSv.exe C:\Program Files\Alwil Software\Avast4\ashWebSv.exe C:\Program Files\Alwil Software\Avast4\ashWebSv.exe C:\Program Files\Alwil Software\Avast4\ashWebSv.exe C:\Program Files\Common Files\InstallShield\UpdateService\ISUSPM.exe C:\Program Files\Common Files\Logitech\KhalShared\KHALMMPR.EXE	
			Attach Cancel	

Figure 1.3 Attaching to CarryItEasy.exe

There it is. Press the "Attach" button and then press "F9" to continue running the application. Now, set a breakon-access bp on the code section of CarryItEasy and press the login dialog, you will now, hopefully, end up somewhere in the running code. Search for "All intermodular calls" and find the calls to DeviceloControl (for more info, consult MSDN) and set breakpoints on all of them.

🗱 BllyDbg - CarryItEasy.exe - [Found intermodular calls]			
R File <u>V</u> iew <u>D</u> ebug Plugins Options <u>W</u> indow <u>H</u> elp	- 8 ×		
► < × ► II + + + F F + → + L E M T W H C / K B R S	∷		
Address Disassembly Destination	~		
0048604F CALL [DWORD DS:<&USER32.DestroyWindow>] USER32.DestroyWindow			
004DD069[CALL_CDW0RD_DS:<&USER32.DestroyWindow>] USER32.DestroyWindow			
0042109F CHEL LDWURD DS:(&USER32.DestroyWindow)]USER32.DestroyWindow	E		
8048442A CALL [DWORD DS:<&kERNEL32.DeviceIoContrikernel32.DeviceIoControl			
004DE012 CALL [DWORD DS:<&USER32.DispatchMessage USER32.DispatchMessageW			
004E1717 CALL [DWORD DS:<&USER32.DispatchMessage USER32.DispatchMessageW			
004E936ELCHLL LUWURD DS:{&USER32.DIspatchmessage(USER32.DIspatchmessageW 004E936ELCHLL (UNP AWINEROUL Desugart State Property is a WUNEROUL COMPANY is a W			
094EIDDD CHLL (JNF.&WINSPOOL.DocumentPropertiesW) WINSPOOL.DocumentPropertiesW			
09450269[CHL] [DWDRD DS:<&USER32.DrawFocusRect>1]USER32.DrawFocusRect			
004650EB CALL [DWORD DS:<&USER32.DrawFocusRect>] USER32.DrawFocusRect			
004660AD_CALL [DWORD_DS:<&USER32.DrawTextExW>] USER32.DrawTextExW			
0042ED40 CALL [DWORD DS:<&USER32.DrawTextW>] USER32.DrawTextW			
00434338 CALL [DWORD DS:<&USER32.DrawTextW>] USER32.DrawTextW			
00485816 CHLL_LDWORD_DS:<&USER32.DrawlextW>] USER32.DrawlextW 00485816 CHLL_LDWORD_DS:<&USER32.DrawlextW>] USER32.DrawlextW			
004D940[CHLL [DWORD D3:\@U9ER3c.DFawTextW/] U9ER32.DFawTextW 004D940[CU] [DWORD D3:\@U9ER32.DrawTextW/] U9ER32.DrawTextW			
CAMPAGE CALL FOR DATE AND DE CALCERDA DE ANTENNAN A DECIDA DE ANTENNAN	<u>``</u>		
Break-on-access when executing (004DECC3)	Paused		

Figure 1.4 Setting breakpoints on DeviceloControl

With this done, return to the login dialog and enter any password e.g BGates :) and press the "login"-button. Now OllyDbg will break here:

💥 BllyDbg - CarryItEasy.exe - [CPU	- thread 00000650, module	e CarryltE]		
C File Yiew Debug Plugins Options Window Help				
	→ → L E M T W	HC/KBR····S	E . ?	
Biblication FFIS 28044450 CHLL CDWD 00484430 85C0 85C0 155 EARL 00484433 98564 FF000000 JE CarryIt CarryIt 00484438 8846 MOV ERX.ID D04 EARL D04 EARL 00484438 8846 MOV ERX.ID D04 EARL D04 EARL 00484438 8445 HOV ERX.ID D04 EARL D04 EARL 004844438 8459 D00 EARL D04 EARL D04 EARL 00484445 8459 D01 EARL HOVE EXX.ID D04 EARL 00484445 0FE448 HOVE EXX.ID HOVE EXX.ID D04 EARL 00484450 0E11 10 HOVE EXX.ID HOVE EXX.ID 00484450 0E14 0B12 INV.ID DRX.ID 00484450 0E16 INV.ID INV.ID DRX.EXX.EXX.EXX.EXX.EXX.EXX.EXX.EXX.EXX.E	105:(&KERNEL32.DeviceIoCont .00434537 00RD 05:ESI+41 TE DS:ERX+21 .00484596 BVTE 05:ERX+321 BVTE 05:ERX+321 00RD 05:ERX+321 00RD 05:ESI1	kernel32.DeviceloContro	Registers () () ERX 0004000 EXX 0004000 EXX 00090120 EXX 00050400 EXX 00050400 EXX 00056744 EXI 00056744 EXI 00056744 EXI 00066744 EXI 00066744 EXI 0006744 EXI 0006744 EXI 0006744 EXI 0007744 <	HHX)
Address Hex dunp 010E0043 2C 00 00 00 00 10 18 00 010E0043 2C 00 00 00 00 10 18 00 010E0043 2C 00 00 00 00 10 18 00 010E0053 06 00	INCLI Image: Constraint of the second	CICLETC 00000128 hevi 0104FED4 0004D004 10Con 0104FED5 01040043 10Son 0104FED5 0000250 10Bup 0104FED5 0000250 10Bup 0104FED5 0000250 10Bup 0104FED5 0000250 10Bup 0104FE55 0000250 0utb 0104FE55 00002500 0utb 0104FE55 00002050 0utb 0104FE55 0104FF34 00002500 0104FEF6 01025050 0104FF14 0104FE55 0005240 0005240	P 6 85 002 2 0 DS 002 2 0 DS 002 2 0 DS 002 2 0 DS 002 4 0 DS 00 4 0 D	S SODIE O(FFFFFFF) S SODIE O(FFFFFFFF)
Breakpoint at CarryItE.0048442A				Paused

Figure 1.5 Break at DeviceloControl with fake password

Follow the "InBuffer" in dump and you will see our input password "BGates". Press "F9" one time and break again on DeviceloControl. Now press "F8" and check the place where the text "BGates" was before! Now it displays your real password! ARTeam (of course)

🏶 BllyDbg - CarryltEasy.exe - [CPU - thread 00000650, module CarryltE]				
C File View Debug Plugins Options Window Help	_ Ə ×			
	H C / K B R S 🔚 📰 ?			
00484428 FF15 28A44F00 CALL EDWORD DS:<&KERNEL32.DeviceIoContr 00484430 8500 TEST EAX.EAX	kernel32.DeviceIoControl			
00484432 →0F84 FF000000 JE CarryItE.004844537 00484438 8846 04 00484438 8846 02 00484438 8448 02 00484438 8449 02 00484448 840 00484440 ~0F84 50010000 00484446 0FB648 32 00484446 0FB648 32 00484444 0FB740 3C 00484444 8816 00484444 8816 00484448 8816 00484443 816 00484443 884 00484450 CIE1 10 00484450 0FE1 0 00484450 0FE1 0 00484450 0FE1 10 00484450 0FE1 10 00484450 0FE1 10 00484450 0FE1 10 00484450 0FE1 10 004 0FE 00 00 0FEX FEX 00 0FEX 00 0FEX	EAA 00000001 Kernel32.7C801694 EDX 7C50EB94 ntdll.KlFastSystem EDX 0000001 ESP 0104FP0C ESP 0000250 ESI 0005F448 EDI 00006200 EIP 00484430 EIP 00484430 CarryItE.00484430			
EAX=00000001	P 0 CS 0023 32Dit 0(FFFFFFF) P 0 CS 0013 32Dit 0(FFFFFFF) P 0 SS 0023 32Dit 0(FFFFFFFF) Z 0 DS 0023 32Dit 0(FFFFFFFF)			
Address Hex dump ASCII	0104FF0C 00D6F590 UNICODE "BGates"			
0102005305 05 00	0104FF14 0006F448 0104FF10 00000000 0104FF20 00000000 0104FF24 004017CC RETURN to CarryItE.004917CC from CarryItE.0048 0104FF20 00000000 0104FF2C 000040004 0104FF2C 000040004 0104FF34 77048801 USER32.77D48801 0104FF34 77048801 USER32.77D48801 0104FF34 00006F448			
	Paused			

Figure 1.6 Break at DeviceloControl with the real password

Now we need not to know more. Remove the bp's and let the application run freely. Enter "ARTeam" as password and.. Voilá! You now have complete access to everything that resides inside the so-called "protected" partition. Now this will of course satisfy the needs for some reverse engineers. But I want to take it a step further. How about modifying CarryItEasy to show the real password instead of the password hint?!

1.3. Patching protection application to reveal the password

My thought was to modify the application CarryItEasy. When you press the "Password Reminder" button, the reminder, which is read from UFD during launch, is shown in the dialog window. Problem is that if we are going to reverse engineer this application we will certainly need more than one try at the reminder button and since it becomes hidden directly after pressing.. Ummm. It needs some improvements :) The following section of code hides the reminder button.

004DFD13	FF7424 04	PUSH	[DWORD	SS:ESP+4]
004DFD17	FF71 1C	PUSH	[DWORD	DS:ECX+1C]
004DFD1A	FF15 28A74F00	CALL	[DWORD	DS:<&USER32.ShowWindow>]
004DFD20	C2 0400	RETN	4	

Carry it I	Easy - Login	×
You need to enter your password to start	: Carry it Easy!	
) Password	Login Cancel
Password hint:Who Rules?	Password reminder	
To change your password go to the		Security Settings

Figure 1.7 Yeah, who rules? Button is still there :)

Now restart the CarryItEasy.exe and reattach OllyDbg, the breakpoints set earlier will still be there. Now, follow the "InBuffer" in dump just as before. When you see the "Who Rules?" text set a hardware breakpoint on write on the first char of the text. Continue to execute with "F9". Then you will break in kernel32.dll for a while and then return to CarryItEasy.exe, here:

00491C82	51	PUSH ECX
00491C83	56	PUSH ESI
00491C84	57	PUSH EDI
00491C85	50	PUSH EAX
00491C86	E8 D53CFFFF	CALL CarryItE.00485960
00491C8B	8B55 FC	MOV EDX, [DWORD SS:EBP-4]
00491C8E	8B7D 08	MOV EDI, [DWORD SS:EBP+8]
00491C91	03D2	ADD EDX,EDX
00491C93	8BCA	MOV ECX,EDX
00491C95	8BF0	MOV ESI,EAX
00491C97	8BC1	MOV EAX, ECX
00491C99	C1E9 02	SHR ECX,2
00491C9C	F3:A5	REP MOVS [DWORD ES:EDI], [DWORD DS:ESI]
00491C9E	8BC8	MOV ECX,EAX
00491CA0	83E1 03	AND ECX, 3
00491CA3	F3:A4	REP MOVS [BYTE ES:EDI], [BYTE DS:ESI]
00491CA5	8B7B 04	MOV EDI, [DWORD DS:EBX+4]
00491CA8	81C7 A0000000	ADD EDI,0A0
00491CAE	8955 FC	MOV [DWORD SS:EBP-4],EDX
00491CB1	75 04	JNZ SHORT CarryItE.00491CB7
00491CB3	33C0	XOR EAX,EAX
00491CB5	EB 29	JMP SHORT CarryItE.00491CE0
00491CB7	57	PUSH EDI
00491CB8	FF15 24A44F00	CALL [DWORD DS:<&KERNEL32.lstrlenA>
00491CBE	8BF0	MOV ESI,EAX

Registers:

EDX 010E14F9 ASCII "ho Rules?" ESP 0104FEF0 UNICODE "ARTeam" EDI 010E14F8 ASCII "Who Rules?"

As you can plainly see the correct password is currently stored as pointer to a UNICODE string in ESP. Setting a breakpoint @ 00491C82 reveals something interesting. The call at 00491C86 converts the password in ASCII format to UNICODE format, which suits us just fine :). At 00491CA8 something of interest caught my eye. EDI is a pointer to the "InBuffer" (read from UFD), adding 0xA0h to the starting point of the buffer will point to the string "Who Rules?" and then the code goes on in similar fashion, convert ASCII to UNICODE etc. Changing the ADD EDI, 0A0 to ADD EDI,092, just like previous procedure above, will then point to the ASCII string "ARTeam" and convert it to a UNICODE string. Now pressing of the reminder button is so much nicer. :D Make things easy on yourself now, write the one-byte patch to disk to make things permanent and you're all done!

Carry it Easy - Login	×
You need to enter your password to start Carry it Easy!	
Password	Login Cancel
Password hint:ARTeam	
To change your password go to the	Security Settings

Figure 1.8 The real password is now pwnd!!