

# CRACKSLATINOS 2007

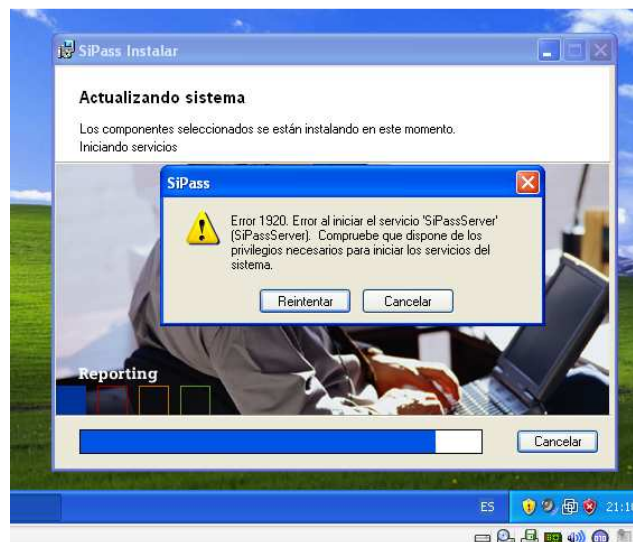
## ORCA+OLLY+WINDBG+VMWARE II

---

ESTE DOCUMENTO ES DE CARÁCTER PUBLICO, SI LO COMPRASTE PIDE QUE TE DEVUELVAN TU DINERO.

Continuando.....

Cargamos la VM, llevamos el msi parchado y reinstalamos ahí:



En este punto, el windbg ya esta depurando a la vm (teoría 722 del maestro), sabemos que el servicio ya esta instalado y se niega a correr.

Lo que tenemos que hacer es depurar el servicio, tenemos que romper en el debugger justo donde sabemos que llama a la dll de las licencias.

No se me ocurre mucho, así que robe la idea del armadillo team: nanomites.

Pondremos INT3 cerca de cada llamada y veremos los valores de retorno. Podemos tener el exe abierto en olly y guardar cambios, y usamos el service manager para arrancarlo en la vm.

Veamos de nuevo los accesos al registro:

AscoS...	Query...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\LocalServer32\LocalServer32	
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB\ASCOCardTechnology	0x9
AscoS...	CloseK...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB	
AscoS...	Create...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB	Access: 0xF003F
AscoS...	CloseK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\LocalServer32	
AscoS...	OpenK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\InprocServer32	
AscoS...	OpenK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\InprocServer\86	
AscoS...	OpenK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\InprocHandler32	
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB\ASCOSiteName	"crackslatinos"
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB\ASCOSiteName	"crackslatinos"
AscoS...	CloseK...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB	
AscoS...	OpenK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\InprocHandler\86	
AscoS...	OpenK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\LocalServer32	Access: 0x2000000
AscoS...	Query...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\LocalServer32(Default)	"C:\Archivos de programa\SiPass\AscoS...
AscoS...	Create...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB	Access: 0xF003F
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB\ASCOValidityCheck	"AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB\ASCOValidityCheck	"AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
AscoS...	CloseK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}\LocalServer32	
AscoS...	CloseK...	HKLM\Software\Classes\CLSID\{039E6362-8611-11D3-9043-00105A6E778D}	
AscoS...	OpenK...	HKLM\System\CurrentControlSet\Control\ComputerName	Access: 0x20019
AscoS...	CloseK...	HKLM\SOFTWARE\Landis & Staefa\Advantage\Version4\Server\ServerConfigurations\JELB	
AscoS...	Create...	HKLM\SOFTWARE\Landis & Staefa\ADVANTAGE\Version4\AdvantageLog\AscoServer	Access: 0xF003F
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\ADVANTAGE\Version4\AdvantageLog\AscoServer\Level	0x0
AscoS...	OpenK...	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	Access: 0x20019
AscoS...	Query...	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName\ComputerName	"JELB"
AscoS...	CloseK...	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	
AscoS...	CloseK...	HKLM\SOFTWARE\Landis & Staefa\ADVANTAGE\Version4\AdvantageLog\AscoServer	
AscoS...	Create...	HKLM\SOFTWARE\Landis & Staefa\ADVANTAGE\Version4\AdvantageLog\AscoServer	Access: 0xF003F
AscoS...	Query...	HKLM\SOFTWARE\Landis & Staefa\ADVANTAGE\Version4\AdvantageLog\AscoServer\LogFile	"C:\TEMP\AscoServer.EVT"
AscoS...	CloseK...	HKLM\System\CurrentControlSet\Control\ComputerName	

Vemos que accesa el SiteName, veamos las Apis:

00480B02	CALL	<JMP.&PXUTILR.ASC0GetSMsgrPCPEndPoint	PXUTILR.ASC0GetSMsgrPCPEndPoint
00480A94	CALL	<JMP.&PXUTILR.ASC0GetSMsgrPCProtoSeq	PXUTILR.ASC0GetSMsgrPCProtoSeq
00480B0D	CALL	<JMP.&PXUTILR.ASC0GetSMsgrPCProtoSeq	PXUTILR.ASC0GetSMsgrPCProtoSeq
00451E58	CALL	<JMP.&PXUTILR.ASC0GetSMsgrMessageOrig	PXUTILR.ASC0GetSMsgrMessageOriginator
00451E7A	CALL	<JMP.&PXUTILR.ASC0GetSMsgrMessageRece	PXUTILR.ASC0GetSMsgrMessageReceiver
00451D30	CALL	<JMP.&PXUTILR.ASC0GetValidityLicence	PXUTILR.ASC0GetValidityLicenceName
004516FA	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
004517DE	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
004518C4	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
0048098E	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
00480A24	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
00480BAD	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
00482323	CALL	<JMP.&PXUTILR.ASC0GetValidityModule	PXUTILR.ASC0GetValidityModule
00451CDD	CALL	<JMP.&PXUTILR.ASC0GetValidityNumber	PXUTILR.ASC0GetValidityNumberOfCCTV
004502B6	CALL	<JMP.&PXUTILR.ASC0GetValidityNumber	PXUTILR.ASC0GetValidityNumberOfClients
00451CCD	CALL	<JMP.&PXUTILR.ASC0GetValidityNumber	PXUTILR.ASC0GetValidityNumberOfClients
00480B37	CALL	<JMP.&PXUTILR.ASC0SetServerName	PXUTILR.ASC0SetServerName
004717A5	CALL	<JMP.&PXUTILR.?AsyncSend@AsyncQueue	PXUTILR.?AsyncSend@AsyncQueueCommand@@SAXPAU1@@Z
0043CE13	CALL	DWORD PTR DS:[<&MSUCR71._atoi64>]	MSUCR71._atoi64
0043CE6C	CALL	DWORD PTR DS:[<&MSUCR71._atoi64>]	MSUCR71._atoi64
0041ED40	CALL	DWORD PTR DS:[<&MSUCR71._atoi>]	MSUCR71._atoi
0041E8F5	CALL	DWORD PTR DS:[<&MSUCR71._atoi>]	MSUCR71._atoi

ValidityLicenceName es una candidata que resalta.

Veamos la zona a la llamada:

Windows XP Professional

[CPU - main thread, module AscoSrv]

File View Debug Plugins Options Window Help

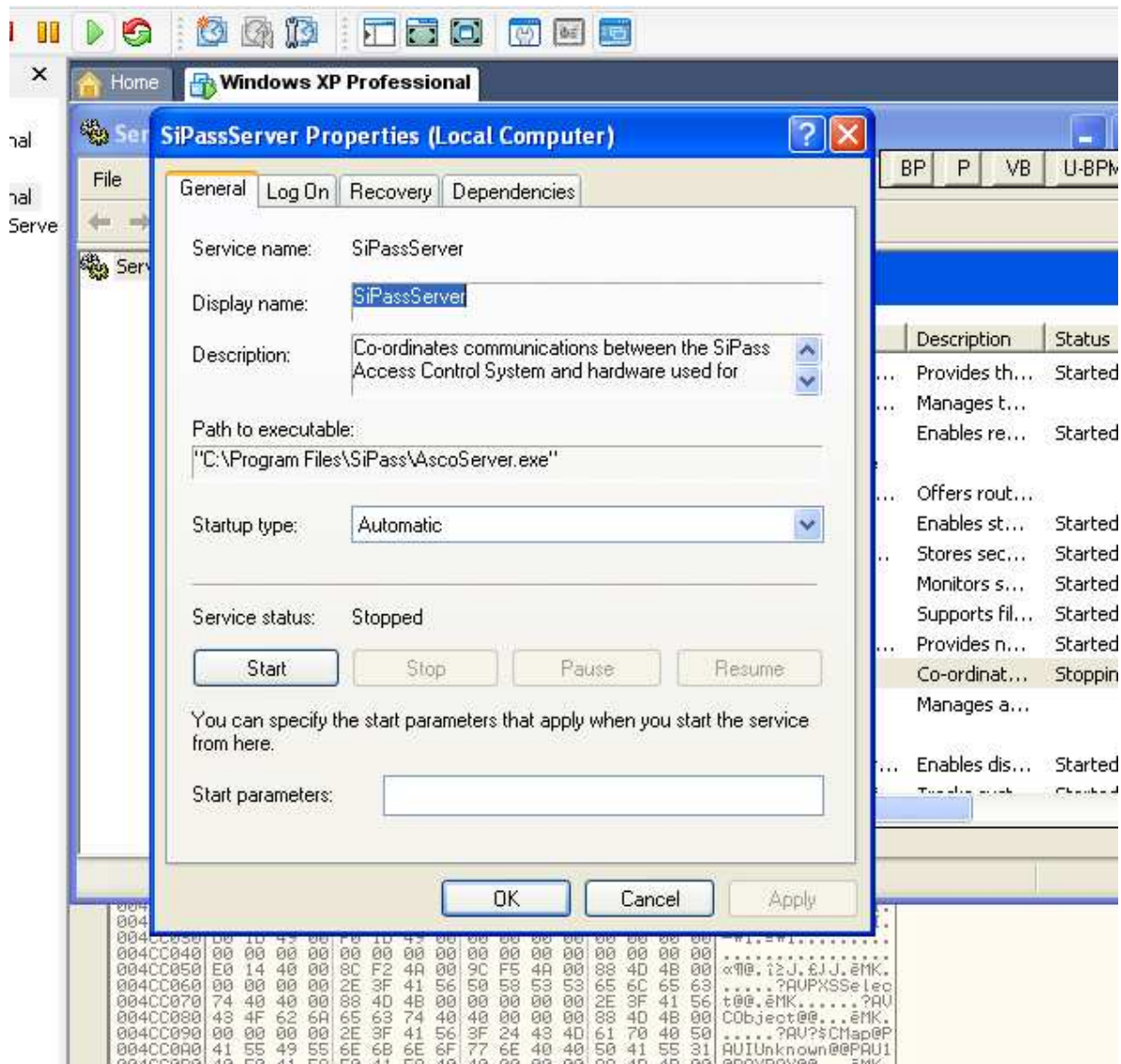
BP P VB U-BPM

Address Hex dump Disassembly Comment

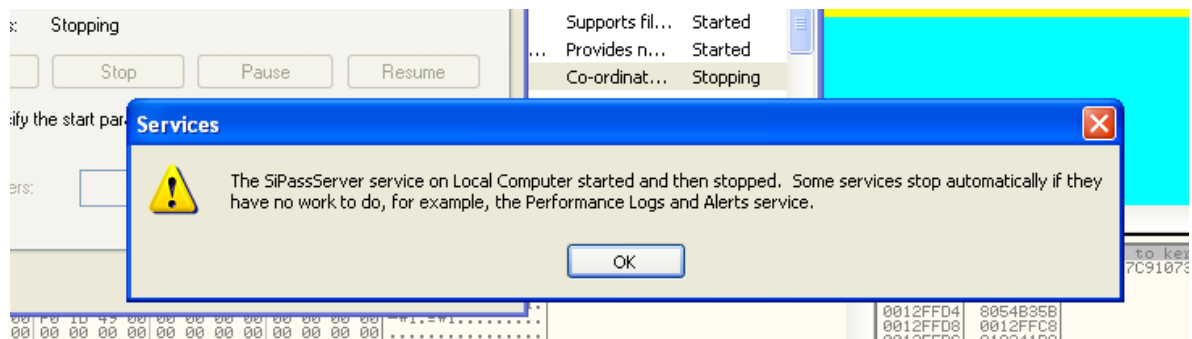
00451D2A	. C645 FC 02	MOV BYTE PTR SS:[EBP-4],2	
00451D2E	. 6A 00	PUSH 0	
00451D30	. E8 DF170300	CALL <JMP.6PXUTILR.ASCOGetValidityLicenceName>	
00451D35	. 83C4 04	ADD ESP,4	
00451D38	. 50	PUSH EAX	
00451D39	. 8D95 3CFDFFFF	LEA EDI,DWORD PTR SS:[EBP-2C4]	
00451D3F	. 52	PUSH EDI	
00451D40	. FF15 10234900	CALL DWORD PTR DS:[<MFC71.#2322>]	MFC71.7C146A9D
00451D46	. 83C4 08	ADD ESP,8	
00451D49	. 68 F42A4A00	PUSH AscoSrv.004A2AF4	ASCII "DEMO"
00451D4E	. 8D8D 3CFDFFFF	LEA ECX,DWORD PTR SS:[EBP-2C4]	
00451D54	. FF15 A0224900	CALL DWORD PTR DS:[<MFC71.#1486>]	MFC71.7C188CD7
00451D5A	. 85C0	TEST EAX,EAX	
00451D5C	~ 0F85 AB000000	JNZ AscoSrv.00451E0D	
00451D62	. C785 ECFCFFFF 1C3C49	MOV DWORD PTR SS:[EBP-314],AscoSrv.00493C1C	
00451D6C	. C785 F0FCFFFF 000000	MOV DWORD PTR SS:[EBP-310],0	
00451D76	. C785 F4FCFFFF C50900	MOV DWORD PTR SS:[EBP-30C],9C5	
00451D80	. C645 FC 03	MOV BYTE PTR SS:[EBP-4],3	
00451D84	. 68 FC2A4A00	PUSH AscoSrv.004A2AFC	
00451D89	. 8D85 F8FCFFFF	LEA EAX,DWORD PTR SS:[EBP-308]	format = "employeeCemp_id0"
00451D8F	. 50	PUSH EAX	s
00451D90	. FF15 9C234900	CALL DWORD PTR DS:[<MSVCRT71.printf@16>]	printf

Pues ahí valida que no sea demo, ponemos el int3 en 451d5a, guardamos el exe y rearrancamos el servicio:

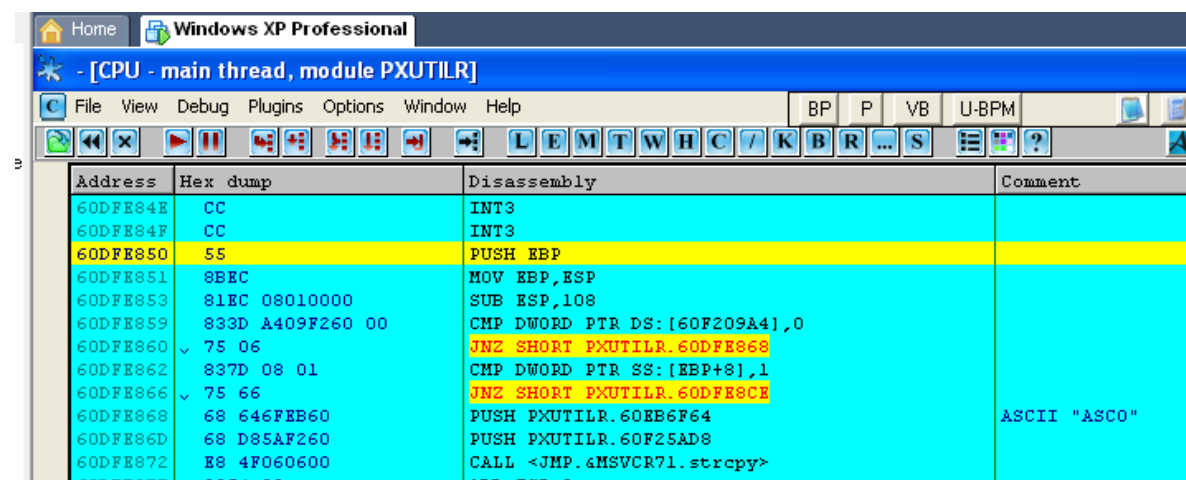
Address	Hex dump	Disassembly	Comment
00451D2A	. C645 FC 02	MOV BYTE PTR SS:[EBP-4],2	
00451D2E	. 6A 00	PUSH 0	
00451D30	. E8 DF170300	CALL <JMP.&PMUTILR.ASCOGetValidityLicenceName>	
00451D35	. 83C4 04	ADD ESP,4	
00451D38	. 50	PUSH EAX	
00451D39	. 8D95 3CFDFFFF	LEA EDX,DWORD PTR SS:[EBP-2C4]	
00451D3F	. 52	PUSH EDX	
00451D40	. FF15 10234900	CALL DWORD PTR DS:[<&MFC71.#2322>]	MFC71.7C146A9D
00451D46	. 83C4 08	ADD ESP,8	
00451D49	. 68 F42A4A00	PUSH AscoServ.004A2AF4	ASCII "DEMO"
00451D4E	. 8D8D 3CFDFFFF	LEA ECX,DWORD PTR SS:[EBP-2C4]	
00451D54	. FF15 A0224900	CALL DWORD PTR DS:[<&MFC71.#1486>]	MFC71.7C188CD7
00451D5A	CC	INT3	
00451D5E	90	NOP	
00451D5C	.. 0F85 AB000000	JNZ AscoServ.00451E0D	
00451D62	. C785 8CFDFFFF 1C3C4900	MOV DWORD PTR SS:[EBP-314],AscoServ.00493C1C	
00451D6C	. C785 F0FCFFFF 00000000	MOV DWORD PTR SS:[EBP-310],0	
00451D76	. C785 F4FCFFFF C5090000	MOV DWORD PTR SS:[EBP-30C],9C5	
00451D80	. C645 FC 03	MOV BYTE PTR SS:[EBP-4],3	
00451D84	. 68 FC2A4A00	PUSH AscoServ.004A2AFC	format = "employeeEmp_id"
00451D89	. 8D85 F8FCFFFF	LEA EAX,DWORD PTR SS:[EBP-308]	
00451D8F	. 50	PUSH EAX	



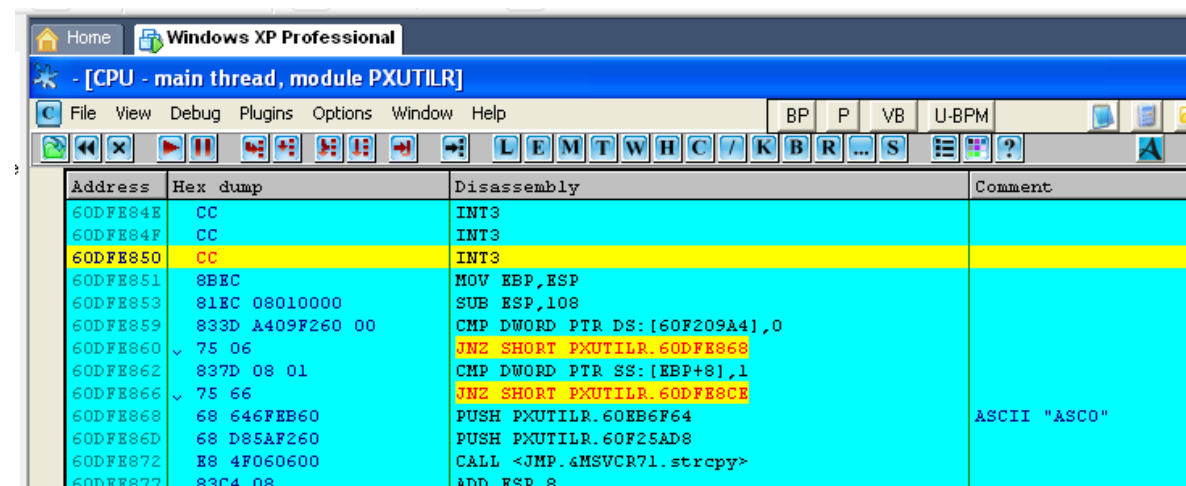
Al darle start:



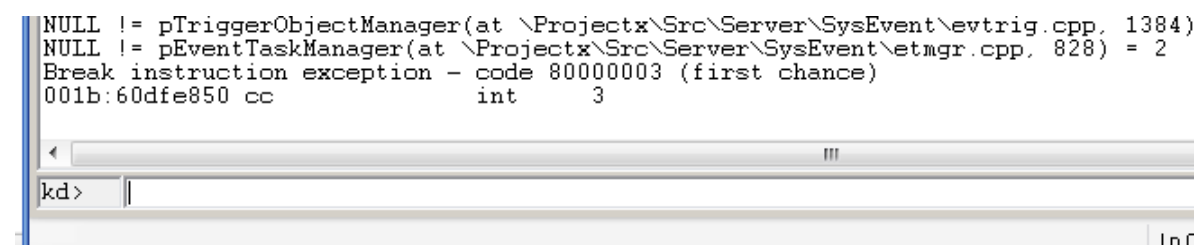
Pues no hubo interrupción, el código del exe del servicio no pasa por ahí, veamos si alguien mas usa la api, pondremos el int3 dentro de la dll asi:



Ahí en ValidityLicenceName, luego:



Reiniciamos el servicio y:



Ahí salto el debugger, no es ascosever.exe el primero en acceder el registro, es alguien mas, veamos quien es:

```

Break instruction exception - code 80000003 (first chance)
001b:60dfe850 cc          int     3
kd> eb eip 55
kd> u
001b:60dfe850 55          push    ebp
001b:60dfe851 8bec        mov     ebp,esp
001b:60dfe853 81ec08010000 sub     esp,108h
001b:60dfe859 833da409f26000 cmp     dword ptr ds:[60F209A4h],0
001b:60dfe860 7506        jne     60dfe868
001b:60dfe862 837d0801    cmp     dword ptr [ebp+8],1
001b:60dfe866 7566        jne     60dfe8ce
001b:60dfe868 68646feb60 push    60EB6F64h
kd> d esp
00aafa94 e2 b7 df 60 00 00 00 00-00 00 00 00 04 5d 88 8a ...]...
00aafaa4 48 00 00 00 2a 54 de 77-58 66 f2 60 03 00 00 00 H...*T.wXf...
00aafab4 38 54 de 77 50 fd aa 00-00 00 00 00 00 00 00 00 8T.wP...
00aafac4 24 54 de 77 f2 1d de 77-00 00 00 00 00 00 00 00 $T.w...w...
00aafad4 00 00 00 00 00 00 00 00-10 fb aa 00 00 00 00 3f 00 ...?
00aafae4 32 07 91 7c 06 00 00 00-a8 07 3f 00 00 00 00 3f 00 2...|...?...?
00aafaf4 00 00 00 00 e8 fa aa 00-00 00 00 00 2c fd aa 00 ...
00aafb04 18 ee 90 7c 38 07 91 7c-ff ff ff 32 07 91 7c ...|8...|...2...

```

Ahí se ve el código de la api, y la pila, donde esta la dirección de retorno, 60dfb7e2, damos g al windbg y veamos en el olly de la vm esa dirección que al parecer es de la dll:

Home Windows XP Professional			
[CPU - main thread, module pxutilr]			
File View Debug Plugins Options Window Help			
BP P VB U-BPM			
L E M T W H C / K B R S			
Address	Hex dump	Disassembly	Comment
60DFB7C8	- 8B8D 74FEFFFF	MOV ECX,DWORD PTR SS:[EBP-18C]	
60DFB7CE	- 51	PUSH ECX	
60DFB7CF	- 8D8D A0FEFFFF	LEA ECX,DWORD PTR SS:[EBP-160]	
60DFB7D5	- FF15 1C44E760	CALL DWORD PTR DS:[<&MFC71.#784>]	MFC71.7C14FF74
60DFB7DB	- 6A 00	PUSH 0	
60DFB7DD	- E8 6E300000	CALL pxutilr.ASCOGGetValidityLicenceName	
60DFB7E2	- 83C4 04	ADD ESP,4	
60DFB7E5	- 8985 70FEFFFF	MOV DWORD PTR SS:[EBP-190],EAX	
60DFB7EB	- 8B95 70FEFFFF	MOV EDX,DWORD PTR SS:[EBP-190]	
60DFB7F1	- 52	PUSH EDX	
60DFB7F2	- 8D8D A4FEFFFF	LEA ECX,DWORD PTR SS:[EBP-15C]	
60DFB7F8	- FF15 1C44E760	CALL DWORD PTR DS:[<&MFC71.#784>]	MFC71.7C14FF74
60DFB7FE	- 6A 00	PUSH 0	Arg1 = 00000000
60DFB800	- E8 5B2C0000	CALL pxutilr.ASCOGGetValiditySiteString	ASCOGGetValiditySiteString
60DFB805	- 83C4 04	ADD ESP,4	
60DFB808	- 50	PUSH EAX	[s

Pues ahí esta, vemos que la llamada es parte de una subrutina muy larga, buscamos el inicio:



Address	Hex dump	Disassembly	Comment
60DFB62B	CC	INT3	
60DFB62C	CC	INT3	
60DFB62D	CC	INT3	
60DFB62E	CC	INT3	
60DFB62F	CC	INT3	
60DFB630	55	PUSH EBP	
60DFB631	8BEC	MOV EBP,ESP	
60DFB633	6A FF	PUSH -1	
60DFB635	68 8DD7E660	PUSH pxutilr.60EBD78D	SE handler installation
60DFB63A	64:A1 00000000	MOV EAX,DWORD PTR FS:[0]	
60DFB640	50	PUSH EAX	
60DFB641	64:8925 00000000	MOV DWORD PTR FS:[0],ESP	
60DFB648	81EC 1C020000	SUB ESP,21C	
60DFB64E	898D E8FDFFFF	MOV DWORD PTR SS:[EBP-218],ECX	
60DFB654	C785 E8FEFFFF 01000000	MOV DWORD PTR SS:[EBP-118],1	
60DFB65E	837D 08 00	CMP DWORD PTR SS:[EBP+8],0	
60DFB662	75 53	JNZ SHORT pxutilr.60DFB6B7	
60DFB664	68 8866EB60	PUSH pxutilr.60EB6688	SRC = "Server"
60DFB669	8B85 E8FDFFFF	MOV EAX,DWORD PTR SS:[EBP-218]	

Ahora si analizamos la rutina hacia abajo, podemos ver como va accedendo los campos del registro:

Address	Hex dump	Disassembly	Comment
60DFB7CF	8D8D A0FEFFFF	LEA ECX,DWORD PTR SS:[EBP-160]	
60DFB7D5	FF15 1C44E760	CALL DWORD PTR DS:[<MFC71.7C14FF74>]	MFC71.7C14FF74
60DFB7DB	6A 00	PUSH 0	Arg1 = 00000000
60DFB7DD	E8 6E300000	CALL pxutilr.ASC0GetValidityLicenceName	ASC0GetValidityLicenceName
60DFB7E2	83C4 04	ADD ESP,4	
60DFB7E5	8985 70FEFFFF	MOV DWORD PTR SS:[EBP-190],EAX	
60DFB7EB	8B95 70FEFFFF	MOV EDX,DWORD PTR SS:[EBP-190]	
60DFB7F1	52	PUSH EDX	
60DFB7F2	8D8D A4FEFFFF	LEA ECX,DWORD PTR SS:[EBP-15C]	
60DFB7F8	FF15 1C44E760	CALL DWORD PTR DS:[<MFC71.7C14FF74>]	MFC71.7C14FF74
60DFB7FE	6A 00	PUSH 0	Arg1 = 00000000
60DFB800	E8 5B2C0000	CALL pxutilr.ASC0GetValiditySiteString	ASC0GetValiditySiteString
60DFB805	83C4 04	ADD ESP,4	
60DFB808	50	PUSH EAX	
60DFB809	FF15 5447E760	CALL DWORD PTR DS:[<MSVCR71.atoi>]	atoi
60DFB80F	83C4 04	ADD ESP,4	
60DFB812	8985 6CFEFFFF	MOV DWORD PTR SS:[EBP-194],EAX	
60DFB818	8B85 6CFEFFFF	MOV EAX,DWORD PTR SS:[EBP-194]	
60DFB81E	8985 A8FEFFFF	MOV DWORD PTR SS:[EBP-158],EAX	
60DFB824	6A 00	PUSH 0	Arg1 = 00000000
60DFB826	E8 951B0000	CALL pxutilr.ASC0GetValidityFacilityString	ASC0GetValidityFacilityString
60DFB82B	83C4 04	ADD ESP,4	

Luego vean como empieza a escribir los datos en una tabla, probablemente de donde el servicio leera luego:

* - [CPU - main thread, module pxutilr]			
File View Debug Plugins Options Window Help			
BP P VB U-BPM			
L E M T W H C 7 K B R S			
Address	Hex dump	Disassembly	Comment
60DFB892	- 8B8D 5CFEFFFF	MOV ECX,DWORD PTR SS:[EBP-1A4]	
60DFB898	- 898D B8FEFFFF	MOV DWORD PTR SS:[EBP-148],ECX	
60DFB89E	- 6A 00	PUSH 0	[Arg1 = 00000000]
60DFB8A0	- E8 8B340000	CALL pxutilr.60DFED30	pxutilr.60DFED30
60DFB8A5	- 83C4 04	ADD ESP,4	
60DFB8A8	- 8985 58FEFFFF	MOV DWORD PTR SS:[EBP-1A8],EAX	
60DFB8AE	- 8B95 58FEFFFF	MOV EDX,DWORD PTR SS:[EBP-1A8]	
60DFB8B4	- 8995 BCFEFFFF	MOV DWORD PTR SS:[EBP-144],EDX	
60DFB8BA	- 6A 00	PUSH 0	[Arg1 = 00000000]
60DFB8BC	- E8 0F340000	CALL pxutilr.60DFECD0	pxutilr.60DFECD0
60DFB8C1	- 83C4 04	ADD ESP,4	
60DFB8C4	- 8985 54FEFFFF	MOV DWORD PTR SS:[EBP-1AC],EAX	
60DFB8CA	- 8B85 54FEFFFF	MOV EAX,DWORD PTR SS:[EBP-1AC]	
60DFB8D0	- 8985 C0FEFFFF	MOV DWORD PTR SS:[EBP-140],EAX	
60DFB8D6	- 6A 00	PUSH 0	[Arg1 = 00000000]
60DFB8D8	- E8 B3340000	CALL pxutilr.60DFED90	pxutilr.60DFED90
60DFB8DD	- 83C4 04	ADD ESP,4	
60DFB8E0	- 8985 50FEFFFF	MOV DWORD PTR SS:[EBP-1B0],EAX	
60DFB8E6	- 8B8D 50FEFFFF	MOV ECX,DWORD PTR SS:[EBP-1B0]	
60DFB8EC	- 898D C4FEFFFF	MOV DWORD PTR SS:[EBP-13C],ECX	
60DFB8F2	- 6A 00	PUSH 0	[Arg1 = 00000000]
60DFB8F4	- E8 F7340000	CALL pxutilr.60DFEDF0	pxutilr.60DFEDF0

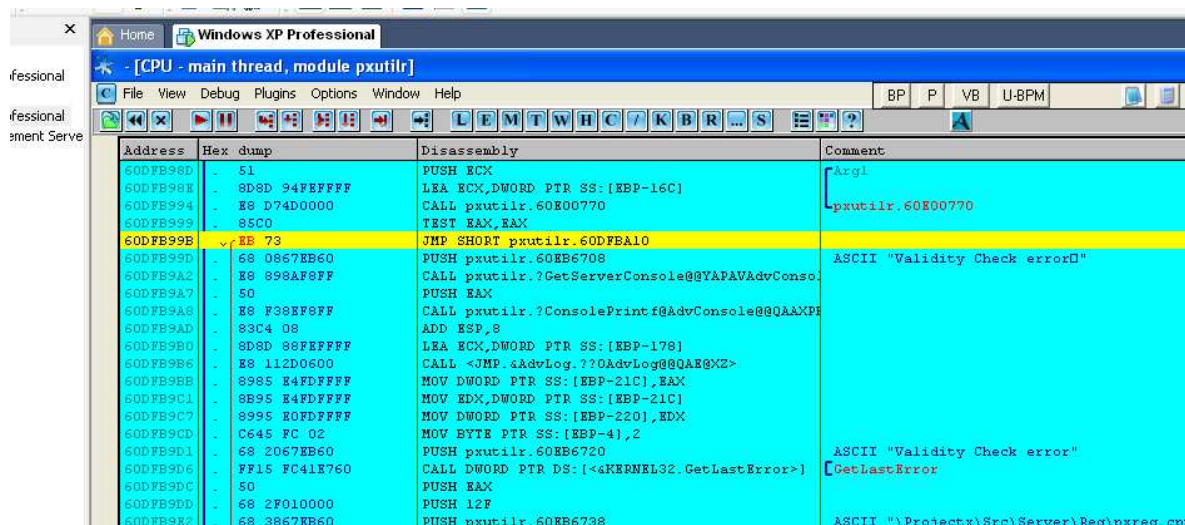
Lo interesante es al final de esta rutina:

* - [CPU - main thread, module pxutilr]			
File View Debug Plugins Options Window Help			
BP P VB U-BPM			
L E M T W H C 7 K B R S			
Address	Hex dump	Disassembly	Comment
60DFB94D	- 83C4 04	ADD ESP,4	
60DFB950	- 8985 40FEFFFF	MOV DWORD PTR SS:[EBP-1C0],EAX	
60DFB956	- 8B95 40FEFFFF	MOV EDX,DWORD PTR SS:[EBP-1C0]	
60DFB95C	- 52	PUSH EDX	
60DFB95D	- 8D8D D4FEFFFF	LEA ECX,DWORD PTR SS:[EBP-12C]	
60DFB963	- FF15 1C44E760	CALL DWORD PTR DS:[<MFC71.7C14FF74>]	MFC71.7C14FF74
60DFB969	- 6A 00	PUSH 0	
60DFB96B	- 68 00010000	PUSH 100	
60DFB970	- 8D85 F4FEFFFF	LEA EAX,DWORD PTR SS:[EBP-10C]	
60DFB976	- 50	PUSH EAX	
60DFB977	- 68 F466EB60	PUSH pxutilr.60EB66F4	ASCII "ASCOValidityCheck"
60DFB97C	- 8B8D E8FDFFFF	MOV ECX,DWORD PTR SS:[EBP-218]	
60DFB982	- E8 290B0000	CALL pxutilr.60DFC4B0	
60DFB987	- 8D8D F4FEFFFF	LEA ECX,DWORD PTR SS:[EBP-10C]	
60DFB98D	- 51	PUSH ECX	[Arg1
60DFB98E	- 8D8D 94FEFFFF	LEA ECX,DWORD PTR SS:[EBP-16C]	pxutilr.60E00770
60DFB994	- E8 D74D0000	CALL pxutilr.60E00770	
60DFB999	- 85C0	TEST EAX,EAX	
60DFB99B	74 73	JE SHORT pxutilr.60DFBA10	
60DFB99D	- 68 0867EB60	PUSH pxutilr.60EB6708	ASCII "Validity Check errorD"
60DFB9A2	- E8 898AF8FF	CALL pxutilr.?GetServerConsole@@YAPAVAdvConso	

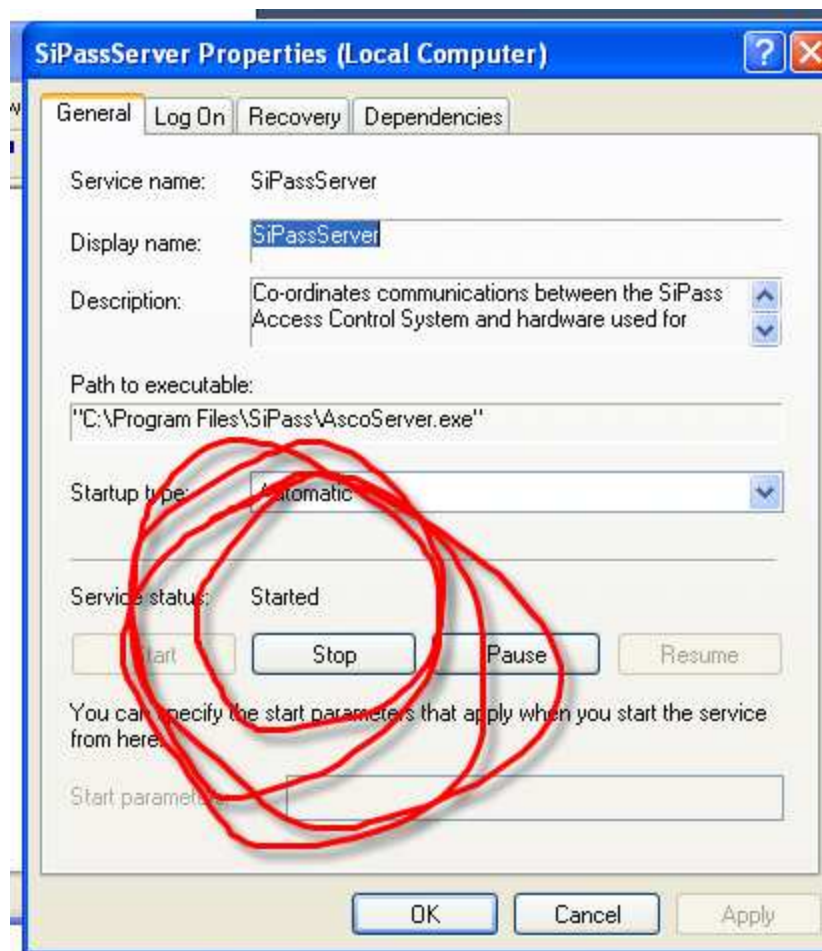
Bueno, ahí un salto mágico en 60dfb99d.

Con olly lo cambiamos para que siempre salte:





Y ahora reintentamos de nuevo arrancar el servicio:



Cls2007

Yes yes yes!!!!!!!!!!!!!!!

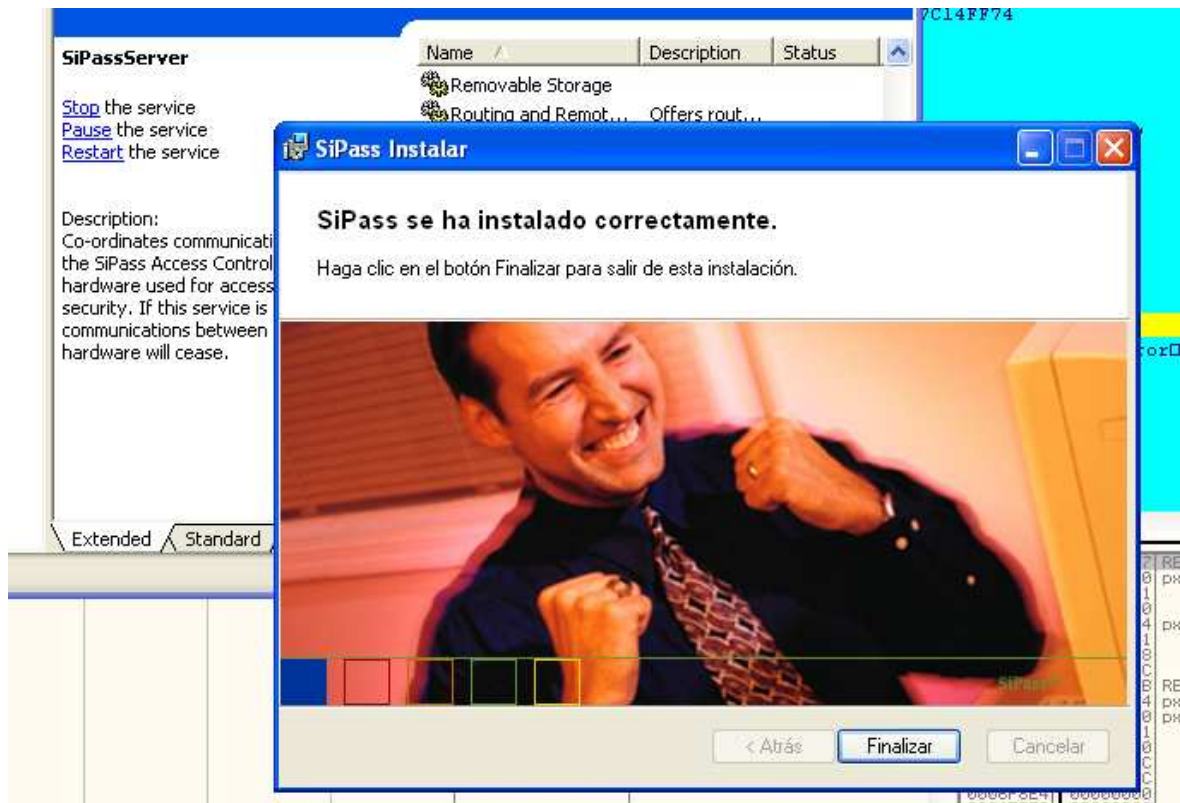
Volvamos al instalador, y le damos reintentar:



Yes yes ¡!!!!!!!!!!

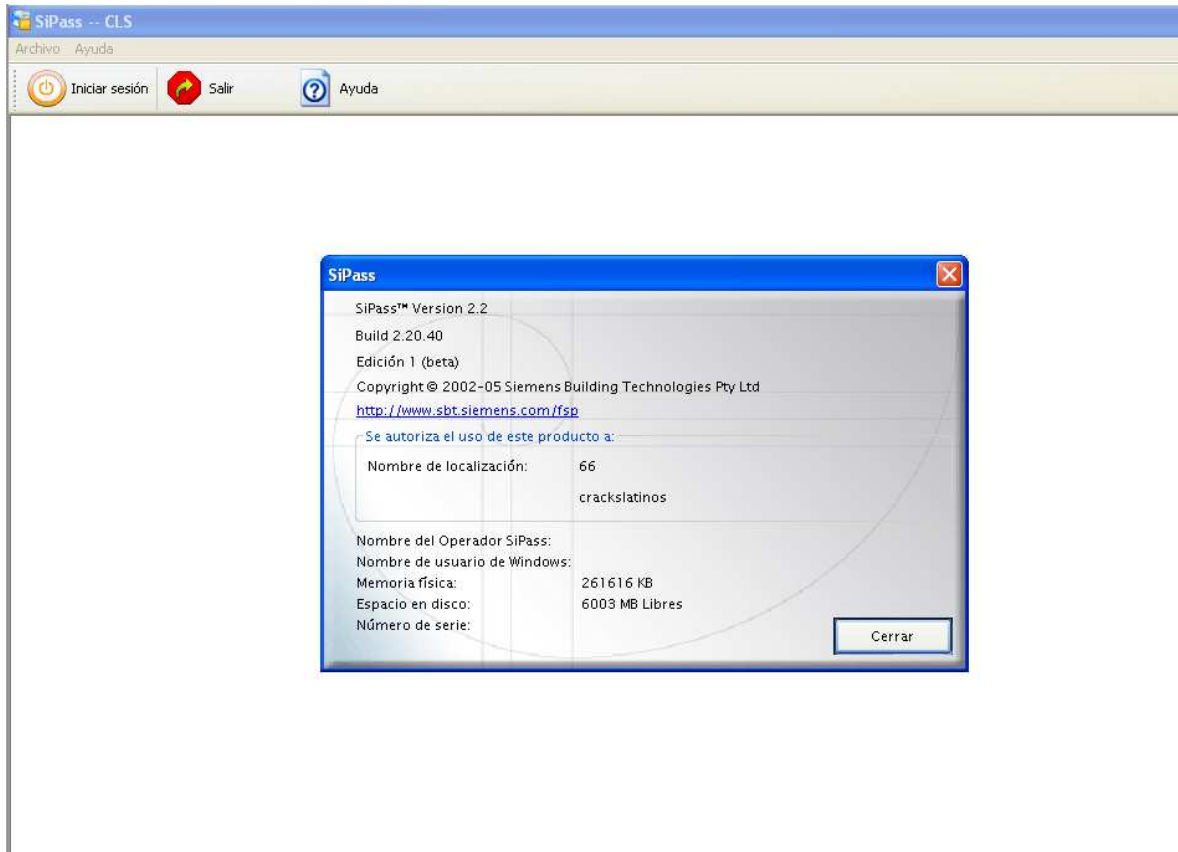
Y Luego:

Cls2007



Solo falta verificar si arranca registrado:}

Clis2007



Sip.

Ahí esta, después de mucho pensar y trabajar.

Estuvo difícil, pero lo aprendido es muchisisisisismo, eso es lo que importa.

Esto no hubiese podido llegar a feliz termino sin la ayuda y sabiduría de crackslatinos, un saludo a todos los listeros y gracias mil por el apoyo y sobre todo la amistad.

**CRACKSLATINOS 2007**

Joref.