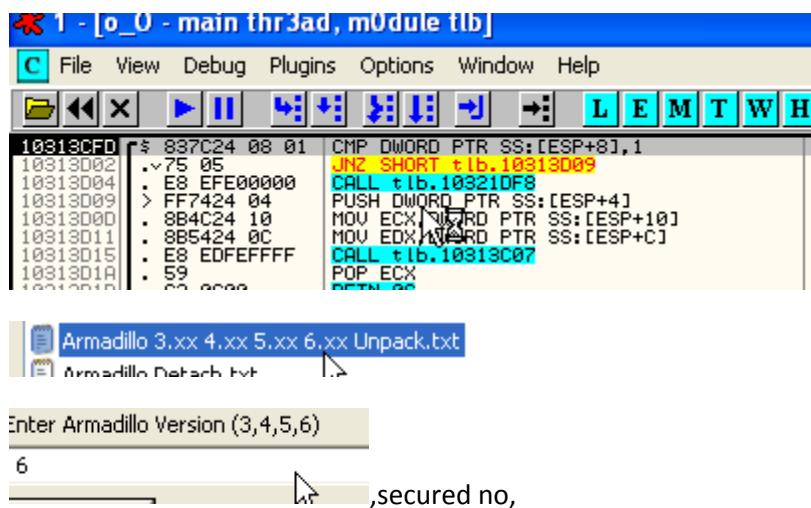


True Launch Bar



Comienzo investigando la dll esta empacada con armadillo 7.4:

```
<- 28-10-2010 13:31:51 - [2.0] ->
C:\archivos de programa\TrueLaunchBar\
Protected Armadillo
<-Find Protect
Protection system (Professional)
<Protection Options>
System File
Import Table Elimination
Strategic Code Splicing
<Backup Key Options>
Main Key Only, No Backup Keys
<Compression Options>
Minimal/Fastest Compression
<Other Options>
Disable Monitoring Thread
<-Find Version
Version 7.40 27-07-2010
<- Elapsed Time 00h 00m 03s 312ms ->
```



10168554	8BFF	MOV EDI,EDI	OEP
10168556	55	PUSH EBP	
10168557	8BEC	MOV EBP,ESP	
10168559	837D 0C 01	CMP DWORD PTR SS:[EBP+C],1	
1016855D	75 05	JNZ SHORT t1b.10168564	
1016855F	E8 1B0F0100	CALL t1b.1017947F	
10168564	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
10168567	8B4D 10	MOV ECX,DWORD PTR SS:[EBP+10]	
1016856A	8B55 0C	MOV EDX,DWORD PTR SS:[EBP+C]	
1016856D	E8 ECFFFFFF	CALL t1b.1016845E	
10168572	59	POP ECX	
10168573	5D	POP EBP	
10168574	C2 0C00	RETN 0C	
10168577	CC	INT3	
10168578	CC	INT3	
10168579	CC	INT3	
1016857A	CC	INT3	
1016857B	CC	INT3	
1016857C	CC	INT3	
1016857D	CC	INT3	
1016857E	CC	INT3	
1016857F	CC	INT3	
10168580	833D 3CA31E10 0	CMP DWORD PTR DS:[101EA33C],0	
10168587	75 05	JNZ SHORT t1b.10179820	
1016858D	83EC 08	SUB ESP,8	
10168590	0FAE5C24 04	STMXCSR DWORD PTR SS:[ESP+4]	
10168595	8B4424 04	MOV EAX,DWORD PTR SS:[ESP+4]	
10168599	25 801F0000	AND EAX,1F80	
1016859E	3D 801F0000	CMP EAX,1F80	
101685A3	75 0F	JNZ SHORT t1b.10179820	
101685A5	D93C24	FSTCW WORD PTR SS:[ESP+4]	
101685A8	66:8B0424	MOV AX,WORD PTR SS:[ESP+4]	
101685AC	66:83E0 7F	AND AX,7F	
101685B0	66:83F8 7F	CMP AX,7F	
101685B4	8D6424 08	LEA ESP,DWORD PTR SS:[ESP+8]	

MSG ODbgScript

Script appears to have reached OEP, check

Ahora

Processes: (F5 to refresh)

Modules:

Armlnline.exe (BBC)

loaddll.exe (84)

OfficeLiveSignIn.exe (ABC)

winword.exe (8D0)

notepad.exe (BEC)

notepad.exe (424)

PrintScreen.exe (6BC)

notepad.exe (7D0)

notepad.exe (9E4)

hpqSTE08.exe (8F0)

hnta08.exe (3A4)

loaddll.exe

ntdll.dll

kernel32.dll

user32.dll

GDI32.dll

imm32.dll

ADVAPI32.dll

RPCRT4.dll

Secur32.dll

t1b.dll

kernel32.dll

Analysing module...

Searching for Code Splices...

00A3C000	00000000		
00A50000	00006000		
00A60000	00001000		
00A70000	00001000		
00A80000	0013D000		
00BC0000	00006000		
00D20000	00020000		
10000000	00001000	t1b	PE header
10001000	001A4000	.oewko	
101A5000	0003C000	.b1fjy	data,expor
101F1000	00000000	t1b	

Code Splicing

Start Of Spliced Code: 0x D20000

Length Of Spliced Code: 0x 10000

Undo

Remove Splices

cambio a

Code Splicing

Start Of Spliced Code: 0x D20000

Length Of Spliced Code: 0x 20000

Undo

Remove Splices

Decia tamaño 5c4, revisando es un poco mayor, coloco 7c4

Import Elimination

Base Of Existing IAT: 0x971298

Length Of Existing IAT: 0x7C4

New Base VA Of IAT: 0x10296000

Rebase IAT

```
----- Code Splicing -----  
Process memory buffered successfully.  
1832 splices repaired.  
Splice repairing complete. Patching process...  
Patch succesful.  
  
----- Rebasing IAT -----  
Process memory buffered successfully.  
5197 DLL calls found total.  
Analysing...  
358 API functions referenced from 11 DLLs.  
Redirecting DLL references:  
5197 calls redirected total.  
Patching process...  
Process successfully patched.
```

Ahora el import rec

SELECT A MODULE

Image Base	Image Size	Name
008B0000	00009000	normaliz.dll
10000000	00763000	tlb.dll
3F600000	000F6000	wininet.dll

IAT Infos needed

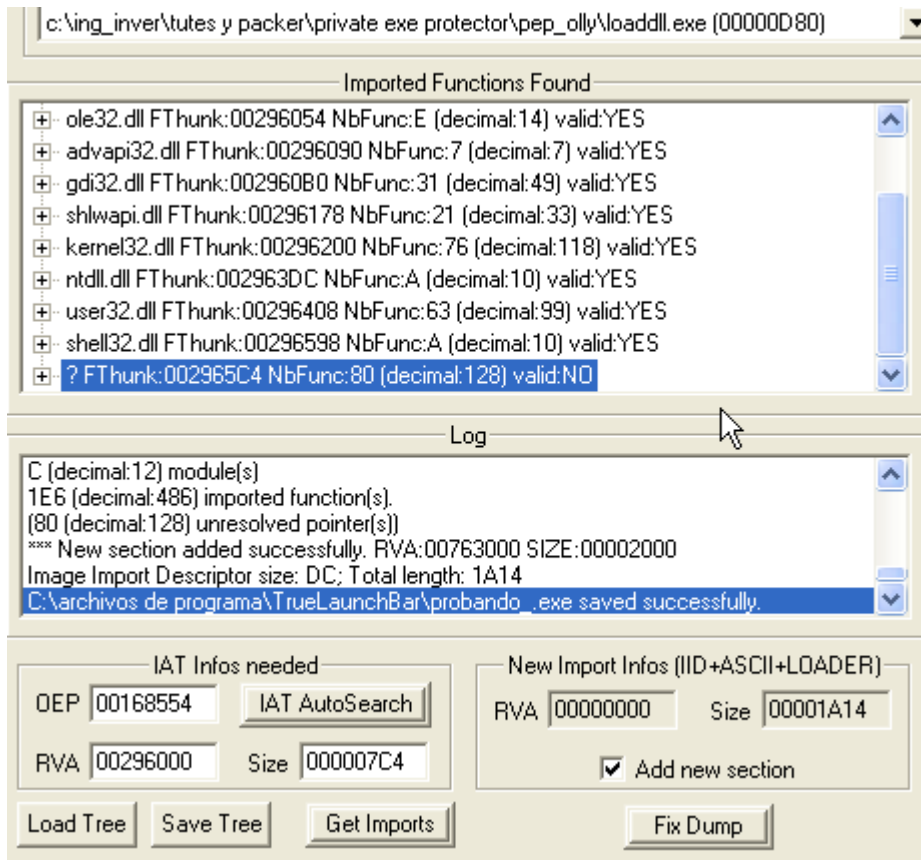
OEP: 00168554

RVA: 00296000

Size: 000005C4

IAT AutoSearch

Hago el dump desde import ret y agrego todo incluyendo los invalidos



Y renombro el tbl.dll a tbl_original

Y este unpacked como tbl.dll

pruebo y ejecuta /unpacked:

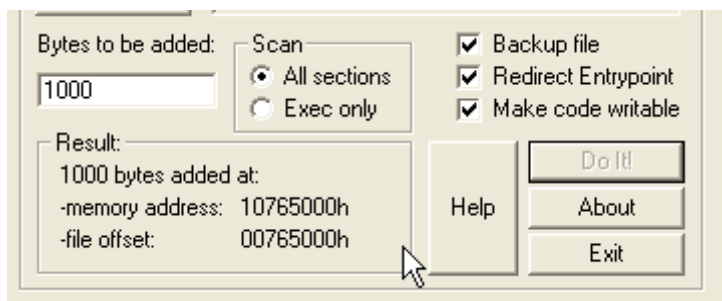


[Directory Table]				
Directory Information				
	RVA	Size		
ExportTable:	001E0100	000000B6	...	L H
ImportTable:	00763000	000000DC	...	L H
Resource:	006B9000	000A96D0	...	L H
Exception:	00000000	00000000		L H
Security:	004CB000	00001810		H
Relocation:	00399000	0000852C	...	L H
Debug:	001A5620	0000001C	...	L H
Copyright:	00000000	00000000	...	L H
Globalptr:	00000000	00000000		
TlsTable:	00000000	00000000	...	L H
LoadConfig:	00000000	00000000		L H
BoundImport:	00000000	00000000	...	L H
IAT:	00000000	00000000		H
DelayImport:	001DDBEC	00000080		L H
COM:	00000000	00000000	...	L H
Reserved:	00000000	00000000		H

Lo especial es que ciertos punteros que parecieran ser malos tienen direcciones apuntadas a

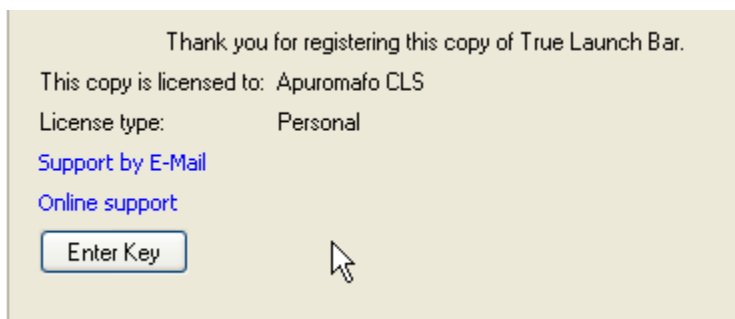
100F8C2F	68 F48F1E10	PUSH tlb.101E8FF4	ASCII "GetModuleFileNameW"
100F8C34	68 184F1B10	PUSH tlb.101B4F18	ASCII "kernel32.dll"
100F8C39	68 5CF01A10	PUSH tlb.101AF05C	
100F8C3E	E8 92140000	CALL tlb.100FA005	
100F8C43	^FF25 F48F1E10	JMP DWORD PTR DS:[101E8FF4]	tlb.100F8C24
100F8C49	^FF25 F48F1E10	JMP DWORD PTR DS:[101E8FF4]	tlb.100F8C24
100F8C4F	68 33A80F10	PUSH tlb.100FA833	
100F8C54	FF35 14A31E10	PUSH DWORD PTR DS:[101EA314]	
100F8C5A	68 60901E10	PUSH tlb.101E9060	
100F8C5F	68 2C4F1B10	PUSH tlb.101B4F2C	ASCII "lstrcpw"
100F8C64	68 5CF01A10	PUSH tlb.101AF05C	ASCII "kernel32.dll"
100F8C69	E8 67140000	CALL tlb.100FA005	
100F8C6E	^FF25 60901E10	JMP DWORD PTR DS:[101E9060]	tlb.100F8C4F
100F8C74	^FF25 60901E10	JMP DWORD PTR DS:[101E9060]	tlb.100F8C4F
100F8C7A	68 40A80F10	PUSH tlb.100FA840	
100F8C7F	FF35 10A31E10	PUSH DWORD PTR DS:[101EA310]	
100F8C85	68 64901E10	PUSH tlb.101E9064	
100F8C8A	68 384F1B10	PUSH tlb.101B4F38	ASCII "lstrcpyW"
100F8C8F	68 5CF01A10	PUSH tlb.101AF05C	ASCII "kernel32.dll"
100F8C94	E8 3C140000	CALL tlb.100FA005	
100F8C99	^FF25 64901E10	JMP DWORD PTR DS:[101E9064]	tlb.100F8C7A
100F8C9F	^FF25 64901E10	JMP DWORD PTR DS:[101E9064]	tlb.100F8C7A
100F8CA5	68 1BB80F10	PUSH tlb.100FB81B	
100F8CAA	FF35 0CA31E10	PUSH DWORD PTR DS:[101EA30C]	
100F8CB0	68 588F1E10	PUSH tlb.101E8F58	
100F8CB5	68 444F1B10	PUSH tlb.101B4F44	ASCII "RegQueryValueExW"
100F8CBA	68 904E1B10	PUSH tlb.101B4E90	ASCII "advapi32.dll"
100F8CBF	E8 11140000	CALL tlb.100FA005	
100F8CC4	^FF25 588F1E10	JMP DWORD PTR DS:[101E8F58]	tlb.100F8CA5
100F8CCA	^FF25 588F1E10	JMP DWORD PTR DS:[101E8F58]	tlb.100F8CA5
100F8CD0	68 15B80F10	PUSH tlb.100FB815	
100F8CD5	FF35 08A31E10	PUSH DWORD PTR DS:[101EA308]	
100F8CDB	68 5C8F1E10	PUSH tlb.101E8F5C	
100F8CE0	68 584F1B10	PUSH tlb.101B4F58	ASCII "RegQueryValueW"
100F8CE5	68 904E1B10	PUSH tlb.101B4E90	ASCII "advapi32.dll"
100F8CEA	E8 E6130000	CALL tlb.100FA005	
100F8CEF	^FF25 5C8F1E10	JMP DWORD PTR DS:[101E8F5C]	tlb.100F8CD0
100F8CF5	^FF25 5C8F1E10	JMP DWORD PTR DS:[101E8F5C]	tlb.100F8CD0
100F8CFB	68 4DB80F10	PUSH tlb.100FB84D	
100F8D00	FF35 04A31E10	PUSH DWORD PTR DS:[101EA304]	
100F8D06	68 44911E10	PUSH tlb.101E9144	
100F8D0B	68 744F1B10	PUSH tlb.101B4F74	ASCII "wprintfW"
100F8D10	68 684F1B10	PUSH tlb.101B4F68	ASCII "user32.dll"
100F8D15	E8 8B130000	CALL tlb.100FA005	
100F8D1A	^FF25 44911E10	JMP DWORD PTR DS:[101E9144]	tlb.100F8CFB
100F8D20	^FF25 44911E10	JMP DWORD PTR DS:[101E9144]	tlb.100F8CFB
100F8D26	68 F6070F10	PUSH tlb.100F07F6	

Y como tambien aveces puede resultar pruebo si en la dll es posible Registrarnos con las variables:



Colocando un bp en ejecucion en GetEnvironmentVariableA tendremos las variables necesarias

Con esto se crea por ejemplo un about:



Pd:como el objetivo era descargar un armadillo y aprender de las variables, pues desinstalo la aplicacion y borro las ramas de armadillo con Trial Reset

Saludos Apuromafo