

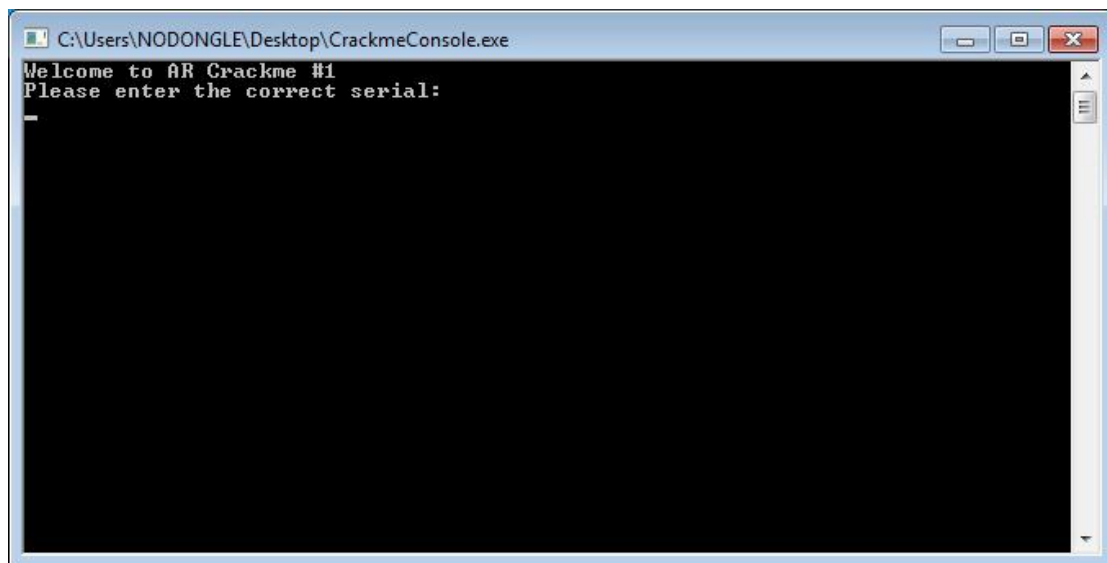
سری آموزش های R4NDOM-فارسی

قسمت ۱۱-فرق بین LAME و NOOB

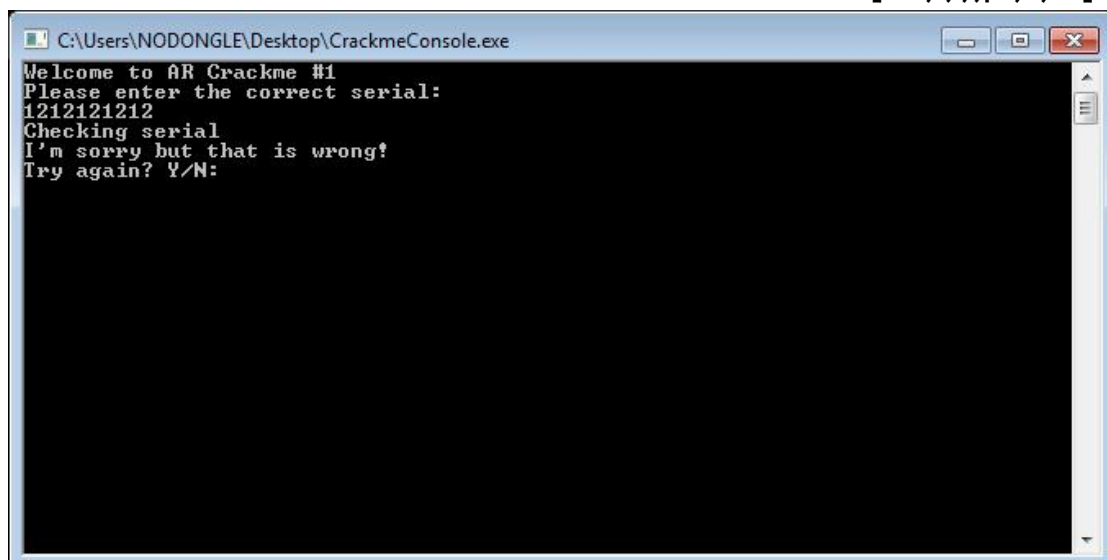
در این قسمت ما دوباره اقدام به پچ میکنیم ولی با این تفاوت که کمی عمیق تر ه مسئله نگاه میکنیم همچنین برنامه ی هدف یک CRACME تمت کنسول است و ما سعی فوایم کرد که پسورد مخفی شده را پیدا کنیم

بیایید شروع کنیم :

برنامه های کنسول ۳۲ بیتی مانند دیگر برنامه ها هستند با این تفاوت که آنها از رابط گرافیکی استفاده نمیکنند، برنامه هدف (ضمیمه شده) با نام CrackmeConsole.exe است، برنامه را یک بار اجرا کنید :



ساده است نه؟ یک رمز عبور وارد کنید :



فیلی فب، کلید "N" را برای خروج بفشارید:

```

C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
1212121212
Checking serial
I'm sorry but that is wrong!
Try again? Y/N:
n
Visit http://cracking.accessroot.com/ for everything AR
Long live the ARTeam!
Greetz & Shoutoutz to Whitefire for hosting our forum, forum.exetools.com, tech-
arena.com, and the ARTeam

Press any key to continue . . . _

```

حال برنامه را درون Olly باز کنید و رشته ها را جستجو کنید:

Address	Disassembly	Text string
0040100C	MOV BH, BL	<Initial CPU selection>
00401479	PUSH CrackmeC.0040E140	ASCII "bad cast"
00402467	PUSH CrackmeC.0040E2B8	ASCII "77777777"
00402485	PUSH CrackmeC.0040E29C	ASCII "Welcome to AR Crackme #1"
004024B0	PUSH CrackmeC.0040E278	ASCII "Please enter the correct serial: "
00402523	PUSH CrackmeC.0040E268	ASCII "Checking serial"
004025DB	PUSH CrackmeC.0040E248	ASCII "I'm sorry but that is wrong!"
00402640	PUSH CrackmeC.0040E234	ASCII "Try again? Y/N: "
004026CA	PUSH CrackmeC.0040E1FC	ASCII "Visit http://cracking.accessroot.com/ for everything AR"
0040272F	PUSH CrackmeC.0040E1E4	ASCII "Long live the ARTeam!"
00402790	PUSH CrackmeC.0040E178	ASCII "Greetz & Shoutoutz to Whitefire for hosting our forum, forum.exetools.com, tech-
00402842	PUSH CrackmeC.0040E170	ASCII "arena.com, and the ARTeam"
004028A0	PUSH CrackmeC.0040E14C	ASCII "pause"
0040290E	PUSH CrackmeC.0040E1FC	ASCII "Congratulations that is correct!"
00402AB2	ASCII "h<.A".0	ASCII "Visit http://cracking.accessroot.com/ for everything AR"
00402AC7	PUSH CrackmeC.0040E2AB2	ASCII "h<.A"
00402FCE	PUSH CrackmeC.0040E330	ASCII "ios_base::badbit set"
00403002	PUSH CrackmeC.0040E318	ASCII "ios_base::failbit set"
00403018	PUSH CrackmeC.0040E300	ASCII "ios_base::eofbit set"
00403200	PUSH CrackmeC.0040E36C	ASCII "invalid string position"
0040329B	PUSH CrackmeC.0040E384	ASCII "string too long"
00403A32	PUSH CrackmeC.0040E140	ASCII "bad cast"
004043CD	MOV DWORD PTR [EBP-4], CrackmeC.0040E484	ASCII "bad allocation"
00404CE3	MOV EAX, CrackmeC.0040E4A0	ASCII "Unknown exception"
00404DCF	PUSH CrackmeC.0040E500	ASCII "COMSPEC"
00404E00	MOV DWORD PTR [EBP-C], CrackmeC.0040E4FC	ASCII "c"
00404E3A	MOV DWORD PTR [EBP-10], CrackmeC.0040E4F4	ASCII "command.com"
00404E43	MOV DWORD PTR [EBP-10], CrackmeC.0040E4E4	ASCII "cmd.exe"
00405759	MOV DWORD PTR [412428], CrackmeC.0040E654	UNICODE " " <<<<< "H"
00405DAE	PUSH CrackmeC.0040EC18	ASCII " " <<<<< "H"
00406219	PUSH CrackmeC.0040E058	UNICODE " " h<<<< "H"
00406347	PUSH CrackmeC.0040EC2C	ASCII " " <<<<< "H"
0040822D	PUSH CrackmeC.0040ECA0	ASCII "PATH"
00408349	PUSH CrackmeC.0040ECB8	ASCII "mscoree.dll"
00408358	PUSH CrackmeC.0040ECA8	ASCII "CorExitProcess"
00408588	PUSH CrackmeC.0040ECE4	ASCII " " <<<<< "H"
00408772	MOV EDI, CrackmeC.0040EEB4	ASCII " " <<<<< "H"
00408777	MOV DWORD PTR [EBP-128], CrackmeC.0040EEB4	ASCII "Unknown security failure detected!"
00408808	MOV EDI, CrackmeC.0040EDE0	ASCII "A security error of unknown cause has been detected which has corrupted the prog
0040880D	MOV DWORD PTR [EBP-128], CrackmeC.0040EDE0	ASCII "Buffer overrun detected!"
00408836	PUSH CrackmeC.0040ED24	ASCII "A buffer overrun has been detected which has corrupted the program's internal st
00408877	PUSH CrackmeC.0040ED20	ASCII "<program name unknown>"
004088A7	MOV EDI, CrackmeC.0040ED1C	ASCII " " <<<<< "H"
004088B3	PUSH CrackmeC.0040ED10	ASCII " " <<<<< "H"
004088DD	PUSH CrackmeC.0040ECE8	ASCII "Program: "
00408AD7	PUSH CrackmeC.0040ED24	ASCII "Microsoft Visual C++ Runtime Library"
00408B0A	PUSH CrackmeC.0040ED20	ASCII "<program name unknown>"
00408B3E	PUSH CrackmeC.0040F288	ASCII " " <<<<< "H"
00408B50	PUSH CrackmeC.0040ED1C	ASCII "Runtime Error! Program: "
00408B6C	PUSH CrackmeC.0040ECE8	ASCII " " <<<<< "H"
0040A568	PUSH CrackmeC.004100E8	ASCII "Microsoft Visual C++ Runtime Library"
0040A579	PUSH CrackmeC.004100E4	ASCII "ACP"
0040A653	ADD EAX, CrackmeC.0040F30C	ASCII "OCP"
0040A673	ADD EAX, CrackmeC.0040F304	ASCII "1252"
0040A67D	ADD EAX, CrackmeC.0040F300	ASCII "850"
0040A687	ADD EAX, CrackmeC.0040F2F8	ASCII "ESP"
0040A691	ADD EAX, CrackmeC.0040F2EC	ASCII "040a"
0040AD6E	PUSH CrackmeC.004100EC	ASCII "Norwegian-Nynorsk"
0040C216	PUSH CrackmeC.00410164	ASCII "user32.dll"
0040C231	PUSH CrackmeC.00410158	ASCII "MessageBoxA"
0040C242	PUSH CrackmeC.00410148	ASCII "GetActiveWindow"
0040C244	PUSH CrackmeC.00410134	ASCII "GetLastActivePopup"
0040C265	PUSH CrackmeC.00410118	ASCII "GetUserObjectInformationA"
0040C276	PUSH CrackmeC.00410100	ASCII "GetProcessWindowStation"

تا اینجا همه چی خوب بوده؟ 😊 بسیار خوب بروی پیام هایلایت شده کلیک کنید تا ممل نمایش آن برویم :

00402580	74 05	JE SHORT CrackmeC.004025C4	
0040258F	1BC0	SBB EAX,EAX	kernel32.BaseThreadInitThunk
004025C1	8B08 FF	SBB EAX,-1	
004025C4	3BC3	CMP EAX,EBX	
004025C6	75 13	JNZ SHORT CrackmeC.004025D8	
004025C8	3BD5	CMP EDX,EBP	
004025CA	72 0F	JB SHORT CrackmeC.004025D8	
004025CC	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025CE	3BD5	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackmeC.004028AA	
004025D8	68 48E24000	PUSH CrackmeC.0040E248	ASCII "I'm sorry but that is wro
004025E0	68 A02F4100	PUSH CrackmeC.00412FA8	
004025E5	E8 A6F4FFFF	CALL CrackmeC.00401A98	
004025EA	83C4 08	ADD ESP,8	
004025ED	8BF0	MOV ESI,EAX	kernel32.BaseThreadInitThunk
004025EF	6A 0A	PUSH 0A	
004025F1	8BCE	MOV ECX,ESI	
004025F3	E8 D8F8FFFF	CALL CrackmeC.00401ED0	
004025F8	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
004025FA	8B51 04	MOV EDX,DWORD PTR DS:[ECX+4]	
004025FD	8A4C32 08	MOV CL,BYTE PTR DS:[EDX+ESI+8]	
00402601	8D0432	LEA EAX,DWORD PTR DS:[EDX+ESI]	
00402604	33FF	XOR EDI,EDI	
00402606	F6C1 06	TEST CL,6	
00402609	75 14	JNZ SHORT CrackmeC.0040261F	
0040260B	8B40 28	MOV EAX,DWORD PTR DS:[EAX+28]	
0040260E	8B10	MOV EDX,DWORD PTR DS:[EAX]	
00402610	8BC8	MOV ECX,EAX	kernel32.BaseThreadInitThunk
00402612	FF52 2C	CALL DWORD PTR DS:[EDX+2C]	
00402615	83F8 FF	CMP EAX,-1	
00402618	75 05	JNZ SHORT CrackmeC.0040261F	
0040261A	BF 04000000	MOV EDI,4	
0040261E	9906	MOV EDI,DWORD PTR DS:[ESI]	

همانطور که میبینید پرشی از آدرس 4025C6 به این قسمت صورت میگیرد(با فلش قرمز رنگ مشخص شده)، و همینطور در خط 4025D5 یک دستورالعمل JE داریم، بیائید ببینیم به کدام قسمت میرود:

004025C6	3BD5	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackmeC.004028AA	
004025D8	68 48E24000	PUSH CrackmeC.0040E248	ASCII "I'm sorry but that is wro
004025E0	68 A02F4100	PUSH CrackmeC.00412FA8	
004025E5	E8 A6F4FFFF	CALL CrackmeC.00401A98	
004025EA	83C4 08	ADD ESP,8	
004025ED	8BF0	MOV ESI,EAX	kernel32.BaseThreadInitThunk
004025EF	6A 0A	PUSH 0A	
004025F1	8BCE	MOV ECX,ESI	
004025F3	E8 D8F8FFFF	CALL CrackmeC.00401ED0	
004025F8	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
004025FA	8B51 04	MOV EDX,DWORD PTR DS:[ECX+4]	
004025FD	8A4C32 08	MOV CL,BYTE PTR DS:[EDX+ESI+8]	
00402601	8D0432	LEA EAX,DWORD PTR DS:[EDX+ESI]	
00402604	33FF	XOR EDI,EDI	
00402606	F6C1 06	TEST CL,6	
00402609	75 14	JNZ SHORT CrackmeC.0040261F	
0040260B	8B40 28	MOV EAX,DWORD PTR DS:[EAX+28]	
0040260E	8B10	MOV EDX,DWORD PTR DS:[EAX]	
00402610	8BC8	MOV ECX,EAX	kernel32.BaseThreadInitThunk
00402612	FF52 2C	CALL DWORD PTR DS:[EDX+2C]	
00402615	83F8 FF	CMP EAX,-1	
00402618	75 05	JNZ SHORT CrackmeC.0040261F	
0040261A	BF 04000000	MOV EDI,4	
0040261E	8B06	MOV EAX,DWORD PTR DS:[ESI]	
00402621	8B48 04	MOV ECX,DWORD PTR DS:[EAX+4]	
00402624	8BCE	ADD ECX,ESI	
00402626	3BF8	CMP EDI,EBX	
00402628	74 16	JE SHORT CrackmeC.00402640	
0040262A	8B41 08	MOV EAX,DWORD PTR DS:[ECX+8]	
0040262D	8B51 28	MOV EDX,DWORD PTR DS:[ECX+28]	
00402630	0BC7	OR EAX,EDI	
00402632	3BD5	CMP EDX,EBP	

پرش نسبتاً طولانی است، همین طور بیایید پائین تر:

00402881	72 0D	JB SHORT CrackmeC.00402898	
00402883	8B5424 24	MOV EDX,DWORD PTR SS:[ESP+24]	CrackmeC.<ModuleEntryPoint>
00402887	52	PUSH EDX	
00402888	E8 C81E0000	CALL CrackmeC.0040474D	
0040288D	83C4 04	ADD ESP,4	
00402890	8B4C24 48	MOV ECX,DWORD PTR SS:[ESP+48]	
00402894	64:890D 00000000	MOV DWORD PTR FS:[0],ECX	ntdll.77D2E115
00402898	8B4C24 3C	MOV ECX,DWORD PTR SS:[ESP+3C]	kernel32.BaseThreadInitThunk
0040289F	33C0	XOR EAX,EAX	
004028A1	E8 C8260000	CALL CrackmeC.00404F66	
004028A6	83C4 4C	ADD ESP,4C	
004028A9	C3	RETN	
004028AA	68 4CE14000	PUSH CrackmeC.0040E14C	ASCII "Congratulations that is correc
004028AF	68 A02F4100	PUSH CrackmeC.00412FA8	
004028B4	E8 D7F1FFFF	CALL CrackmeC.00401A98	
004028B9	83C4 08	ADD ESP,8	
004028BC	8BF0	MOV ESI,EAX	kernel32.BaseThreadInitThunk
004028BE	6A 0A	PUSH 0A	
004028C0	8BCE	MOV ECX,ESI	
004028C2	E8 D9F6FFFF	CALL CrackmeC.00401ED0	
004028C7	8B06	MOV EAX,DWORD PTR DS:[ESI]	
004028C9	8B48 04	MOV ECX,DWORD PTR DS:[EAX+4]	
004028CC	8D0431	LEA EAX,DWORD PTR DS:[ECX+ESI]	
004028CF	8A48 08	MOV CL,BYTE PTR DS:[EAX+8]	
004028D2	33FF	XOR EDI,EDI	
004028D4	F6C1 06	TEST CL,6	
004028D7	75 14	JNZ SHORT CrackmeC.004028ED	
004028D9	8B40 28	MOV EAX,DWORD PTR DS:[EAX+28]	

اها(☺). به نظر اينجا بايد فرود بياييم ،ولى بياييد برگرديم :

00402571	8B41 08	MOV EAX,DWORD PTR DS:[ECX+8]	
00402574	8B51 28	MOV EDX,DWORD PTR DS:[ECX+28]	
00402577	8BC7	OR EAX,EDI	
00402579	3BD3	CMP EDX,EBX	
0040257B	75 03	JNZ SHORT CrackneC.00402580	
0040257D	83C8 04	OR EAX,4	
00402580	53	PUSH EBX	
00402581	50	PUSH EAX	kernel32.BaseThreadInitThunk
00402582	E8 100A0000	CALL CrackneC.00402F97	
00402587	8B7C24 2C 10	CMP DWORD PTR SS:[ESP+2C],10	
0040258C	8B7C24 18	MOV EDI,DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackneC.00402596	
00402592	8D7C24 18	LEA EDI,DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX,DWORD PTR SS:[ESP+44]	
0040259A	3BD3	CMP EDX,EBX	
0040259C	9B6C24 28	JE SHORT CrackneC.004025C8	
004025A0	74 25	CMPEB EDI,EBP	
004025A2	8BD5	MOV ECX,EDX	CrackneC.<ModuleEntryPoint>
004025A4	8BCA	MOV ECX,EDX	
004025A6	72 02	JB SHORT CrackneC.004025AA	
004025A8	8BCD	MOV ECX,EBP	
004025AA	8B7C24 48 10	CMP DWORD PTR SS:[ESP+48],10	
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+34]	
004025B3	73 04	JNB SHORT CrackneC.004025B9	
004025B5	8D7424 34	LEA ESI,DWORD PTR SS:[ESP+34]	
004025B9	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025BB	F3A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]	
004025BD	74 05	JE SHORT CrackneC.004025C4	
004025BF	1BC0	SBB EAX,EAX	kernel32.BaseThreadInitThunk
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CMP EAX,EBX	
004025C6	75 13	JNZ SHORT CrackneC.004025D8	
004025C8	3BD5	CMP EDX,EBP	
004025CA	72 0F	JB SHORT CrackneC.004025D8	
004025CC	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025CE	3BD5	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackneC.004028AA	
004025D8	68 48E24000	PUSH CrackneC.0040E248	ASCII "I'm sorry but that is wrong!"
004025E0	68 A02F4100	PUSH CrackneC.00412FA0	
004025E5	E8 A6F4FFFF	CALL CrackneC.00401A90	
004025EA	83C4 08	ADD ESP,8	
004025ED	8BF0	MOV ESI,EAX	kernel32.BaseThreadInitThunk
004025EF	6A 0A	PUSH 0A	
004025F1	8BC6	MOV ECX,ESI	
004025F3	E8 D8F8FFFF	CALL CrackneC.00401ED0	
004025F8	8BE	MOV ECX,DWORD PTR DS:[ESI]	
004025FA	8B51 04	MOV EDX,DWORD PTR DS:[ECX+4]	
004025FD	8A4C32 08	MOV CL,BYTE PTR DS:[EDX+ESI+8]	
00402601	8D0432	LEA EAX,DWORD PTR DS:[EDX+ESI]	
00402604	33FF	XOR EDI,EDI	
00402606	F6C1 06	TEST CL,6	
00402609	75 14	JNZ SHORT CrackneC.0040261F	
0040260B	8B40 28	MOV EAX,DWORD PTR DS:[EAX+28]	
0040260E	8B10	MOV EDX,DWORD PTR DS:[EAX]	

مشفص شد كه اين پرس از آدرس 4025D5 به قسمت Good Boy ميرود،كمى بالاتر در ادرس 4025CA يك دستور JB (پرشى) ميپيديد كه آن هم به سمت Bad Boy ميرود :

004025BF	1BC0	SBB EAX,EAX	kernel32.BaseThreadInitThunk
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CMP EAX,EBX	
004025C6	75 13	JNZ SHORT CrackneC.004025D8	
004025C8	3BD5	CMP EDX,EBP	
004025CA	72 0F	JB SHORT CrackneC.004025D8	
004025CC	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025CE	3BD5	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackneC.004028AA	
004025D8	68 48E24000	PUSH CrackneC.0040E248	ASCII "I'm sorry but that is wrong!"
004025E0	68 A02F4100	PUSH CrackneC.00412FA0	
004025E5	E8 A6F4FFFF	CALL CrackneC.00401A90	
004025EA	83C4 08	ADD ESP,8	

و همچنين در آدرس 4025C6 يك پرس به سمت Bad Boy داريم :

004025BB	F3A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]	
004025BD	74 05	JE SHORT CrackneC.004025C4	
004025BF	1BC0	SBB EAX,EAX	kernel32.BaseThreadInitThunk
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CMP EAX,EBX	
004025C6	75 13	JNZ SHORT CrackneC.004025D8	
004025C8	3BD5	CMP EDX,EBP	
004025CA	72 0F	JB SHORT CrackneC.004025D8	
004025CC	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025CE	3BD5	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackneC.004028AA	
004025D8	68 48E24000	PUSH CrackneC.0040E248	ASCII "I'm sorry but that is wrong!"
004025E0	68 A02F4100	PUSH CrackneC.00412FA0	
004025E5	E8 A6F4FFFF	CALL CrackneC.00401A90	
004025EA	83C4 08	ADD ESP,8	

همینطور که مشخص است تنها پرشی که به سمت Good Boy میرود در آدرس 4025D5 است، فب سناریو مشخص شد، تمام پرش ها به سمت Bad Boy نباید ارجاع شود، تا به آدرس 4025D5 و از آنجا مستقیماً به Good Boy برویم، اگر کمی بالاتر را نگاه کنیم درست در آدرس 402582 ما اولین دستورالعمل call / compare را میبینیم :

00402579	3B03	CMP EDX,EBX	
0040257B	75 03	JNZ SHORT CrackmeC.00402580	
0040257D	83C8 04	OR EAX,4	
00402580	53	PUSH EBX	
00402581	50	PUSH EAX	kernel32.BaseThreadInitThunk
00402582	E8 100A0000	CALL CrackmeC.00402F97	
00402587	837C24 2C 10	CMP DWORD PTR SS:[ESP+2C],10	
0040258C	8B7C24 18	MOV EDI,DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackmeC.00402596	
00402592	8D7C24 18	LEA EDI,DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX,DWORD PTR SS:[ESP+44]	
0040259A	3B03	CMP EDX,EBX	
0040259C	8B6C24 28	MOV EBP,DWORD PTR SS:[ESP+28]	
004025A0	74 26	JE SHORT CrackmeC.004025C8	
004025A2	3B05	CMP EDX,EBP	CrackmeC.<ModuleEntryPoint>
004025A4	8BCA	MOV ECX,EDX	
004025A6	72 02	JB SHORT CrackmeC.004025AA	
004025A8	8BC0	MOV ECX,EBP	
004025AA	837C24 48 10	CMP DWORD PTR SS:[ESP+48],10	
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+34]	
004025B3	73 04	JNB SHORT CrackmeC.004025B9	
004025B5	8D7424 34	LEA ESI,DWORD PTR SS:[ESP+34]	
004025B9	33C0	XOR EAX,EAX	kernel32.BaseThreadInitThunk
004025BB	F3A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:	

اگر باز کمی به بالا نگاه کنیم یک پرش در آدرس 40256F میبینیم که از این فراخوانی عبور میکند :

0040255F	75 05	JNZ SHORT CrackmeC.00402566	
00402561	BF 04000000	MOV EDI,4	
00402566	8B06	MOV EAX,DWORD PTR DS:[ESI]	
00402568	8B48 04	MOV ECX,DWORD PTR DS:[EAX+4]	
0040256B	03CE	ADD ECX,ESI	
0040256D	3BF8	CMP EDI,EBX	
0040256F	74 16	JE SHORT CrackmeC.00402587	
00402571	8B41 08	MOV EAX,DWORD PTR DS:[ECX+8]	
00402574	8B51 28	MOV EDX,DWORD PTR DS:[ECX+28]	
00402577	08C7	OR EAX,EDI	
00402579	3B03	CMP EDX,EBX	
0040257B	75 03	JNZ SHORT CrackmeC.00402580	
0040257D	83C8 04	OR EAX,4	
00402580	53	PUSH EBX	
00402581	50	PUSH EAX	kernel32.BaseThreadInitThunk
00402582	E8 100A0000	CALL CrackmeC.00402F97	
00402587	837C24 2C 10	CMP DWORD PTR SS:[ESP+2C],10	
0040258C	8B7C24 18	MOV EDI,DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackmeC.00402596	
00402592	8D7C24 18	LEA EDI,DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX,DWORD PTR SS:[ESP+44]	
0040259A	3B03	CMP EDX,EBX	
0040259C	8B6C24 28	MOV EBP,DWORD PTR SS:[ESP+28]	
004025A0	74 26	JE SHORT CrackmeC.004025C8	
004025A2	3B05	CMP EDX,EBP	CrackmeC.<ModuleEntryPoint>
004025A4	8BCA	MOV ECX,EDX	

باز کمی بالاتر یک فراخوانی و مقایسه دیگر وجود دارد، حالا بروی هر دو تابع BP میگذاریم :

00402542	8B42 04	MOV EAX,DWORD PTR DS:[EDX+4]	
00402545	8A4C30 08	MOV CL,BYTE PTR DS:[EAX+ESI+8]	
00402549	03CE	ADD EAX,ESI	
0040254B	33FF	XOR EDI,EDI	
0040254D	74C1 06	TEST CL,6	
00402550	75 14	JNZ SHORT CrackmeC.00402566	
00402552	8B48 28	MOV EAX,DWORD PTR DS:[EAX+28]	
00402555	8B10	MOV EDX,DWORD PTR DS:[EAX]	
00402557	8BC8	MOV ECX,EAX	
00402559	FF52 2C	CALL DWORD PTR DS:[EDX+2C]	kernel32.BaseThreadInitThunk
0040255C	83F8 FF	CMP EAX,-1	
0040255F	75 05	JNZ SHORT CrackmeC.00402566	
00402561	BF 04000000	MOV EDI,4	
00402566	8B06	MOV EAX,DWORD PTR DS:[ESI]	
00402568	8B48 04	MOV ECX,DWORD PTR DS:[EAX+4]	
0040256B	03CE	ADD ECX,ESI	
0040256D	3BF8	CMP EDI,EBX	
0040256F	74 16	JE SHORT CrackmeC.00402587	
00402571	8B41 08	MOV EAX,DWORD PTR DS:[ECX+8]	
00402574	8B51 28	MOV EDX,DWORD PTR DS:[ECX+28]	
00402577	08C7	OR EAX,EDI	
00402579	3B03	CMP EDX,EBX	
0040257B	75 03	JNZ SHORT CrackmeC.00402580	
0040257D	83C8 04	OR EAX,4	
00402580	53	PUSH EBX	
00402581	50	PUSH EAX	kernel32.BaseThreadInitThunk
00402582	E8 100A0000	CALL CrackmeC.00402F97	
00402587	837C24 2C 10	CMP DWORD PTR SS:[ESP+2C],10	
0040258C	8B7C24 18	MOV EDI,DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackmeC.00402596	
00402592	8D7C24 18	LEA EDI,DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX,DWORD PTR SS:[ESP+44]	
0040259A	3B03	CMP EDX,EBX	
0040259C	8B6C24 28	MOV EBP,DWORD PTR SS:[ESP+28]	

حالا برنامه را درون Olly اجرا کرده و یک پسورد دلفواه وارد کرده من "12121212" را وارد میکنم و Olly بروی تابع اولی متوقف میشود:

00402543	8A4C50 06	MOV CL, BYTE PTR DS:[EAX+ESI+6]	
00402549	03C6	ADD EAX, ESI	CrackmeC.00412FA0
0040254B	33FF	XOR EDI, EDI	
0040254D	F6C1 06	TEST CL, 6	
00402550	75 14	JNZ SHORT CrackmeC.00402566	
00402552	8B40 28	MOV EAX, DWORD PTR DS:[EAX+28]	
00402555	8B10	MOV EDX, DWORD PTR DS:[EAX]	CrackmeC.0040E408
00402557	8BC8	MOV ECX, EAX	CrackmeC.00412F40
00402559	FF52 2C	CALL DWORD PTR DS:[EDX+2C]	CrackmeC.004037A1
0040255C	83F8 FF	CMPEAX, -1	
0040255F	75 05	JNZ SHORT CrackmeC.00402566	
00402561	BF 04000000	MOVEDI, 4	
00402566	8B06	MOV EAX, DWORD PTR DS:[ESI]	CrackmeC.0040E470
00402568	8B48 04	MOV ECX, DWORD PTR DS:[EAX+4]	
0040256B	03CE	ADDECX, ESI	CrackmeC.00412FA0
0040256D	3BF8	CMPEDI, EBX	
0040256F	74 16	JE SHORT CrackmeC.00402587	
00402571	8B41 08	MOV EAX, DWORD PTR DS:[ECX+8]	
00402574	8B51 28	MOV EDX, DWORD PTR DS:[ECX+28]	
00402577	0BC7	OREAX, EDI	
00402579	3B03	CMPEDX, EBX	
0040257B	75 03	JNZ SHORT CrackmeC.00402580	
0040257D	83C8 04	OREAX, 4	
00402580	53	PUSHEBX	
00402581	50	PUSHEAX	CrackmeC.00412F40
00402582	E8 100A0000	CALL CrackmeC.00402F97	
00402587	837C24 2C 10	CMPDWORD PTR SS:[ESP+2C], 10	
0040258C	8B7C24 18	MOVEDI, DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackmeC.00402596	
00402592	8D7C24 18	LEAEDI, DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX, DWORD PTR SS:[ESP+44]	
0040259A	3B03	CMPEDX, EBX	
0040259C	8B6C24 28	MOVEBP, DWORD PTR SS:[ESP+28]	

مال کلید F8 را بفشارید تا به آدرس 40256F برسید، همانطور که میبینید این پرش از فراخوانی و مقایسه عبور می کند :

0040254D	F6C1 06	TEST CL, 6	
00402550	75 14	JNZ SHORT CrackmeC.00402566	
00402552	8B40 28	MOV EAX, DWORD PTR DS:[EAX+28]	CrackmeC.00404D25
00402555	8B10	MOV EDX, DWORD PTR DS:[EAX]	
00402557	8BC8	MOV ECX, EAX	CrackmeC.0040E470
00402559	FF52 2C	CALL DWORD PTR DS:[EDX+2C]	CrackmeC.004037A1
Jumps past our call			
0040255C	83F8 FF	CMPEAX, -1	
0040255F	75 05	JNZ SHORT CrackmeC.00402566	
00402561	BF 04000000	MOVEDI, 4	
00402566	8B06	MOV EAX, DWORD PTR DS:[ESI]	CrackmeC.0040E470
00402568	8B48 04	MOV ECX, DWORD PTR DS:[EAX+4]	
0040256B	03CE	ADDECX, ESI	CrackmeC.00412FA0
0040256D	3BF8	CMPEDI, EBX	
0040256F	74 16	JE SHORT CrackmeC.00402587	
00402571	8B41 08	MOV EAX, DWORD PTR DS:[ECX+8]	
00402574	8B51 28	MOV EDX, DWORD PTR DS:[ECX+28]	CrackmeC.00412F40
00402577	0BC7	OREAX, EDI	
00402579	3B03	CMPEDX, EBX	
0040257B	75 03	JNZ SHORT CrackmeC.00402580	
0040257D	83C8 04	OREAX, 4	
00402580	53	PUSHEBX	
00402581	50	PUSHEAX	CrackmeC.0040E470
00402582	E8 100A0000	CALL CrackmeC.00402F97	
00402587	837C24 2C 10	CMPDWORD PTR SS:[ESP+2C], 10	
0040258C	8B7C24 18	MOVEDI, DWORD PTR SS:[ESP+18]	
00402590	73 04	JNB SHORT CrackmeC.00402596	
00402592	8D7C24 18	LEAEDI, DWORD PTR SS:[ESP+18]	
00402596	8B5424 44	MOV EDX, DWORD PTR SS:[ESP+44]	
0040259A	3B03	CMPEDX, EBX	
0040259C	8B6C24 28	MOVEBP, DWORD PTR SS:[ESP+28]	

کلید F8 را آنقدر بفشارید تا به آدرس 4025C6 برسید، این همان آدرسی است که ما را به سمت Bad Boy میبرد:

004025A6	72 02	JB SHORT CrackmeC.004025AA	
004025A8	8BCD	MOVECX, EBP	
004025AA	837C24 48 10	CMPDWORD PTR SS:[ESP+48], 10	
004025AF	8B7424 34	MOVEDI, DWORD PTR SS:[ESP+34]	
004025B3	73 04	JNB SHORT CrackmeC.004025B9	
004025B5	8D7424 34	LEAEDI, DWORD PTR SS:[ESP+34]	
004025B9	33C8	XOREAX, EAX	
004025BB	F3A6	REPE CMPS BYTE PTR ES:[EDI], BYTE PTR DS:[EDI]	
004025BD	74 05	JE SHORT CrackmeC.004025C4	
004025BF	1BC9	SBB EAX, EAX	
004025C1	83D8 FF	SBB EAX, -1	
004025C4	3BC3	CMPEAX, EBX	
004025C6	75 13	JNZ SHORT CrackmeC.004025DB	
004025C8	3B05	CMPEDX, EBP	
004025CA	72 0F	JB SHORT CrackmeC.004025DB	
004025CC	33C8	XOREAX, EAX	
004025CE	3B05	CMPEDX, EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMPEAX, EBX	
004025D5	0F84 CF020000	JE CrackmeC.004028AA	
004025D8	68 48E24000	PUSH CrackmeC.0040E240	
004025DB	68 A02F4100	PUSH CrackmeC.00412FA0	
004025E5	E8 A6F4FFFF	CALL CrackmeC.00401A90	
004025EA	83C4 08	ADD ESP, 8	
004025ED	8BF0	MOVESI, EAX	
004025EF	6A 0A	PUSH 0A	
004025F1	8BC0	MOVECX, ESI	
004025F3	E8 D8F8FFFF	CALL CrackmeC.00401ED0	
004025F8	8B0E	MOVECX, DWORD PTR DS:[ESI]	

مالا با تغییر پریم Z=1 این پرش را غیر فعال کنید :

C	0	ES	0023
P	1	CS	001B
A	0	SS	0023
Z	1	DS	0023
S	1	FS	003B
T	0	GS	0000
D	0		
O	0	LastErr	

حالا کلید F8 را بفشارید، میبینید که مستقیم به سمت Good Boy ارجاع داده شدیم :

004025C0	33C0	XOR EAX,EAX	
004025C1	3B05	CMP EDX,EBP	
004025C2	0F95C0	SETNE AL	
004025C3	3BC3	CMF EAX,EBX	
004025C4	0F84 CF020000	JE CrackmeC.004028AA	
004025C5	68 48E24000	PUSH CrackmeC.0040E248	
004025C6	68 A02F4100	PUSH CrackmeC.00412FA0	
004025C7	E8 A6F4FFFF	CALL CrackmeC.00401A90	
004025C8	93C0 00	ADD ESP,0	
004025C9	8BF0	MOV ESI,EAX	
004025CA	6A 0A	PUSH 0A	
004025CB	8BCF	MOV ECX,ESI	
004025CC	E8 D8F8FFFF	CALL CrackmeC.00401ED0	
004025CD	8B0E	MOV ECX,DWORD PTR DS:[ESI]	
004025CE	8B51 04	MOV EDX,DWORD PTR DS:[ECX+4]	
004025CF	8A4C32 08	MOV CL,BYTE PTR DS:[EDX+ESI+8]	
004025D0	8D0432	LEA ECX,DWORD PTR DS:[ECX+ESI]	
004025D1	33FF		
004025D2	F6C1 06	TEST EBX,EBX	
004025D3	75 14	JNZ CrackmeC.0040261F	
004025D4	8B40 28	MOV EAX,DWORD PTR DS:[EAX+28]	
004025D5	8B10	MOV EDX,DWORD PTR DS:[EAX]	
004025D6	8BC8	MOV ECX,EAX	
004025D7	FF52 2C	CALL DWORD PTR DS:[ECX+2C]	
004025D8	83F8 FF	CMF EAX,-1	
004025D9	75 05	JNZ SHORT CrackmeC.0040261F	
004025DA	8F 04000000	MOV EDI,4	
004025DB	8B06	MOV EAX,DWORD PTR DS:[ESI]	
004025DC	8B48 04	MOV ECX,DWORD PTR DS:[EAX+4]	
004025DD	83C5	CMF ESI	

Jump to good boy

حالا به برنامه نگاه کنید :

```

C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
12121212
Checking serial
Congratulations that is correct!
Visit http://cracking.accessroot.com/ for everything AR
Long live the ARTeam!
Greetz & Shoutoutz to Whitefire for hosting our forum, forum.exetools.com, tech-arena.com, and the ARTeam

Press any key to continue . . .

```

حالا طبق آموزش های قبلی پرش قبلی را با دستور Nop تغییر و نتیجه را ثبت و برنامه را اجرا کنید، و یک پسورد وارد کنید اما اینجا طول پسورد کوتاه یا بلند باشد ☺ برنامه پیام Bad Boy را نمایش داد ☺ بیایید علت ما را بفهمیم :

مفاری عمیق تر

درواقع اگر بخواهیم از نگاه آموزش قبلی به این موضوع نگاه کنیم درواقع ما از تکنیک LAME استفاده کرده ایم بیایید بروی پرشی که ما را به Bad Boy ارجاع میدهد (که قبلا پی کردیم) متمرکز شویم و بفهمیم که چرا با وجود پی کردن باز هم با تغییر طول پسورد این پی درست عمل نمیکند ، از طرفی با درج توضیحات میتوان برنامه را مستند سازی کرد ، من از ## در ابتدای توضیحات استفاده میکنم که موقع رجوع پیدا کردنش راحت تر باشد، البته شما میتوانید هر جور که راحتید عمل کنید:

004025B9	33C0	XOR EAX,EAX	CrackmeC.00412F40
004025BA	F3A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:	CrackmeC.00412F40
004025BB	74 05	JE SHORT CrackmeC.004025C4	CrackmeC.00412F40
004025BC	1BC8	SBB EAX,EAX	
004025BD	8308 FF	SBB EAX,-1	
004025BE	3BC3	CMF EAX,EBX	
004025BF	75 13	JNZ SHORT CrackmeC.004025D8	### Jump to bad boy
004025C0	3B05	CMP EDX,EBP	CrackmeC.00412F40
004025C1	72 0F	JB SHORT CrackmeC.004025D8	
004025C2	33C0	XOR EAX,EAX	
004025C3	3B05	CMP EDX,EBP	
004025C4	0F95C0	SETNE AL	
004025C5	3BC3	CMF EAX,EBX	
004025C6	0F84 CF020000	JE CrackmeC.004028AA	
004025C7	68 48E24000	PUSH CrackmeC.0040E248	
004025C8	68 A02F4100	PUSH CrackmeC.00412FA0	
004025C9	E8 A6F4FFFF	CALL CrackmeC.00401A90	

به قسمت انتخاب شده (کادر قرمز رنگ) تصویر زیر نگاه کنید:

004025A8	837C24 48 10	CHP DWORD PTR SS:[ESP+48],10	
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+84]	
004025B2	73 04	JNG SHORT CrackneC.004025B9	
004025B5	8D7424 34	LEA ESI,DWORD PTR SS:[ESP+84]	
004025B9	33C0	XOR EAX,EAX	CrackneC.00412F40
004025BB	F3:A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:	
004025BD	74 05	JE SHORT CrackneC.004025C4	
004025BF	1BC0	SBB EAX,EAX	CrackneC.00412F40
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CHP EAX,EBX	### Jump to bad boy
004025C6	75 13	JNZ SHORT CrackneC.004025D8	
004025C8	3B05	CMPS EDI,EBP	
004025CA	72 0F	JB SHORT CrackneC.004025D8	
004025CC	33C0	XOR EAX,EAX	CrackneC.00412F40
004025CE	3B05	CMPS EDI,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMPS EAX,EBX	
004025D5	0F84 CF020000	JE CrackneC.004028AA	
004025D8	68 48E24000	PUSH CrackneC.0040E248	
004025DB	68 A02F4100	PUSH CrackneC.00412FA0	ASCII "I'm sorry but that is wrong!"

چند دستور SBB و یک دستور مقایسه میبینید که در واقع معنی فاصی برای درک کلی کد به ما نمیدهند

دستورالعمل **sbb** (subtract with borrow) مشابه دستور sub است با این تفاوت که حاصل تفریق عملوند دوم و CF از عملوند

اول را محاسبه می نماید (dest:=dest-src-CF)

اما اگر به تصویر زیر نگاه کنید همه چیز مشخص خواهد شد:

004025A2	3B05	CHP EDX,EBP	
004025A4	8BCA	MOV ECX,EDX	CrackneC.0040E408
004025A6	72 02	JB SHORT CrackneC.004025AA	
004025A8	8BCD	MOV ECX,EBP	
004025AA	837C24 48 10	CHP DWORD PTR SS:[ESP+48],10	
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+84]	
004025B3	73 04	JNG SHORT CrackneC.004025B9	
004025B5	8D7424 34	LEA ESI,DWORD PTR SS:[ESP+84]	CrackneC.00412F40
004025B9	33C0	XOR EAX,EAX	
004025BB	F3:A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:	CrackneC.00412F40
004025BD	74 05	JE SHORT CrackneC.004025C4	
004025BF	1BC0	SBB EAX,EAX	CrackneC.00412F40
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CHP EAX,EBX	### Jump to bad boy
004025C6	75 13	JNZ SHORT CrackneC.004025D8	
004025C8	3B05	CMPS EDI,EBP	
004025CA	72 0F	JB SHORT CrackneC.004025D8	
004025CC	33C0	XOR EAX,EAX	CrackneC.00412F40
004025CE	3B05	CMPS EDI,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMPS EAX,EBX	
004025D5	0F84 CF020000	JE CrackneC.004028AA	
004025D8	68 48E24000	PUSH CrackneC.0040E248	
004025DB	68 A02F4100	PUSH CrackneC.00412FA0	ASCII "I'm sorry but that is wrong!"
004025DE	E8 A6F4FFFF	CALL CrackneC.00401A90	

دستورالعمل **Lea** : این دستور آدرس موثر یک محل فاص از حافظه را درون یک ثبات همه منظوره ذخیره می کند. منظور از آدرس

موثر آدرس نهائی حافظه بعد از کلیه محاسبات آدرسی است. همچنین این دستور بروی پرچم ها تاثیری ندارد

همین طور فضا بعدی دستورالعمل **REPE CMPS** را داریم :

Intel x86 Instructions

File Edit Bookmark Options Help

Contents Index Back Print

REP/REPE/REPZ/REPNE/REPNZ—Repeat String Operation Prefix

See also

F2 AF REPNE SCAS m32 Find EAX, starting at ES:[EDI]

Description

Repeats a string instruction the number of times specified in the count register ((E)CX) or until the indicated condition of the ZF flag is no longer met. The REP (repeat), REPE (repeat while equal), REPNE (repeat while not equal), REPZ (repeat while zero), and REPNZ (repeat while not zero) mnemonics are prefixes that can be added to one of the string instructions. The REP prefix can be added to the INS, OUTS, MOVSB, LODSB, and STOS instructions. The REPE, REPNE, REPZ, and REPNZ prefixes can be added to the CMPS and SCAS instructions. (The REPZ and REPNE prefixes are synonymous forms of the REPE and REPNE prefixes, respectively.) The behavior of the REP prefix is undefined when used with non-string instructions.

The REP prefixes apply only to one string instruction at a time. To repeat a block of instructions, use the LOOP instruction or another looping construct.

All of these repeat prefixes cause the associated instruction to be repeated until the count in register (E)CX is decremented to 0 (see the following table). (If the current address-size attribute is 32, register ECX is used as a counter, and if the address-size attribute is 16, the CX register is used.) The REPE, REPNE, REPZ, and REPNZ prefixes also check the state of the ZF flag after each iteration and terminate the repeat loop if the ZF flag is not in the specified state. When both termination conditions are tested, the cause of a repeat termination can be determined either by testing the (E)CX register with a JECXZ instruction or by testing the ZF flag with a JZ, JNZ, and JNE instruction.

Repeat Conditions

Repeat Prefix	Termination Condition 1	Termination Condition 2
REP	ECX=0	None
REPE/REPZ	ECX=0	ZF=0
REPNE/REPNZ	ECX=0	ZF=1

When the REPE/REPZ and REPNE/REPNZ prefixes are used, the ZF flag does not require initialization because both the CMPS and SCAS instructions affect the ZF flag according to the results of the comparisons they make.

A repeating string operation can be suspended by an exception or interrupt. When this happens, the state of the registers is preserved to allow the string operation to be resumed upon a return from the exception or interrupt handler. The source and destination registers point to the next string elements to be operated on, the EIP register points to the string instruction, and the ECX register has the value it held following the last successful iteration of the instruction. This mechanism allows long string operations to proceed without affecting the interrupt response time of the system.

When a fault occurs during the execution of a CMPS or SCAS instruction that is prefixed with REPE or REPNE, the EFLAGS value is restored to the state prior to the execution of the instruction. Since the SCAS and CMPS instructions do not use EFLAGS as an input, the processor can resume the instruction after the page fault handler.

Use the REP INS and REP OUTS instructions with caution. Not all I/O ports can handle the rate at which these instructions execute.

A REP STOS instruction is the fastest way to initialize a large block of memory.

دستور بزرگی است نه؟ 😊 اما کارکرد بسیار جالب و در عین حال ساده ای دارد، دستور REPXX در کل به معنای تکرار است تا موقعی که اتفاق خاصی رخ دهد، مثلاً تکرار یک کاری تا موقعی که ECX=0 شود، دستور بعد از آن CMPS نام دارد، که به معنای مقایسه است، Cmps دو عنصر رشته‌ای را تفریق کرده و فلگ‌ها را بر اساس آنها تنظیم می‌کند؛ (هیچکدام از عملوندها تغییر داده نمی‌شوند).

انواع فورمهای REPXX :

۱- repne: «تا زمانی که حالت مساوی وجود ندارد تکرار کن»

۲- repnz: «تا زمانی که صفر نیست تکرار کن»

این پیشوندها تا زمانی که CX صفر نیست، دستورالعمل اولیه را تکرار می‌کنند.

پیشوندهای repe و repz تکرار را تا زمانی که ZF=1 است تکرار می‌کنند.

پیشوندهای repnz و repne تکرار را تا زمانی که ZF=0 است تکرار می‌کنند.

پس معنی فضا 4025BB اینطور است:

دو مقدار حافظه را به صورت بایت به بایت تا انتها با هم مقایسه کن

حالا به فضا 4025B5 نگاه کنید، بیا بیدر این فضا یک BP بگذاریم و برنامه ری استارت کنیم و یک پسورد دلفواه وارد کنیم :

004025A6	72 02	JB SHORT CrackmeC.004025AA	
004025A8	88CD	MOV ECX,EBP	
004025AA	837C24 48 10	CMPL DWORD PTR SS:[ESP+48],10	
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+34]	
004025B3	73 04	JNB SHORT CrackmeC.004025B9	
004025B8	8B7424 34	LEA ESI,DWORD PTR SS:[ESP+34]	
004025B9	33C0	XOR EAX,EAX	CrackmeC.0040E470
004025BB	F3:A6	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:	
004025BD	74 05	JE SHORT CrackmeC.004025C4	CrackmeC.0040E470
004025BF	1BC0	SBB EAX,EAX	
004025C1	83D8 FF	SBB EAX,-1	
004025C4	3BC3	CMPL EAX,EBX	
004025C6	75 13	JNE SHORT CrackmeC.004025D8	
004025C8	3BD5	CMPL EDI,EBP	
004025CA	72 0F	JB SHORT CrackmeC.004025D8	
004025CC	33C0	XOR EAX,EAX	CrackmeC.0040E470
004025CE	3BD5	CMPL EDI,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMPL EAX,EBX	
004025D5	0F84 CF020000	JE CrackmeC.004025AA	
004025D8	68 48E24000	PUSH CrackmeC.0040E248	ASCII "I'm sorry but that is wrong!"
004025E0	68 A02F4100	PUSH CrackmeC.00412FA8	
004025E2	F8 0F000000	LOCK	

می بینید که Olly بروی فضا مورد نظر متوقف شده :

LEA ESI, DWORD PTR SS:[ESP+34]

این فضا آدرس موثر واقع در ثبات ESI را درون پشته قرار می‌دهد، که SS: به معنی پشته ، و [ESP+34] به معنای موقعیت پشته است، اگر به فلش آبی رنگ در پنجره ی Info (بالای پنجره ی دامپ) نگاه کنید می بینید که [SS:[ESP+34]] با آدرس 12FE88 برابر است ، و محتوای این آدرس پسورد وارد شده ی ما است، حال یکبار کلید F8 را بفشارید و به مقدار ثبات ESI نگاه کنید (که اکنون بروی پشته است) :

Registers (FPU)		
EAX	0040E470	CrackmeC.0040E470
ECX	00000008	
EDX	00000008	
EBX	00000000	
ESP	0012FE54	
EBP	00000008	
ESI	0012FE88	ASCII "12121212"
EDI	0012FE6C	ASCII "77777777"
EIP	004025B9	CrackmeC.004025B9
C 1	ES 0023	32bit 0(FFFFFFFF)
P 1	CS 001B	32bit 0(FFFFFFFF)
D 0	FS 0000	32bit 0(FFFFFFFF)

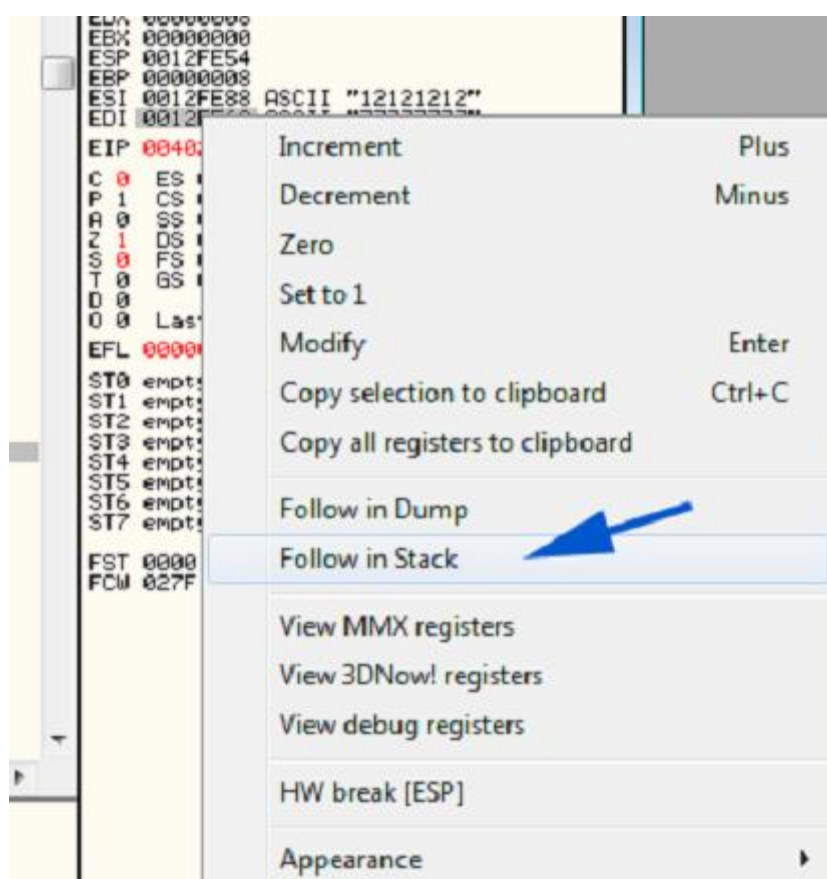
و در این آدرس ماوی پسورد ما است، دستور العمل بعدی مقدار EAX را برابر 0 میکند ، و در خط بعدی ما دستور العمل REPE را داریم :

```

004025A8 8BCD          MOV ECX,EBP
004025AA 837C24 48 10  CMP DWORD PTR SS:[ESP+48],10
004025AF 8B7424 34     MOV ESI,DWORD PTR SS:[ESP+34]
004025B3 73 04        JNB SHORT CrackmeC.004025B9
004025B5 807424 34     LEA ESI,DWORD PTR SS:[ESP+34]
004025B9 33C0         XOR EAX,EAX
004025BB F9A6         REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS:
004025BD 74 05        JE SHORT CrackmeC.004025C4
004025BF 1BC0         SBB EAX,EAX
004025C1 83DB FF     SBB EAX,-1
004025C4 3BC3         CMP EAX,EBX
004025C6 75 13        JNZ SHORT CrackmeC.004025D8
004025C8 3BD5         CMP EDX,EBP
004025CA 72 0F        JB SHORT CrackmeC.004025D8
004025CC 33C0         XOR EAX,EAX
004025CE 3BD5         CMP EDX,EBP
004025D0 0F95C0      SETNE AL
004025D3 3BC3         CMP EAX,EBX
004025D5 0F84 CF020000 JE CrackmeC.004028AA
004025D8 68 48E24000 PUSH CrackmeC.0040E248
004025DE 68 A02F4100 PUSH CrackmeC.00412FA0
004025E5 F8 06F4FFFF CALL CrackmeC.00401090
  
```

ECX=00000008 (decimal 8.)
 DS:[ESI]=stack [0012FE88]=31 ('1')
 ES:[EDI]=stack [0012FE6C]=37 ('7')

همانطور که قبلا توضیح داده شد این دستور مقدار آدرس ذخیره شده در ESI را با مقدار آدرس حافظه ی EDI مقایسه میکند به صورت بایت به بایت (کاراکتر به کاراکتر) ، مقدار ثابت ECX یک واحد کاهش میابد ، و عملیات مقایسه از EDI و ESI ادامه میابد، و این حلقه تا موقعی که ECX=0 باشد ادامه میابد، اگر به تصویر زیر نگاه کنید میبینید که مقدار ECX برابر با 8 است (طول رمز وارد شده ی ما) و هر کاراکتر از پسورد ما با مقدار EDI مقایسه میشود ، اما صبر کنید!! پسورد ما با چه چیزی مقایسه میشود؟ اگر به ثابت EDI نگاه کنید آدرس پشته را میبینید که ماوی کد اسکی "7" است ، بروی ثابت EDI راست کلیک کنید و : "Follow in stack":



پنجره ی پشت به آدرس مورد نظر رجوع میکند: 12FE6C و همانطور که مشاهده میکنید این آدرس حاوی رشته ی "37" است ،اگر به نمودار کد های اسکی بپردازید میبینید که برابر با عدد "7" است :

0012FE6C	37373737	
0012FE70	37373737	
0012FE74	00412000	CrackmeC.00412000
0012FE78	0012FF88	
0012FE7C	00000008	
0012FE80	0000000F	
0012FE84	00000000	
0012FE88	32313231	
0012FE8C	32313231	
0012FE90	00406400	CrackmeC.00406400
0012FE94	003A0680	
0012FE98	00000008	
0012FE9C	0000000F	

بله درست است رمز مورد نظر "77777777" است ،و دلیل مشکل طول پسورد هم در آدرس 4025C8 کاملاً مشخص است به این صورت که ثبات EDX حاوی طول پسورد ما و ثبات EBP حاوی طول رمز واقعی است که در اولین سعی مان بسیار فوشارنس بوده ایم که تعداد کاراکتر ها یکسان بوده ولی در سری بعدی که طول رمز کوتاه یا بلند تر بود پچ ما عمل نمیکرد ☺
حال بیایید برنامه را دوباره اجرا و رمز "77777777" را وارد کنید :

```

C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
77777777

```

و داریم:

```

C:\Users\NODONGLE\Desktop\CrackmeConsole.exe
Welcome to AR Crackme #1
Please enter the correct serial:
77777777
Checking serial
Congratulations that is correct!
Visit http://cracking.accessroot.com/ for everything AR
Long live the ARTeam!
Greetz & Shoutoutz to Whitefire for hosting our forum, forum.exetools.com, tech-arena.com, and the ARTeam
Press any key to continue . . .

```


موفق شدیم ،اما هدف واقعی این آموزش بیان تفاوت بین روش LAME و NOOB بود ،پنج اولی بوسیله ی NOOB و کشف رمز روش LAME بود

کلام آخر:

اگر یادتان باشد ما توضیحات را یکی از روش هایی مدانیم که باعث میشود درک کلی از رفتار برنامه یا قسمتی از آن بهبود یابد،در زیر من همان قسمت کد را مستند سازی کرده ام ،ببینید چقدر رامت تر شده :

00402570	83C8 04	OR EAX,4	### EDI = HC password
00402580	> 53	PUSH EBX	### X
00402581	50	PUSH EAX	### EDI = HC password
00402582	EB 100A0000	CALL CrackmeC.00402F97	### EDX = Password length
00402587	> 897C24 2C 10	CMP DWORD PTR SS:[ESP+2C],10	### CMP length with zero
0040258C	8B7C24 18	MOV EDI,DWORD PTR SS:[ESP+18]	### EBP = length of HC password
00402590	73 04	JNB SHORT CrackmeC.00402596	### Jump if password zero length
00402592	807C24 18	LEA EDI,DWORD PTR SS:[ESP+18]	### Is length < hard coded amount (8 - [esp+28])
00402596	> 8B5424 44	MOV EDX,DWORD PTR SS:[ESP+44]	### EDX = length
0040259A	3B03	CMP EDX,EBX	### X Jmp if our length is < hard coded length (8)
0040259C	8B6C24 28	MOV EBP,DWORD PTR SS:[ESP+28]	### EAX = Length of HC password
004025A0	74 26	JE SHORT CrackmeC.004025C8	### CMP 0x0F with 0x10 ???
004025A2	3B05	CMP EDX,EBP	### MOV First digits of entered password into ESI
004025A4	8BCA	MOV ECX,EDX	### X
004025A6	72 02	JB SHORT CrackmeC.004025AA	### ESI = entered password
004025A8	8BC0	MOV ECX,EBP	### EAX = 0
004025AA	> 837C24 48 10	CMP DWORD PTR SS:[ESP+48],10	### CMP H.C. password with entered password
004025AF	8B7424 34	MOV ESI,DWORD PTR SS:[ESP+34]	### JNP if they are the same
004025B3	73 04	JNB SHORT CrackmeC.004025B9	### EAX = FFFFFFFF
004025B5	807424 34	LEA ESI,DWORD PTR SS:[ESP+34]	### No change to EAX
004025B9	3308	XOR EAX,EAX	### EAX (FFFFFFFF) == EBX (0)
004025BB	F3:06	REPE CMPS BYTE PTR ES:[EDI],BYTE PTR DS	### <> !!!!!
004025BD	74 05	JE SHORT CrackmeC.004025C4	
004025BF	1BC0	SBB EAX,EAX	
004025C1	8308 FF	SBB EAX,-1	
004025C4	> 3BC3	CMP EAX,EBX	
004025C6	> 75 19	JNZ SHORT CrackmeC.004025DB	
004025C8	> 3B05	CMP EDX,EBP	
004025CA	72 0F	JB SHORT CrackmeC.004025DB	
004025CC	33C0	XOR EAX,EAX	
004025CE	3B05	CMP EDX,EBP	
004025D0	0F95C0	SETNE AL	
004025D3	3BC3	CMP EAX,EBX	
004025D5	0F84 CF020000	JE CrackmeC.004028AA	
004025D8	> 68 48E24000	PUSH CrackmeC.0040E248	
004025E0	68 A02F4100	PUSH CrackmeC.00412FA0	
004025E5	EB A6F4FFFF	CALL CrackmeC.00401A90	
004025EA	83C4 08	ADD ESP,8	
004025ED	8BF0	MOV ESI,EAX	
004025EF	6A 0A	PUSH 0A	
004025F1	8B05	MOV EBX,EBX	

ASCII "I'm sorry but that is wrong!"