PE Infection – How to Inject a dll



Nightmare - K@53

In association with



www.mihanit.net

Thank you to my friends who help me in this research

(K053,Heli, L U C I F E R(Bl4ck_Ic3))

Author: Nightmare(BioHazard)

Date: 03.05.2009(88.02.13)

Hint : Only the reader is responsible for any abuse of context of this document.

Introduction:

Our goal is to infect a Portable Executable file with a DLL. For that we need to obtain information on the structure PE.

So first we will discuss about PE Structure which is contain *PE Sections & Add a Section(Manually & Automaticly)*.

Then we have a brief discussion of the *Import Table*.

Finally, we will discuss how to inject a DLL

PE section(s):

00300000	00020000			
00400000	00001000	firefox(PE_header
00401000	00001000	firefox(.text	çode
00402000	00001000	firefox(.rdata	imports
00403000	00001000	firefox(.data	data
00404000	00048000	firefox(.rsrc	resources
0044C000	00001000	firefox(.reloc	relocations
00450000	00009000			
00490000	00009000			
00550000	00002000			
00560000	00103000			
00670000	00104000			
00970000	00001000			
00310000	00003000			
ипнииии	00008000			
00H10000	00002000			
00H50000	00004000			
00820000	00001000	swpg		PEheader
00851000	00012000	swpg	CODE	code
00863000	00001000	swpg	DHTH	data
00864000	00001000	swpg	BSS	
00865000	00001000	swpg	.idata	IMPORTS
иивеении	00001000	swpg	.reloc	relocations
00B67000	00001000	swpg	.rsrc	resources
00870000	00004000			
01210000	00002000			
54000000	00001000	klg	0005	PE header
54001000	00012000	KI9	CODE	code
54013000	00001000	K19	DHIH	data
54014000	00001000	K19	BSS	
54015000	00001000	K19	.Idata	IMPORTS
54016000	00001000	K19	.reloc	relocations
54017000	00001000	кıg	.rsrc	resources

As you can see a Portable Executable contains ".text, .rdata, .data, .rsrc, reloc, BSS" are PE sections which has been generated by windows loader.

".text" stores main code of a PE.

".rdata" contains read only data such as String literals , Debug Directory &

".data" stores all static data and initialized global.

"BSS" contains all uninitialized global.

".rsrc & .reloc" store details for relocating the image while loading

Now we need to know how to add a section into a PE. At first I'd like to do it in a hard way that needs a little Patience, some neurons in your head and a good Hex-Editor J.

First we have to gather the last section's information, so open our $target(firefox \mathbf{J})$ in a hex editor program and find *.reloc*.



So that the first 8 bytes for the name of the section. The next 4bytes are shown is Virtual Size. The next 4bytes are virtual address. The next 4bytes are Raw Size. The next 4bytes are Raw Offset then we have 12bytes which is nulled , then 4bytes for flags. And one important point is that our section alignment is 1000h.

Now we will add a section with 1000h Vsize and Rsize(it's not related to section alignment). Now we need to calculate our new section's information.

```
New Virtual Address = Virtual Address + Virtual Size + Section Alignment

// = 0004C000 + 00000506 + 1000 = 0004D506 -> 06 D5 04 00 (We use nearer 0004D000 > 00 D0 04

00)

New Raw Offset = Raw Offset + Raw Size

// = 00049600 + 600 = 00049C00 -> 00 9C 04 00

Virtual Size and Raw size for the new section = 1000h -> 00 10 00 00

Flags = E00000E0 -> E0 00 00 E0
```

So we have calculated our new section's information , Now we have to add them to PE with Hex Editor

00000268	00	00	00	00	40	00	00	40	2E	72	65	6C	6F	63	00	00	06	05	00	00	00	CO	@@.reloc
0000027E	04	00	00	06	00	00	00	96	04	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000294	40	00	00	42	2E	42	46	4E	49	47	48	54	00	10	00	00	00	DO	04	00	00	10	<pre>@B.BFNIGHT</pre>
000002AA	00	00	00	-9C	04	00	00	00	00	00	00	00	00	00	00	00	00	00	ΕO	00	00	ΕO	

*Tip: Vsize = Virtual Size , Rsize = Raw Size

Then insert 1000h bytes to the target.



Now check the target with *LordPE* then change *NumberofSection* to 6. After that you can realize that our new section has been added.

me VO	feet	VSize	BOffset	BSize	Flags		
d 000	01000	000000EE	5 00000400	00001000	60000020		
ata 000	02000	0000095	6 00001400	0000QA00	40000040		
ta 000	03000	0000038	4 00001E00	00000200	C0000040		
c 000	04000	0004753	8 00002000	00047600	40000040		
OC UUL NIGHT OOC	40000 40000	0000100	0 00049600	00001000	42000040 E00000E0		
•• I 1			00000	000 0000	000 000		PE Editor
Isystem			00000			00000	Break & Ent
			00000	1004 00000 1164 48580	000 000	00000	Rebuild PE
Atoprophysics 20	ustem32\-	eve some				0000	and the second sec
🗞 \systemroot\: کې \??\d:\windo	ystem32\: ws\systen	smss.exe n32\csrss.e	xe 00000	1E0 4A68	000 000	05000	Unsplit
•Systemroot\: •Sy	ystem32\: ws\systen d:\progi	smss.exe n32\csrss.e ram files'	xe 00000 \mozilla firefox\	1EO 4A680 firefox(analy	000 000 /ze).exe	05000	Unsplit Dumper Serv
 Systemroot\: \??\d:\windo PE Editor] - Basic PE Heade 	ystem32\: ws\systen d:\progi r Informat	smss.exe n32\csrss.e ram files' ion	xe 00000	firefox(analy	000 000 /ze).exe	05000	Unsplit Dumper Serv Options
(systemroot): (system	ystem32\: ws\system d:\progr r Informat	smss.exe n32\csrss.e ram files ion n1540	xe 00000 Amozilla firefox\	firefox(anal)	0000 000 rze).exe	05000	Unsplit Dumper Serv Options
(systemroot): (systemroot): (systemroot): (PE Editor] - Basic PE Head EntryPoint: ImagePage:	ystem32\/ ws\system d:\progr r Informat 0000	smss.exe n32\csrss.e ram files' ion 015A0 00000	xe 00000 mozilla firefox\ Subsystem: NumberOfSections	firefox(anal)	0000 000 rze).exe 02 06	05000	Unsplit Dumper Serv Options About
 Systemrooth: Systemrooth: Arrow Arrow Arrow PE Editor] - Basic PE Heade EntryPoint: ImageBase: SizeOffmage: 	ystem32\ ws\system d:\progr r Informat 0000 0004	smss.exe n32\csrss.e ion 01540 00000 4C506	xe 00000 xe 00000 Subsystem: NumberOfSections TimeDateStamp:	firefox(anal)	0000 000 rze).exe 02 06	O5000	Unsplit Dumper Serv Options About
 \systemroot\; \??\d:\windo PE Editor] - Basic PE Heade EntryPoint: ImageBase: SizeOfImage: BaseOfCode: 	ystem32\: ws\system d:\progr r Informat 0000 0004 0000	smss.exe n32\csrss.e ion 01540 00000 4C506 01000	Subsystem: NumberOfSections TimeDateStamp: SizeOfHeaders:	firefox(anal)	0000 000 /ze).exe 02 06 1D 00 ? +	05000 OK Save Sections Directories	Unsplit Dumper Serv Options About Exit
 Systemrooths Systemrooths PE Editor] - Basic PE Heade EntryPoint: ImageBase: SizeOfImage: BaseOfCode: BaseOfData: 	ystem32\ ws\system d: \progu 0004 0004 0000 0000 0000	smss.exe n32\csrss.e ion 01540 00000 4C506 01000 02000	xe 00000 xe 00000 Subsystem: NumberOfSections TimeDateStamp: SizeOfHeaders: Characteristics:	firefox(anal)	0000 000 rze).exe 02 06 0 10 00 ? + 02	O5000 OK Save Sections Directories	Unsplit Dumper Serv Options About Exit
 Systemrooths Arychain and a second seco	ystem32\ ws\system r Informat 0004 0004 0000 0000 t: 0000	smss.exe n32\csrss.e ion 01540 00000 4C506 01000 02000 01000	Subsystem: NumberOfSections TimeDateStamp: SizeOfHeaders: Characteristics: Checksum:	firefox(anal)	0000 000 72e).exe 02 06 1D 00 ? + 02 28 ?	05000 OK Save Sections Directories FLC	Unsplit Dumper Serv Options About Exit
 Systemrooths Arystemrooths Arystemrooths Basic PE Heade EntryPoint: ImageBase: SizeOfImage: BaseOfCode: BaseOfData: SectionAlignment: 	ystem32\ ws\system t: \progu 0000 0000 0000 0000 0000 0000 t: 0000 0000	smss.exe n32\csrss.e ion 015A0 00000 4C506 01000 02000 01000 00200	Subsystem: NumberOfSections TimeDateStamp: SizeOfHeaders: Characteristics: Checksum: SizeOfOptionalHea	firefox(anal) 000 49F109 000004 01 0004DF der: 00	0000 000 72e).exe 02 06 0 10 0 00 ? + 02 28 ? E0	O5000 OK Save Sections Directories FLC TDSC	Unsplit Dumper Serv Options About Exit

Congratulation you have added a new section to the program by your own hand J.

Now we want to add .BFNIGHT section in an easy way that needs only a good PE Editor like LordPE.

Open the target with lordPE then click the Section button, next right click and after that you can see that we have a option which is called *Add Section Header*.

A new section will be added by this option, So click it then you have to edit the new section's information with *Edit Section Header* like this.

[Section T	able]	I nake constant to a	day 1	X	
Name	VOffset	Edit SectionHea	der j		OK
.text .rdata	00001000	Name:	BFNIGHT	ОК	Save
.data .rsrc	00003000	VirtualAddress:	0004D000	Cancel	Sections
.reloc	0004C000	VirtualSize:	00001000		Directories
.NewSec	0004D000	I RawOffset:	00049C00		Directories
		BawSize:	00001000		FLC
		Flags:	E00000E0		TDSC
	FileAlignment:			1	Compare
	Mayic.			•	L

Then click ok and save. But now we need to insert 1000h bytes to the target with hex editor.

IAT (Import Address Table):

Every PE has a list of functions that aren't originally part of that PE. These functions are called Import which is located in the OS DLLs while PE doesn't know where they are located, So every win32 executable has an *Import Table Address* or *IAT* inside PE.

IAT contains all of information of imports , It means it is used as a lookup table when a PE is calling a windows32 API. Briefly the windows loader has to find each address of API which is called by the PE before starting.

Now take a look at inside our target(again *firefox* J)

004015A0	\$ E8 9B030000	CALL firefox(.00401940	
004015A5	.^ E9 26FDFFFF	JMP firefox(.004012D0	
004015AA	\$- FF25 <u>54204000</u>	UMP DWORD PTR DS:E<&MOZCRT19.operator new>3	MOZCRT19.operator new
004015B0	\$ 380D <u>08304000</u>	CMP_ECX, DWORD_PTR_DS: [403008]	
004015B6	.~ 75 02	JNZ SHORT firefox(.004015BA	
004015B8	. F3:	PREFIX REP:	Superfluous prefix
004015B9	. C3	RETN	
004015BA	→ E9 31040000	JMP firefox(.004019F0	
004015BF	CC	INTS	
004015C0	\$− FF25 <u>5C204000</u>	UMP DWORD PTR DS:[<&MOZCRT19.operator delet]	MOZCRT19.operator delete
004015C6	. 8B4424 04	MOV EAX, DWORD PTR SS:[ESP+4]	
004015CA	. 8B00	MOV EAX, DWORD PTR DS:[EAX]	
004015CC	. 8138_63736DE0	CMP DWORD PTR DS: [EAX], E06D7363	
004015D2	.∨ 75 2A	JNZ SHORT firefox(.004015FE	
004015D4	. 8378 10 03	CMP DWORD PTR DS:[EAX+10],3	
004015D8	.~ 75 24	<pre>JNZ SHORT firefox(.004015FE</pre>	
004015DA	. 8B40 14	MOV EAX, DWORD PTR DS:[EAX+14]	
004015DD	. 3D 20059319	CMP_EAX, 19930520	
004015E2	.v 74 15	JE SHORT firefox(.004015F9	
004015E4	. 3D 21059319	CMP_EAX, 19930521	
004015E9	.∨ 74 0E	JE SHORT firefox(.004015F9	
004015EB	. <u>3D</u> 22059319	CMP_EAX, 19930522	
004015F0	.v 74 07	JE SHORT firefox(.004015F9	
004015F2	. 3D 00409901	CMP EAX, 1994000	
004015F7	.~ 75 05	JNZ SHORT firefox(.004015FE	
004015F9	> E8_FE040000	CALL (JMP.&MOZCRT19.terminate)	
004015FE	> 3300	XOR_EAX, EAX	
00401600	. C2 0400	RETN 4	
00401603	. 68 <u>C6154000</u>	PUSH firefox(.004015C6	<pre>pTopLevelFilter = firefox(.004015C6</pre>
00401608	. FF15 <u>24204000</u>	CALL DWORD PTR DS: [<&KERNEL32.SetUnhandledE	SetUnhandledExceptionFilter
0040160E	. 3300	I XOR FAX-FAX	

CALL function which is shown at the figure is a good example of an API. It calls *SetUnhandledExceptionFilter* API. Right click on the function then click Assemble. You can see that it's like CALL DWORD PTR [XXXXXX].[XXXXXX] is the address of values which are located in memory

Go to the 402024 in olly hex dump (right click -> go to -> expression).

Address	Hee	e du	amp.						ASCII	
00402024	5D	49	84	70	42	98	80	7C	llä¦BÿǦ	
0040202C	46	24	80	7C	2E	98	80	7C	F\$Ç!.ÿÇ!	
00402034	00	<u>00</u>	<u>00</u>	00	60	ΕD	00	60		
0040203C	F2	56	01	60	20	<u>B2</u>	00	60	≥V0°µ."	
00402044	70	ED	ЙŇ	60	20	ZE.	90	60	pí. `0`	
0040204C	20	B8	ЙЙ	60	DМ	25	йñ	60	0.0%	
00402054	ЦØ.	De	95	60	HQ	DC.	04	60		
00402050	50	ųς	92	50	C0	63	96	20	[] 변경 주 주 주 주 [
00402064	40	16	90	20	00	15	90	20		
00402060	20	HC.	00	20	20	15	00	20	L L	
00402074	20	臣	00	20	60	15	00	20		
00402070	EG	15	00	20	20	64	00	20	YY 😚	
00402090	na.	ià	ãã.	60	er.	26	ãõ.	60		
00402094	20	55	ăй	ĕã.	žă	22	ăā.	ĕã		
00402090	10	2F	йй	ĞЙ	żά	10	ăй	ĔĞ		
00402004	6ŏ.	46	йй	ĞЙ	ĠЙ	2¥	йй	ĞЙ	iF ws I	
004020AC	ЗÓ	В9	ŏŏ	ĞŎ.	õõ	ōø	õõ	ŏŏ	ØI.	
004020B4	ĒĀ	Ø7	45	7Ē	00	ÖÖ.	ÖÖ.	00	Ω E″	
004020BC	CØ	DÈ	ØB	60	DØ	8Č	ØΒ	60	- °.⊖Œ	
004020C4	50	DF	ØB	60	30	6B	ØB	60	P ^e ð Okð	
004020CC	00	00	00	00	E0	11	ØF	60	α ∢ **	
004020D4	00	00	00	00	00	10	E0	60		
004020DC	00	11	E0	60	70	10	E0	60	dà phà	
004020E4	10	10	ĒØ	60	00	00	00	00	Ha'	
004020EC	1B	18	4B	60	38	B5	ŞĘ.	60	+1K'8	
004020F4	68	5A	65	60	31	C2	61	60	hZe'1+a'	
004020FC	57	E1	62	60	90	90	90	90	1 ^{βD}	
00402104	90	90	90	90	38	12	40	96	E‡@.	
0040210C	00	60	00	00	00	00	00	00		

Nightmare(BioHazard) – K053 Great IRAN

As the figure shows the address of the API is 7C84495D in my system. Notice that most of the API addresses start like 7XXXXXXX.

Now let see how windows loader can find addresses for every API. First windows loader reads the header of PE, so bytes at RVA 3C are read. The VA of the bytes will be 40003C(imagebase is 400000). Then if we plus 80h to the value of VA, we will find Virtual Address of Import Table.

But what's the Import Table?

The Import Table contains all of the information that is needed for windows to link the APIs for PE.

The structure of Import Table is simple . A header for each imported DLL plus additional null header that marks the end of Import Table. For example if you import APIs from Kernel32.dll and User32.dll , you will find 3 headers. 2 of them are for Kernel32.dll and User32.dll and a zero-header for marking the end.

The header of each dll is called *IMPORT_IMAGE_DIRECOTRY*. 'IMAGE' word refers that all of this stuff is done in memory.

But data structure which stores the information import symbols called IMAGE_IMPORT_DESCRIPTOR that has 5 properties. I'd like to use exact description of *Identifying Malicious Reverse Engineering Code Through* Book.

OriginalFirstThunk: - This member contains the RVA (pointer) of an array of IMAGE_TUNK_DATA structures. IMAGE_TUNK_DATA structures, is a union of dword size. This can be considered as a pointer to IMAGE_IMPORT_BY_NAME structure. The structure of IMAGE_IMPORT_BY_NAME structure is as follows: -IMAGE_IMPORT_BY_NAME STRUCT Hint Name1

Time/Date Stamp: - After the image is bound, this field is set to the time/data stamp of the DLL. This field is not mandatory; it can be zero.

Forwarder Chain: - The index of the first forwarder reference. This field is not mandatory; it can be zero.

Name: - This member contains the RVA (pointer) of an ASCII string that contains the name of the DLL.

FirstThunk:- As the name suggests the FirstThunk is very similar to that of OriginalFirstThunk . Similar to FirstThunk it also contains pointer (RVA) to array of IMAGE_THUNK_DATA structures. Although both the arrays contain same value, they are at different locations in the executable.

I think those are some boring description that make you bored, so go through see above information in our target. First we have to find Import Table, so open firefox in ollydbg then click on M (Memory Map Button) and double-click on PE Header.

00400000 00001000 fir	efox(PE header	Imag 01001002	R
00401000 00001000 fir	efox(.text	code	Imag 01001002	R
00402000 00001000 fir	efox(.rdata	imports	Imag 01001002	R
00403000 00001000 fir	efox(.data	data	Imag 01001002	R

Now you are able to see the header of firefox. Find RVA of 3C (ImageBase + 3C = 40003c). You can see the value is D8 now plus it with 80h that the RVA will be like: D8h + 80h + 400000h = 400158h. This RVA stores the VA of Import Table.

00400158	74230000	DD 00002374	Import Table address = 2374
0040015C	A0000000	DD 000000A0	Import Table size = A0 (160.)
00400160	00400000	DD 00004000	Resource Table address = 4000
00400164	38750400	DD 00047538	Resource Table size = 47538 (292152.)
00400168	00000000	DD 00000000	Exception Table address = 0
0040016C	00000000	DD 00000000	Exception Table size = 0
00400170	009C0400	DD 00049C00	Certificate File pointer = 49C00
00400174	F8150000	DD 000015F8	Certificate Table size = 15F8 (5624.)
00400178	00C00400	DD 0004C000	Relocation Table address = 4C000
0040017C	94010000	DD 00000194	Relocation Table size = 194 (404.)
00400180	20210000	DD 00002120	Debug Data address = 2120
110100100	10000000	DD 0000010	$D_{-} = D_{-} = -10$ (00.)

The RVA of Import Table Address will be 402374. So go to 402374 like:



Then enter 402374 in to the box and click ok then you need to Analyze Code(CTRL + A). If you consider you can find that we have 8 IMAGE_IMPORT_DESCRIPTORs and last one is a zero-header that marks the end then we have Import Lookup Table that stores the information of each dll that. Then we have the name of each function that is imported. These are Import's information that is pointed by IMAGE_IMPORT_DESCRIPTOR.

00402374 00402378 0040237C 00402380 00402384 00402388 00402388 00402388 00402380	00250200 00000000 00000000 74250000 EC200000 EC240000 0000000 0000000	DD 00002500 DD 0000000 DD 00000000 DD 00002574 DD 000026CC DD 000024EC DD 0000000 DD 0000000	Struct 'IMAGE_IMPORT_DESCRIPTOR' Struct 'IMAGE_IMPORT_DESCRIPTOR'
00402394 00402398 0040239C 004023A0 004023A0	D0250000 D8200000 D0240000 00000000 00000000	DD 000025D0 DD 000020D8 DD 000024D0 DD 000024D0 DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
004023A8 004023AC 004023B0 004023B4 004023B8	. 14260000 BC200000 E4240000 00000000	DD 00002614 DD 0000208C DD 000024E4 DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
004023BC 004023C0 004023C4 004023C8 004023CC	2E260000 D0200000 C8240000 00000000 00000000	DD 0000262E DD 000020D0 DD 000024C3 DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
004023D0 004023D4 004023D8 004023DC 004023DC 004023E0	 46260000 B4200000 4C240000 00000000 00000000 	DD 00002646 DD 00002084 DD 0000244C DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
004023E4 004023E8 004023EC 004023F0 004023F0 004023F4	. 76270000 . 38200000 . 14240000 . 00000000 . 00000000	DD 00002776 DD 0002038 DD 00002414 DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
004023F8 004023FC 00402400 00402404 00402404	 48290000 002000000 000000000 000000000 000000000 0000000000 	DD 00002948 DD 00002000 DD 00000000 DD 00000000 DD 00000000	Struct 'IMAGE_IMPORT_DESCRIPTOR'
0040240C 00402410	. 00000000 . 00000000	DD 00000000 DD 00000000	
	00402411 00402410 00402420 00402420 00402424 00402428 00402428 00402430 00402430 00402430 00402430 00402434 00402431 00402443 00402444 00402444	1 1 <th1< th=""> <th1< th=""> <th1< th=""> <th1< th=""></th1<></th1<></th1<></th1<>	IMport lookup vadie for Kennelsz.ott
	004024440 00402454 00402458 00402458 00402458 00402458 00402458 00402458 00402458 00402458 00402468 00402468 00402470 00402470 00402470 00402470 00402470 00402470 00402470 00402470 00402470 00402470 00402480 0000000000	B00000000 D00000000 102380000 D00002210 022390000 D00002210 022390000 D000022702 D4270000 D00002772 B8270000 D00002772 B8270000 D00002780 B8270000 D00002780 S2260000 D00002780 S2260000 D00002780 93260000 D000002780 93260000 D000002652 93260000 D000002684 93260000 D000002684 B4266000 D000002684 B4266000 D000002684 B22260000 D000002684 B2260000 D000002684 B2260000 D000002684 B2260000 D000002684 B2260000 D00000274 B2260000 D000002744 S2270000 D000002744 S2270000 D000002744 S2270000 D000002744 S2270000 D000002744 S2270000 D0000002744 S22700000 D0000002744	Import lookup table for 'MOZCRT19.dll' Import lookup table for 'USER32.dll'
	004024CC 004024D0 004024D4	E8250000 DD 000025E8	Import lookup table for 'nspr4.dll'
	004024D8 004024DC	08260000 DD 00002608 DD 000025DA DD 000025DA	
	004024E0 004024E4 004024E8	. 12260000 DD 0000261E . 00000000 DD 0000261E	Import lookup table for 'plo4.dll'
	004024EC 004024F0 004024F4	C2250000 C2250000 C2250000 C0 C2250000 C0 C2250000 C0 C2250000 C0 C2250000 C0 C2250000 C0 C2250000 C22500000 C2250000	Import lookup table for 'xpcom.dll'
	004024F4 004024F8 004024FC	72550000 DD 0000257C 00000000 DD 0000257C	
	00402500 00402504 00402508	68250000 DD 00002568 54250000 DD 00002554 42250000 DD 00002554	Import lookup table for 'xul.dll'
	0040250C 00402510	. 2C250000 DD 0000252C . 18250000 DD 00002518	
	00402514 00402518	. 6703 DD 00000000	

004025IH	•	20 22	45	OF .	45	14	Hooli WWCTonearehbboar
0040252H	•	61 00					ASCII "a",0
0040252C	•	6803					DW 036H
3040252E	•	58 52	45	5F	47	61	ASCII "XRE_GetFileFromP"
0040253E	•	61 74	68	00			ASCII "ath",0
30402542		6803					DW 0368
30402544	•	58 52	45	5F	46	71	ASCII "XRE_FreeAppData",0
20402554		6903				1	DW 0369
20402556		58 52	45	SE .	47	6!	ASCII "XRE GetBinaryPat"
20402566		ÃÃ ÃÃ		<u> </u>		~1	ASCII "b".0
20402568		7103					DM 0371
20402560	•	59 52	45	SE .	6D	6	OSCIL "YRE main" 0
20402500	•	00 52	40	OF .	00	°I	DD 00
20402573	•	70 75	60	25	61	21	DE DE UNIT ALLE A
20402574	•	10 10	ъυ	2E	64	익	HSUIT "Xul.dit",0
2040257C	•	2400		40	6 F .	21	
2040257E	•	4E 53	55	4U	ън	ы	HSUII "MS_LOGIERM",0
00402589	•	99					DR 00
2040258H	•	0400				_	DW 0004
0040258C	•	4E 53	55	43	53.	-91	HSUII "NS_UStringContai"
0040259C	•	6E 65	72	49	6E	61	ASCII "nerInit2",0
004025H5	•	00					DB 00
204025A6	•	0200					DW 0002
204025A8	•	4E 53	5F	43	53	7	ASCII "NS_CStringContai"
204025B8		6E 65	72	46	69	61	ASCII "nerFinish",0
304025C2	•	2200					DW 0022
304025C4		4E 53	5F	4C -	6F -	61	ASCII "NS_LogInit",0
304025CF		00					DB 00
204025D0		78 70	63	6F	6D	21	ASCII "xpcom.dll".0
204025DA		7801				-1	DW 0179
204025DC		50 52	SE.	73	6D	71	ASCIL "PR smprintf".0
20402552		9300	0	.0	00	11	DM 0093
30402550	•	5366	FF	47	6E	-1	DW 0075 OCCII MDD CatEstyM 0
204023EH	•	2001	ЭF I	40	00	-1	DW 047D
204025F4 204025F4	•	7801		70	20	-,	DW 0175
204025F6	•	50 52	55	73	ьυ	9	HSUII "FR_SMprintf_free"
20402606	•	00					H5011 0
00402607	•	00					DR 00
30402608	•	3801					DW 0138
3040260A	•	50 52	5F	53	65	7	ASCII "PR_SetEnv",0
30402614	•	6E 73	70	72	34	21	ASCII "nspr4.dll",0
3040261E	•	0800					DW 0008
30402620		50 4C	5F	73	74	71	ASCII "PL_streaseemp",0
3040262E		70 6C	63	34	2E	61	ASCII "plc4.dll".0
30402637		00					DB 00
30402638		F801					DW 01F8
2040263A		4D 65	73	73	61	61	ASCII "MessageBoxA".0
30402646		55 53	45	52	33	3:	ASCII "USER32.dll".0
20402651	-	ññ -				-1	DB 00
20402652		й <u></u> Ерр					DN RARE
20402654		SE SE	32	40	59	4	ASCII "2220VAPAXI0Z".0
20402661	•		02	40	07	-	DR 00
30402662	•	0404					DN 0404
30402664	•	55 76	72	6E	70	zł	OSCIL " usperintf" 0
30402004	•	00	10	OE	10	1	DD 00
2040200F	•	1100					
20402670	•	1100		40	<u>го</u> –	al	DW 0011 OCCLI MODOQUOVDOVGZM Q
20402072	•	or or	22	40	57	4	DD GG
2040267F	•	00					
20402680	•	22 40	70	20	10	11	
20402682		66 63	73	6U	65	ы	HSCII "Westen",0
00402689	•	00					DB 00
2040268H	•	1601				_	DW 0116
2040268C	•	5F 61	6D	73	67	51	HSUII "_amsg_exit",0
00402697	•	00					DB 00
00402698	•	F800					DW RAFS
0040269A	•	SF SF	77	67	65	74	HSCII "wgetmainargs",0
004026A9		00					DB 00
004026AA		2001					DW 012C
304026AC		5F 63	65	78	69	7.	ASCII "_cexit",0
204026B3		00					DB 00
304026B4		7001					DW 017C
304026B6		5F 65	78	69	74	Ø	ASCII "_exit".0
204026BC		6300					DW 0063
204026BE		5F 58	63	79	74	4	ASCIL " XoptFilter".0
204026C0		C704				1	DN 9407
20402600		65 78	69	74	ØØ.		ASCIL "exit".0
20402600	•	00 10	05	14	00		DB 00
30402601		5000					DN 0050
30402602	•	FF FF	77	60	CE.	0	Decil V vicitary 0
20402604	•	5F 5F	11	69	OE .	6	HSCIIWINITERV",0
204026DF	•	00					
004026E0	•	FF01			-		DW BIFF
004026E2	•	5F 69	6E	69	74	\mathbf{C}	HSUII "_initterm",0
004026EC	•	0002			_		DW 0200
004026EE		SF 69	6E	69	74	7.	ASCII "_initterm_e",0
304026FA		3001					DW 013C
304026FC		5F 63	6F	6E	66	61	ASCII "_configthreadloc"
3040270C		61 6C	65	00			ASCII "ale",0
30402710		E400					DW 00E4
30402712		SF SF	73	65	74	71	ASCII "setusermatherr"
20402722		00	-				05011 0

Inject A DLL:

I think now we have enough information to inject a dll to our target, but above all we need a framework, a paper & pencil.

Framework:

- 1. First we need to add a section then make jump to the main EP(Entry Point) and change EP to our section's EP.
- 2. Then all of firefox's IMAGE_IMPORT_DESCRIPTOR should be moved to the new section and Import Table Address's value should be changed to the New Section's Import Table Address
- 3. Next we have to add Import Lookup and a IMAGE_IMPORT_DESCRIPTOR for dll which you want to inject
- 4. Then call it.
- 5. ENJOY!

1st step

You need to Add a section by lordpe with 3000h size like I said, then you need to find the RVA of EP. easy way is value of .BFNIGHT's Voffset. So my EP would be "4D000 + 15F1 = 4E5F1" (consider EP should be a place that has lots of *Code Cave*). Before change the EP you need to write the main of EP in your paper because we want to make a jump to the main EP.

So main EP is 4015A0 and new EP is 4E5F1. So lets change it.

[PE Editor] - d:\	program fi	les\mozilla firefox\firef	ox(analyze).exe	
Basic PE Header In	formation			OK
EntryPoint: ImageBase:	0004E5F1 00400000	Subsystem: Number0fSections:	0002	Save
SizeOfImage:	00050000	TimeDateStamp:	49F1091D	Sections
BaseOfCode:	00001000	SizeOfHeaders:	00000400 ? +	Directories
BaseOfData:	00002000	Characteristics:	0102	FLC
SectionAlignment:	00001000	Checksum:	0004DF28 ?	TDSC
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0	Compare
Magic:	010B	NumOfRvaAndSizes:	00000010 + -	L

Tip : Code Cave is Place of a PE which has null data

29 A9 EE 32 16 ASCII ")⊤€2" 0044E5EC . nR. 0044E5F1 AA2FFBFF JMP firefox(.004015A0 Backup ۲ ۲ Copy 440 0044E5FA 0044E5FB 0044E5FC 0044E5FD Binary ۲ Undo selection Alt+BkSp Assemble Space 0044E5FE 0044E5FF 0044E600 0044E601 Label : Comment ÷ 0044E602 0044E603 0044E604 Breakpoint ۲ 0044E605 Hit trace ۲ 0044E606 0044E606 0044E607 0044E608 ۲ Run trace 0044E608 0044E609 0044E60A 0044E60B Follow Enter 00444E608 0044E600 0044E608 0044E608 0044E610 0044E611 0044E611 0044E613 0044E613 0044E614 0044E615 0044E616 0044E618 0044E618 0044E618 Go to ۲ Follow in Dump ۲ Search for ۲ Find references to ۲ View ۶ Copy to executable ۲ Selection Analysis ۲ Help on symbolic name Ctrl+F1 < 1 Detach Process 00404500 00401500 Address Hex dump ASCII
 Hourses
 Hourses

 004403000
 FF

 00403000
 FF

 00403010
 FE

 00403011
 FE

 00403010
 FE

 00403010
 FE

 00403020
 60

 00403023
 60

 00403025
 60

 00403025
 60

 00403025
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 00403045
 60

 004 00403000 00403008 00403010 00403018 00403020 00403020 00403028 00403028 Process Patcher N⊬@¶∭,∔nD ■ 0... Analyze This! Bookmark ۶ Code Ripper Run Script ۲ Dump debugged process Overflow Return Address ۲ Make dump of process Ultra String Reference ۲ Appearance

Now open target in ollydbg and assemble "JMP 4015A0" then compile it.

If you run target , It will work correctly.

2nd Step

Find IMAGE_IMPORT_DESCRIPTOR like I said then highlight them and Right click and Follow Value in Dump. Then Right click on the highlighted values and Binary then Binary Copy



Now paste them in to the our Null space in our new section like: (Then you have to compile the target again J)

Address	Hex	dum	p				F	ASC/*	T T				1		
0044E633	00 0	00	0 00	00	00	00	30.	•••		Backup		۰.			
0044E638 0044E643	00 0	0 0 0 0	0 06 A AA	1 00	00	00	40 . AG .	•••		Сору		•			
0044E64B	ŏŏ ŏ	ŏŏ	õ õõ	ίŏŏ	ŏŏ	ŏŏ i	ăŏ .					N.	r de	CHAR	1
0044E653	00 0	00	000	00	00	00	30 ·	•••		binary		"	EalC	Cm+E	
0044E663	00 0	00	0 00	1 00	00	00 1	30 . 30 .			Label	:		Fill with 00's		
0044E66B	00 0	0 0	0 00	00	00	<u>00</u>	ā0 .			Breakpoint			Fill with FE's		
0044E673 0044E678	00 0	0 0 0 0	0 00 0 00	100	00	00 0	40 . 30 .	•••							
0044E683	õõ õ	ŏŏ	õ õ	ίŏŏ	ŏŏ	i õõ	ăŏ .			Search for		1	Ripery copy		
0044E68B	00 0	00	000	00	00	00	30.	• • •		Find references	Ctrl+R		Dinary copy		
0044E693 0044E69B	00 0 00 0	и И И	0 06 A AA) 00) 00	- 00 - 00	00 1	аю . Ай .			View everytable file			Binary paste		
0044E6A3	ÕÕ Õ	õõ	õ õe	00	õõ	00	90 .			view executable file					۰.
0044E6AB	00 0	0 0 0 0	0 00 0 00	00	00	00 0	30 . 30 .	•••		Copy to executable file					
0044E6BB	00 0	0 0	0 00	00	00	00 1	30 .			Go to		¥.			
0044E6C3	00 0	00	000	00	00	00	30.	• • •							
0044E6CB 0044E6D3	00 0 00 0	и И И	0 06 A AA) 00 1 00	- 00 - 00	- 00 - 00	аю . Ай .			Hex		۲			
0044E6DB	õõ õ	ŏŏ	õ õe) ÖÖ	ŏŏ	ÖÖ i	āğ .		•	Taut					
0044E6E3	00 0	0 0 0 0	0 00 0 00	00	00	00	30 . 30 .	•••		Text					
0044E6F3	00 0	0 0	0 00	00	00	00 1	30 .			Short		۰.			
0044E6FB	00 0	00	0 00	00	00	00	30.	• • •		Long		×			
0044E703 0044E708	00 0	0 0 0 0	0 06 0 00	1 00	00	00	40 . AG	•••							
0044E713	ŏŏ ŏ	ŏŏ	ŏ ŏĕ	ίŏŏ	ŏŏ	ÖÖ i	ăŏ .			Float		•			
0044E71B	00 0	00	000	00	00	00	30.	• • •		Disassemble					
0044E723 0044E72B	00 0	0 0 0 0	0 00 0 00	00	00	00	90 . 30 .			Special					
0044E733	00 Ö	õ õ	õ õe	00	őð	00	90 .			opecial		-			
0044E73B	00 0	0 0 0 0	0 00 0 00	00	00	00	20 . 20 .	•••		Appearapce					
0044E74B	00 0	õ õ	0 00	00	00	00	ãõ .			white a line					

So we need to change Import Table Address . The first byte of IMAGE_IMPORT_DESCRIPTOR is Import Table Address and we have to change it with lordpe.

Import Table Address(new) = 44E61B (in my system)

RVA = 44E61B - 400000 = 4E61B

0044E613 0044E623 0044E628 0044E633 0044E633 0044E633 0044E633 0044E638 0044E638 0044E653 0044E658 0044E658	00 00 EC 00 D0 00 BC 00 2E	00 20 20 20 20 25 20 25 20 20 20 20 20 20 20 20 20 20 20 20 20	00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00	00 74 EC 00 D8 00 14 E4 00 D0	00 25 20 20 20 20 20 20 20 20 20 20 20	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00	х ∞ ±х ±\$ ±	t% ∞\$ ÷ ¶& Σ\$ ⊥	
0044E663	25	26	00	00	DØ	20	00	00	.8	щ.	•••

Open the target in LordPE and click on Directories Button on the right and change ImportTable to the

irectory Table]	1				
Directory Informat	ion RVA	Size	ОК		
ExportTable:	00000000	00000000 L	H Save		
ImportTable:	0004E61B	000000A0 L	H	ITE Word	
Resource:	00004000	00047538 L	Н	15	
Exception:	00000000	00000000 L	H		
Security:	00049C00	000015F8	H	¶ AaBb	CcDr AaBbC
Relocation:	0004C000	00000194 L	Н	TNO	rmal II No Spi
Debug:	00002120	0000001C L	н	1140	rinar 140 spa
Copyright:	00000000	00000000 L	H	6	
Globalptr:	00000000	00000000			
TIsTable:	00000000	00000000 L	Н		
LoadConfig:	00002250	00000040 L	H		
BoundImport:	00000000	00000000 L	Н		PE Editor
IAT:	00002000	00000104	H	120	
DelayImport:	00000000	00000000 L	Н	100	Break & Enter
COM:	00000000	00000000 L	H	100	Rebuild PE
Reserved:	00000000	0000000	Н	00	Unsplit
uail authiaí	51 0111 11100 11110	LING IN DIGHT HILD	lon(anai)co/rono		Dumper Server
PE Header Informa	ation				Options
nint: 00	04E5F1 %	ibsustem:	0002		
lase: 00	400000 Ni	umberOfSections:	0006	Save	About
mage: 00	050000 Ti	meDateStamp:	49F1091D	Sections	Exit
fCode: 00	001000 Si	zeOfHeaders:	00000400 ? +	Directories	

new RVA

3rd and 4th steps:

In this case I'd like to use Ashraf Cracker's DLL (Password.dll).

http://www.4shared.com/file/112647070/69e44e47/Password.html

So copy this dll in firefox's directory. Password.dll has a function that is called PasswordMain.

Now we need to make Import Lookup Table for the dll. Choose a Null Data Place like 44E6CD and find it in Value Dump view.

First you need to insert dll name (Password.dll)



ASCII	Pas	SWOI	rd.	d11								
UNICODE												
HEX +OC	50 00	61 00	73 00	73 00	77	6 F	72	64	2 E	64	6 C	6 C

Then you need to insert Function name

0044E6CD 50 61 73 73 77 6F 72 64 Password 0044E6D5 2E 64 6C 6C 00 00 00 00 .dll 0044E6D5 50 61 73 73 77 6F 72 64 Password 0044E6E5 40 61 69 6E 00 00 00 00 Main 0044E6F5 00 00 00 00 00 00 00 00 00 0044E6F5 00 00 00 00 00 00 00 00 00	Edit data at 0044E6DD
0044E745 00 <	ASCII PasswordMain UNICODE HEX +00 50 61 73 73 77 6F 72 64 4D 61 69 6E 00 00 00 00
0044E765 00	✓ Keep size OK Cancel

Next you need to insert the RVA that points to the function name and 20 10 00 10

RVA = 44E6DB - 400000 = 4E6DB à DB E6 04 00

0044E6CD	50	61	73	73	77	6F	72	64	Password	
0044E6D5	2E	64	60	6C	00	00	00	00	.dll	
0044E6DD	50	61	73	73	77	6F	72	64	Password	
0044E6E5	40	61	69	6E	00	00	00	00	Main	
0044E6ED	DB	E6	04	00	ØØ.	00	ØØ.	00		
0044F6F5	20	10	80	10	ññ.	ññ.	ññ.	ññ		
0044E6ED	00	00	ññ	ÑØ.	ññ	ññ	ññ	ññ		
0044E705	ññ.									
0044F70D	ññ	ññ.	ññ	ññ	ññ	ññ.	ññ	ññ		
0044F715	ñй	ññ	ññ.	ññ	ññ	йň	йň	ñň		
0044F71D	ññ	ññ	ññ	ññ	ññ	ññ	ñй	ññ		
0044F725	ññ									
0044F72D	ăй	ññ	ăă	ññ	ññ	ññ	ññ	ññ		
0044F735	ññ.	ññ	ăă	ññ	ññ	ăă	ăă	ññ		
0044E73D	ññ									
0044E745	ăă	ññ	ãã	ññ.	ăă	ăă	ññ	ññ		
0044E74D	aa	ññ	aa	ññ.	ãã	ññ	aa	aa		
00445755	aa	aa	ãã	aa	aa	ãã	aa	aa		
0044E7ED	aa	00	00	00	ãã	66	00	00		
00446766	00	00	00	00	00	00	00	00		
00445760	00	00	00	00	00	00	00	00		
00446700	00	00	00	00	00	00	00	00		
00445770	00	00	00	00	00	00	00	00		
00445705	00	00	00	00	00	00	00	00		
00445705	00	00	00	00	00	00	00	00		
0044E70E	00	00	00	00	00	00	00	00		
0044E795	00	00	00	00	00	00	00	00		
NN44F7911	nn	nn	ии	nn	nn	ии	ии	NN1		

ASCII	μ• ▶.
UNICODE	*
HEX +00	DB E6 04 00 00 00 00 00 20 10 00 10 00 00 00 00

So the final step in make a IMAGE_IMPORT_DESCRIPTOR witch this structure

OrginalFirstThunk = 44E6ED – 400000 = 4E6ED à ED E6 04 00
Time/Date Stamp and ForwarderChain = Zero
Name: 4E6CD à CD E6 04 00
FirstThunk: 4E6F5 à F5 E6 04 00

Find the values of the last IMAGE_IMPORT_DESCRIPTOR in the olly's Value Dump and insert those data which is calculated like:

0044E6A7 ED E6 04 00 00 00 00 00 00 00 00 0044E6AF 00 00 00 00 CD E6 04 00 0044E6B7 F5 E6 04 00 00 00 00 00 00 Jp+	Edit data at 0044E6A7
0044E6C7 00 00 00 00 00 00 00 00 00 00 0044E6CF 73 73 77 6F 72 64 2E 64 ssword.d 0044E6DF 6C 6C 00 00 00 00 00 50 61 11Pa 0044E6DF 73 73 73 77 6F 72 64 4D 61 sswordMa 0044E6DF 73 73 73 77 6F 72 64 4D 61 sswordMa	ASCII
0044E6FF 04 00 00 00 00 00 00 00 00 00 0044E6FF 04 00 00 00 00 00 00 00 00 0044E6FF 00 00 00 00 00 00 00 00 0044E6FF 00 00 00 00 00 00 00 0044E6F7 00 00 00 00 00 00 00 00 0044E6F7 00 00 00 00 00 00 00 00	HEX +00 ED E6 04 00 00 00 00 00 00 00 00 00 00 00 00
0044E717 00 00 00 00 00 00 00 00 00 00 00 0044E717 00 00 00 00 00 00 00 00 00 0044E717 00 00 00 00 00 00 00 00 0044E727 00 00 00 00 00 00 00 00 0044E72F 00 00 00 00 00 00 00 00	
0044E737 00 00 00 00 00 00 00 00 00 00 0044E737 00 00 00 00 00 00 00 00 0044E747 00 00 00 00 00 00 00 00 0044E747 00 00 00 00 00 00 00 00 0044E757 00 00 00 00 00 00 00 00 0044E757 00 00 00 00 00 00 00 00	✓ Keep size OK Cancel

Then save the target. Now if you assemble CALL DWORD PTR [44E6F5], you can find that the dll has been injected correctly and it works perfectly.

0044E5F1 0044E5F7 0044E5FC 0044E5FD	FF15 F5E64400 - E9 A42FFBFF 00 00	CALL DWORD PTR DS:C&&Password.PasswordMain> JMP firefox(.004015A0 DB 00 DB 00	Password.PasswordMain
0044E5FE 0044E5FF 0044E600 0044E601 0044E602 0044E602 0044E603 0044E603	00 00 00 00 00 00	heck Password]	3
0044E605 0044E606 0044E607 0044E608 0044E608 0044E608 0044E608 0044E608 0044E608 0044E608 0044E608 0044E608 0044E610 0044E611 0044E611	00 00 00 00 00 00 00 00 00 00 00 00 00	Check Exit	

The password is : 0128793089

Finally we have injected a dll in our target and now we know how to do it.

Our website: http://b0frenzy.freehostia.com

Special Thanks to Lena151 because of his perfect tutorials and website

GREAT IRAN

GREAT BLACK-OUT FRENZY

BE SAFE