

CRACKS LATINOS



Programa	Privacy Dummy! 1.1		
Protección	-----		
Descripción	Un programa		
Dificultad	baja		
DownLoad	http://www.dummysoftware.com/		
Herramienta	Olly		
Cracker	La Calavera	Fecha	

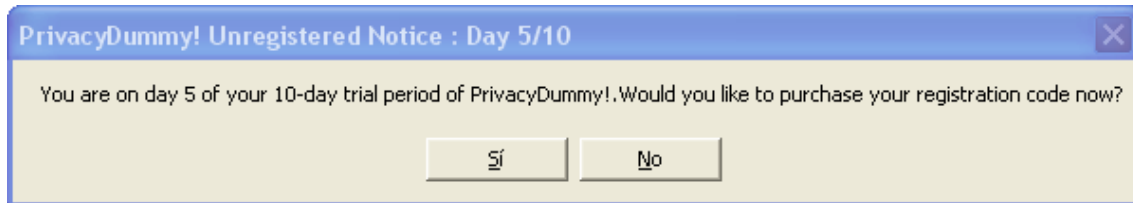
INTRODUCCION

Bueno aquí aburrido jejeje me puse a ver varios programas para exploit pero como no encontré algún bug pues me decidí a crackear este programa en un principio no iba a hacer tute ya que parecía muy sencillo pero me salio con un CRC y bueno aquí esta.

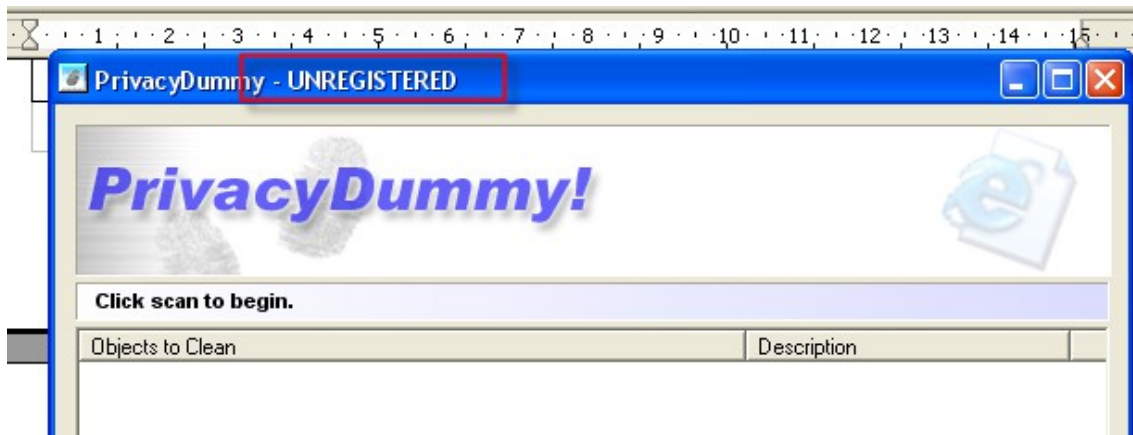
AL ATAKE

Bueno comenzando.

Como ya hace días lo tengo instalado me sale este cartelito

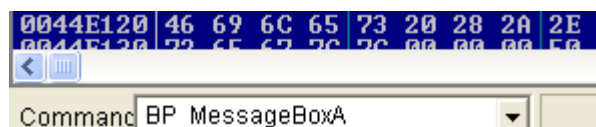


Como verán 5 de 10 días de uso o algo así mi ingles es muy malo :-P le doy a no y arranca el programa

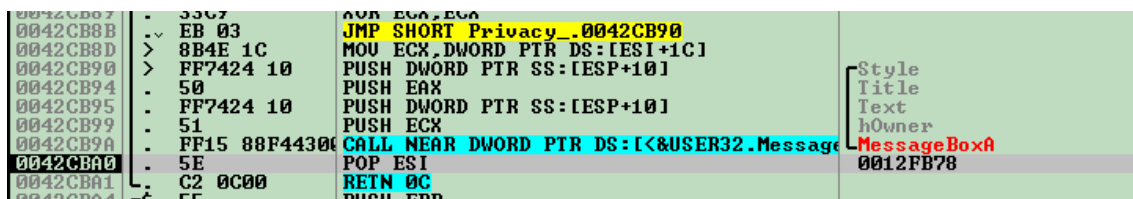


como ven nos pone un lindo UNREGISTERED pero si busco entre las string no hay nada (las string están bastante bien ocultas las mayoría claro) así que como no me quise romper la cabeza me decidí a atacar por el mensaje inicial ya que un MessageBoxA y así buscar un indicio de por donde saltarnos la registracion

bueno lo cargamos en el Olly y ponemos un BP en MessageBoxA



Y le damos a F9 y cuando para aceptamos el mensaje y comenzamos a tracear hasta llegar al programa en si



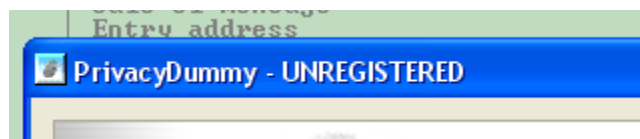
como ven ahí tenemos la llama a la API así que seguimos traceando para ver desde donde es llamada

0040E89B	52	PUSH EDI	
0040E89C	8BCE	MOV ECX,ESI	
0040E89E	C68424 980000	MOV BYTE PTR SS:[ESP+98],5	
0040E8A6	E8 C7E20100	CALL Privacy_.0042CB72	
0040E8AB	83F8 06	CMP EAX,6	
0040E8AE	8D4C24 10	LEA ECX,DWORD PTR SS:[ESP+10]	
0040E8B2	0F94C3	SETB BL	

Ese es el CALL que llama al mensaje y si subimos vemos que estamos dentro de otro call así que traceamos hasta el final y vemos

0040DAF4	75 07	JNZ SHORT Privacy_.0040DAFD	Evita el Mensaje
0040DAF6	8BCD	MOV ECX,EBP	
0040DAF8	E8 330C0000	CALL Privacy_.0040E730	Sale el mensaje
0040DAFD	68 E6E84200	PUSH Privacy_.0042E8E6	Entry address
0040DB02	6A 05	PUSH 5	

Bueno como ven justo por encima tenemos el salto que evita el mensaje así que ponemos un BP en el mensaje y reiniciamos y cuando para cambiamos el FLAG y arranca el programa sin el mensaje inicial o sea que ya evitamos que caduque pero

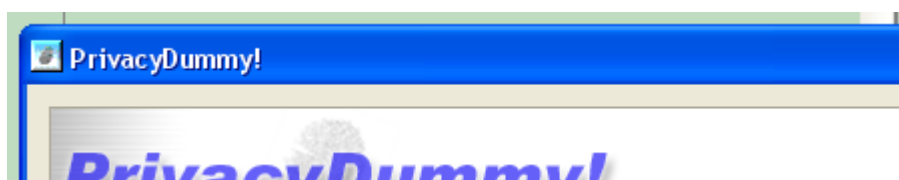


Como ven sigue diciendo UNREGISTERED y eso es feo jejeje ahora bien como se ve por encima del salto hay una direccion que se compara con BL así que ponemos un BP en la comparación y reiniciamos nuevamente

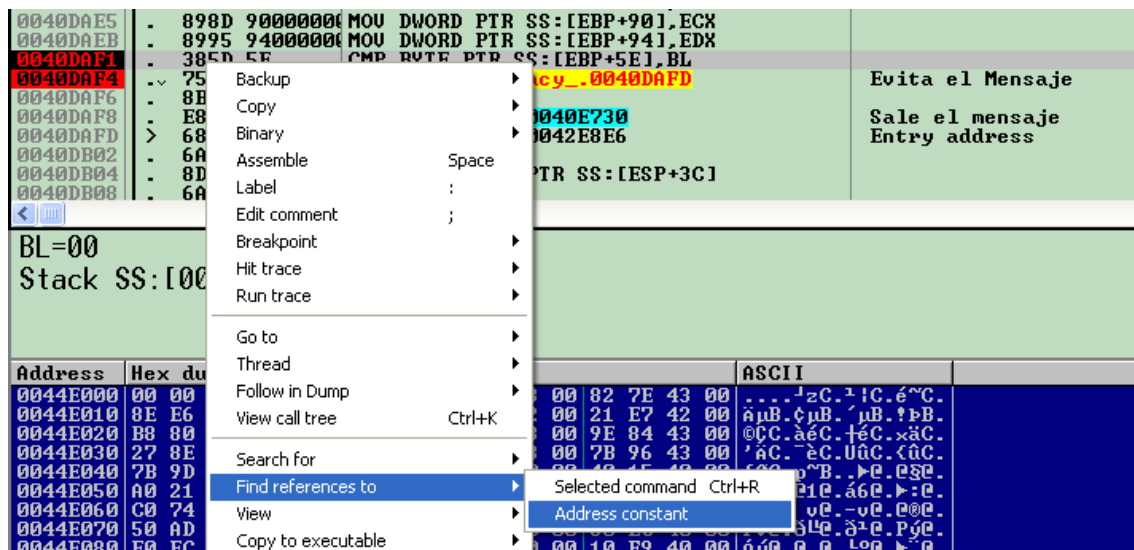
0040DAF4	75 07	JNZ SHORT Privacy_.0040DAFD	Evita el Me
0040DAF6	8BCD	MOV ECX,EBP	
0040DAF8	E8 330C0000	CALL Privacy_.0040E730	Sale el men

BL=00	←
Stack SS:[0012FBD6]=00	←

Como ven ambas valen 0 así que cambiamos el 0 de 12fbd6 (ojo esa direccion es en mi caso ya que es un valor de la pila) lo cambio a 1 y le doy a F9

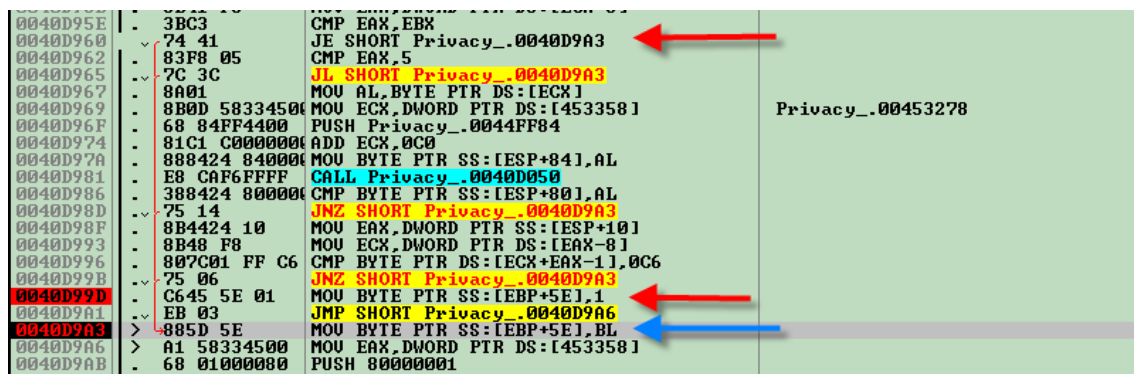


Como ven arranca el programa y registrado bien ahora hay que buscar meter un 1 en esa direccion y para ello vamos a buscarlo



Address	Disassembly	Comment
0040D99D	MOV BYTE PTR SS:[EBP+5E],1	
0040D9A3	MOV BYTE PTR SS:[EBP+5E],BL	
0040DAF1	CMP BYTE PTR SS:[EBP+5E],BL	(Initial CP
0040E13C	LEA ECX,DWORD PTR DS:[EAX+EBX*2+5E]	
0040E428	MOV AL,BYTE PTR SS:[EBP+5E]	
0040E50E	MOV BYTE PTR SS:[EBP+5E],BL	
0040E6D6	MOV BYTE PTR SS:[EBP+5E],BL	

ese es el listado esta vez son pocos así que le vamos a poner un BP en cada uno a ver en cual para reiniciamos



Como vemos en la imagen para en 40D9A3 y justo por encima tenemos el mov 1 a esa direccion y un poco mas arriba tenemos un salto al BP que estamos así que cambiamos

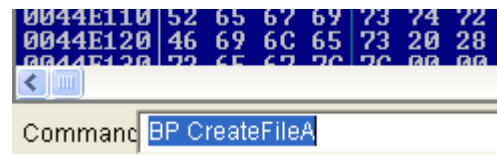
ese salto por un JMP 40D99D y siempre esa direccion valdrá 1 con lo cual estaremos registrados.

Bueno guardamos los cambios con algún otro nombre y vamos a ver como nuestro programa quedo crackeado lo cargo desde fuera del Olly y.... el programa no arranca ???? bueno como dije antes el muy condenado tiene un CRC (o sea que comprueba que todos sus byte estén correcto sin modificación).

Bueno cargamos el programa modificado dentro del Olly y lo primero que intente fue ver si lee en la memoria los byte que hay en la sección de registro (antes funcionaba jejejeje) y nada se cierra sin leer un solo byte hmmmmm

Bueno como ya dije antes estuve mirándolo para ver si encontraba algún bug para exploit y lo primero que mire fue ver que archivos cargaba y el primero que carga es el ejecutable, entonces veamos que hace.

Cargamos el modificado y ponemos un BP en CreateFileA



y le damos a F9, una vez que para traceamos para salir de la API hasta el programa y continuamos traceando hasta salir del CALL

0041F795	-	8BC8	MOU ECX,EAX
0041F797	-	83C4 10	ADD ESP,10
0041F79A	-	3BCB	CMP ECX,EBX
0041F79C	-	7D 04	JGE SHORT Privacy_.0041F7A2
0041F79E	>	33C0	XOR EAX,EAX
0041F7A0	-	EB 1A	JMP SHORT Privacy_.0041F7BC
0041F7A2	>	8B45 14	MOU EAX,DWORD PTR SS:[EBP+14]
0041F7A5	-	FF05 3C554500	INC DWORD PTR DS:[45553C]
0041F7AB	-	8970 0C	MOU DWORD PTR DS:[EAX+C],ESI
0041F7AE	-	8958 04	MOU DWORD PTR DS:[EAX+4],EBX
0041F7B1	-	8918	MOU DWORD PTR DS:[EAX],EBX
0041F7B3	-	8958 08	MOU DWORD PTR DS:[EAX+8],EBX
0041F7B6	-	8958 1C	MOU DWORD PTR DS:[EAX+1C],EBX
0041F7B9	-	8948 10	MOU DWORD PTR DS:[EAX+10],ECX
0041F7BC	>	5F	POP EDI
0041F7BD	-	5E	POP ESI
0041F7BE	-	5B	POP EBX
0041F7BF	-	C9	LEAVE
0041F7C0	-	C3	RETN

Como vemos no hay nada así que seguimos hasta el retn y así vamos saliendo de los distintos CALL hasta llegar a esta parte

```

0040D385 . E8 B4B80000 CALL Privacy_.00418C3E
0040D38A . 8BD8      MOV EBX,EAX
0040D38C . 83C4 08    ADD ESP,8
0040D38F . 85DB      TEST EBX,EBX
0040D391 . 0F84 0A010000 JE Privacy_.0040D4A1
0040D397 . 8B35 B8054400 MOV ESI,DWORD PTR DS:[4405B8]
0040D39D . 8D9424 240100 LEA EDX,DWORD PTR SS:[ESP+124]
0040D3A4 . C78424 240100 MOV DWORD PTR SS:[ESP+124],0
0040D3AF . 8D8424 280100 LEA EAX,DWORD PTR SS:[ESP+128]
0040D3B6 . BF 80000000 MOV EDI,80
0040D3BB > F642 03 80 TEST BYTE PTR DS:[EDX+3],80
0040D3BF . 8B0A      MOV ECX,DWORD PTR DS:[EDX]
0040D3C1 . 74 0B     JE SHORT Privacy_.0040D3CE
0040D3C3 . 03C9      ADD ECX,ECX
0040D3C5 . 8908      MOV DWORD PTR DS:[EAX],ECX
0040D3C7 . 33CE      XOR ECX,ESI
0040D3C9 . 8948 FC   MOV DWORD PTR DS:[EAX-4],ECX
0040D3CC > EB 09     JMP SHORT Privacy_.0040D3D7
0040D3CE > 03C9      ADD ECX,ECX
0040D3D0 . 8948 FC   MOV DWORD PTR DS:[EAX-4],ECX
0040D3D3 . 33CE      XOR ECX,ESI
0040D3D5 . 8908      MOV DWORD PTR DS:[EAX],ECX
0040D3D7 > 83C0 08   ADD EAX,8
0040D3DA . 83C2 04   ADD EDX,4
0040D3DD . 4F        DEC EDI
0040D3DE . 75 DB     JNZ SHORT Privacy_.0040D3BB
0040D3E0 . 8B4424 10 MOV EAX,DWORD PTR SS:[ESP+10]
0040D3E4 . 53        PUSH EBX
0040D3E5 . 50        PUSH EAX
0040D3E6 . 33F6      XOR ESI,ESI
0040D3E8 . 6A 01     PUSH 1

```

Stack DS:[00122333]=5A ('Z')

Bueno cuando salimos del CALL vemos que en 40D385 esta el CALL que llama a la API y un poco mas abajo hay un bucle que empieza a tomar valores y comienza a hacer una series de operaciones y como verán unos de los valores es la “Z” que es la cabecera del binario.

Bueno vamos por buen camino jejeje, seguimos traceando y encontramos otro bucle seguimos hasta el retn y salimos del call

```

0040C95B . 8B8D A4EBFFFI MOV ECX,DWORD PTR SS:[EBP-145C]
0040C961 . E8 7A090000 CALL Privacy_.0040D2E0
0040C966 . 8985 A0EBFFFI MOV DWORD PTR SS:[EBP-1460],EAX
0040C96C . 8B95 A0EBFFFI MOV EDX,DWORD PTR SS:[EBP-1460]
0040C972 . 8995 9CEBFFFI MOV DWORD PTR SS:[EBP-1464],EDX
0040C978 . 83AD 9CEBFFFI SUB DWORD PTR SS:[EBP-1464],1
0040C97F . 83BD 9CEBFFFI CMP DWORD PTR SS:[EBP-1464],0
0040C986 . 74 02     JE SHORT Privacy_.0040C98A
0040C987 > EB 07     JMP SHORT Privacy_.0040C991
0040C98A . 6A 00     PUSH 0
0040C98C > E8 38BE0000 CALL Privacy_.004187C9
0040C991 . 68 F8044500 PUSH Privacy_.004504F8
0040C996 . 8D4D E8   LEA ECX,DWORD PTR SS:[EBP-18]
0040C997 . E8 B61F0200 CALL Privacy_.0042E954

```

estamos FRITOS
ASCII "Loading sound files."

Bueno como verán en la imagen tenemos con la flecha roja donde salimos en la flecha verde el call que si pasamos en el chau programa se cierra y justo por encima un JE y un JMP si JE salta nos manda directo al CALL así que nopeamos el JE y comenzamos a tracear

```

0040C97F . 83BD 9CEBFFFI CMP DWORD PTR SS:[EBP-1464],0
0040C986 . 74 02     JE SHORT Privacy_.0040C98A
0040C987 > EB 07     JMP SHORT Privacy_.0040C991
0040C98A . 6A 00     PUSH 0
0040C98C > E8 38BE0000 CALL Privacy_.004187C9
0040C991 . 68 F8044500 PUSH Privacy_.004504F8
0040C996 . 8D4D E8   LEA ECX,DWORD PTR SS:[EBP-18]

```

estamos FRITOS
ASCII "Loading sound files."

Como vemos el JMP evita el call le damos a F9 y.....



El programa arranca sin ningún problema guardamos los cambios y lo probamos fuera del olly y el programa arranca sin ningún problema y registrado.

Bueno eso es todos ya saben dudas solo pregunten.

Saludos a toda la lista.

Daniel –La Calavera -

