

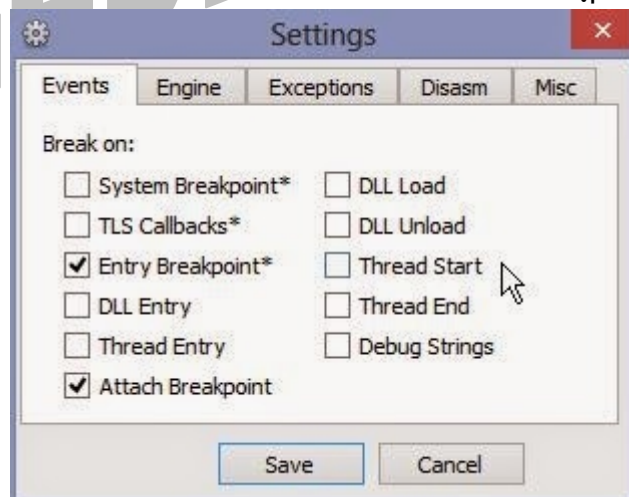
مقدمه ای سریع بر x64_dbg

به آموزش مقدماتی x64_dbg خوش آمدید، در این آموزش با اصول کار و امکانات این دیباگر آشنا میشیم، این دیباگر بسیار شبیه به دیباگر مد کاربر Olly است که هنوز در مرحله ی آزمایش و رفع مشکلات هست.

تنظیمات

پیکربندی x64_dbg کاملاً ساده، شبیه و مفصلتر از ollydbg1.xx است و به سادگی از منوی تنظیمات قابل دسترس است

۱.



همانطور که در تصویر بالا میبینید نمای پیکر بندی این دیباگر بسیار شبیه نسخه ی ollydbg 1.10 با اندکی تغییر است. که به ترتیب آنها را شرح میدهم :

System Breakpoint: این قسمت اگر فعال باشد، وقتی یک پروس در حال بارگذاری باشد، x64_dbg با توقف اجرای آن، فرایند اجرای پروس را در دست میگیرد

TLS Callbacks: پاسخ به تابعی که قبل از اجرای برنامه اصلی فراخوانی شده، به طور مثال میتواند شامل تنظیم پارامتر ها برای پیاده سازی پروتکتور باشد که باعث فضا در عملیات دیباگ بشود.

Entry Breakpoint: این قسمت اگر فعال باشد باعث توقف دیباگر در ابتدای برنامه میشود. البته به شرطی که برنامه پک نشده باشد یعنی در حال دیباگینگ نرمال باشیم.

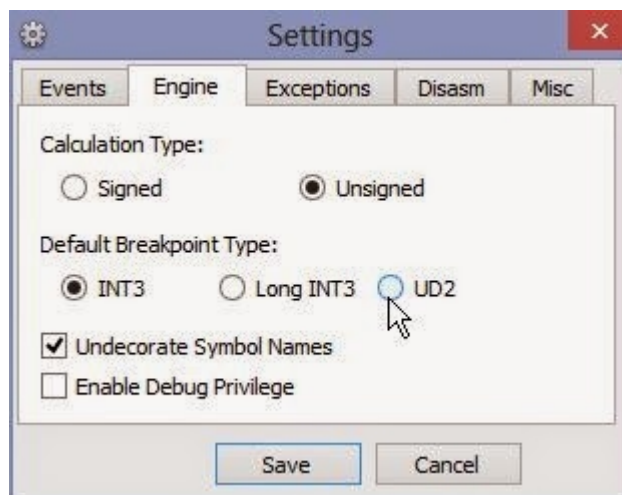
DLL Entry: اگر این گزینه فعال باشد دیباگر بر روی هر DLL که توسط پروس اجرا شود توقف میکند

Thread Entry: اگر این گزینه فعال باشد، دیباگر بر روی اولین دستورالعمل که در هر Thread جدیدی باشد متوقف میشود.

Attach Breakpoint: وقتی که به پروس فعال متصل باشیم، اگر این گزینه فعال باشد، دیباگر بر روی تابع DbgUiRemoteBreakin متوقف میشود.

DLL Load/DLL Unload: در حقیقت این یک عملکرد هست، یعنی موقعی یک dll توسط پروس فعال، بارگذاری میشود یا تخلیه میشود، دیباگر بر روی این عملکرد ها متوقف میشود.

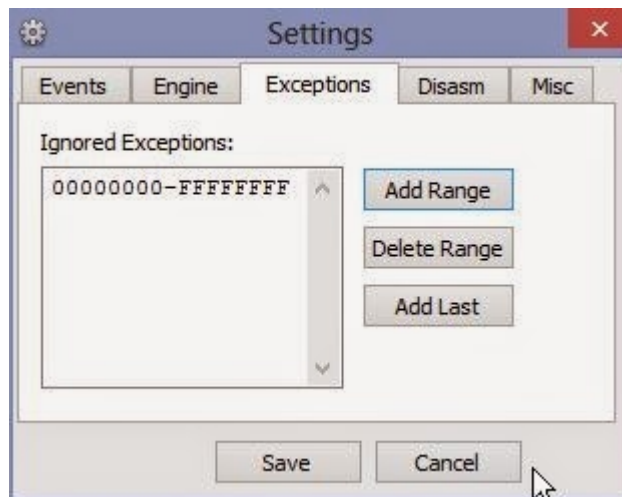
Thread Start/Thread End: این قابلیت به ما اجازه میدهد که کنترل بیشتری بر روی عملیات دیباگ داشته باشیم، یعنی هنگام شروع یا پایان یا متوقف شدن Threads



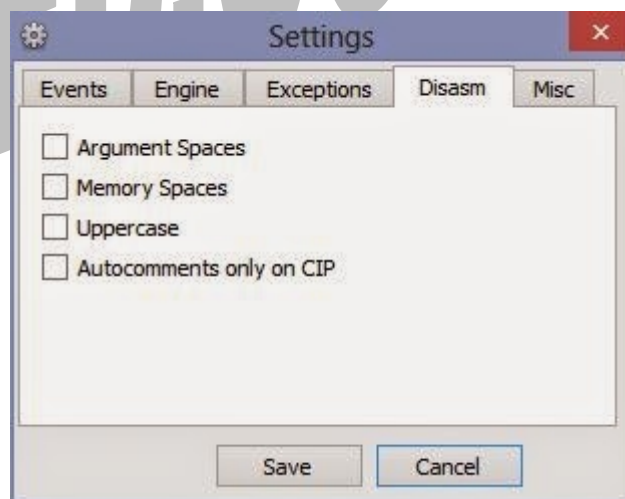
در این قسمت به صورت مختصر با تنظیمات موتور دیباگینگ آشنا میشویم که در اکثر موارد این تنظیمات بدون تغییر هستند

Enable Debug Privilege: این گزینه اجازی اتصال به پروس های سیستمی را میدهد

Breakpoint Type: این قابلیت به شما امکان استفاده از نقاط توقف مختلف را میدهد، این قابلیت موقعی خوب است که پروتکتور نقاط توقف پیش فرض (INT3) را شناسایی یا بلوک کند



پنجره ی استثنا ها، درحقیقت ما به دیباگر میگوئیم که از این نقطه ی توقف نافواسته چشم پوشی کند و روند دیباگ را ادامه دهد، اگر میخواهید تمام استثنا ها به طور خودکار نادیده گرفته شوند میتوانید طبق تصویر بالا محدوده ی آدرس را وارد کنید.



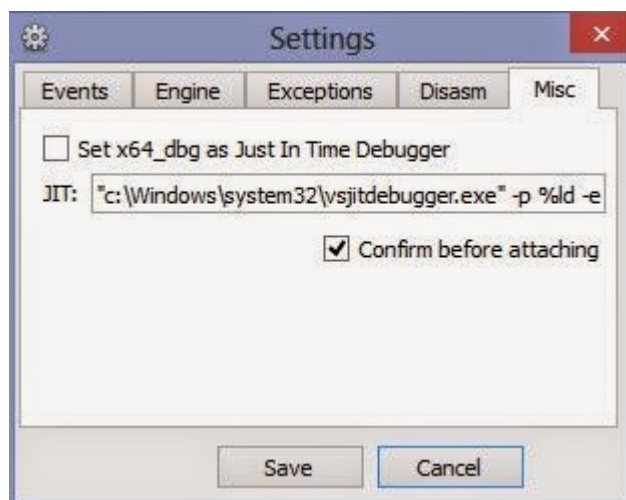
در این قسمت ما توانیم تغییراتی در ظاهر دستورالعمل های اعمال کنیم که :

Argument Spaces: ایجاد فضای خالی مابین هر آرگومان

Memory Spaces: ایجاد فضای خالی بین دستورالعمل های حافظه ایی

Uppercase: تغییر همه متن های کوچک به بزرگ

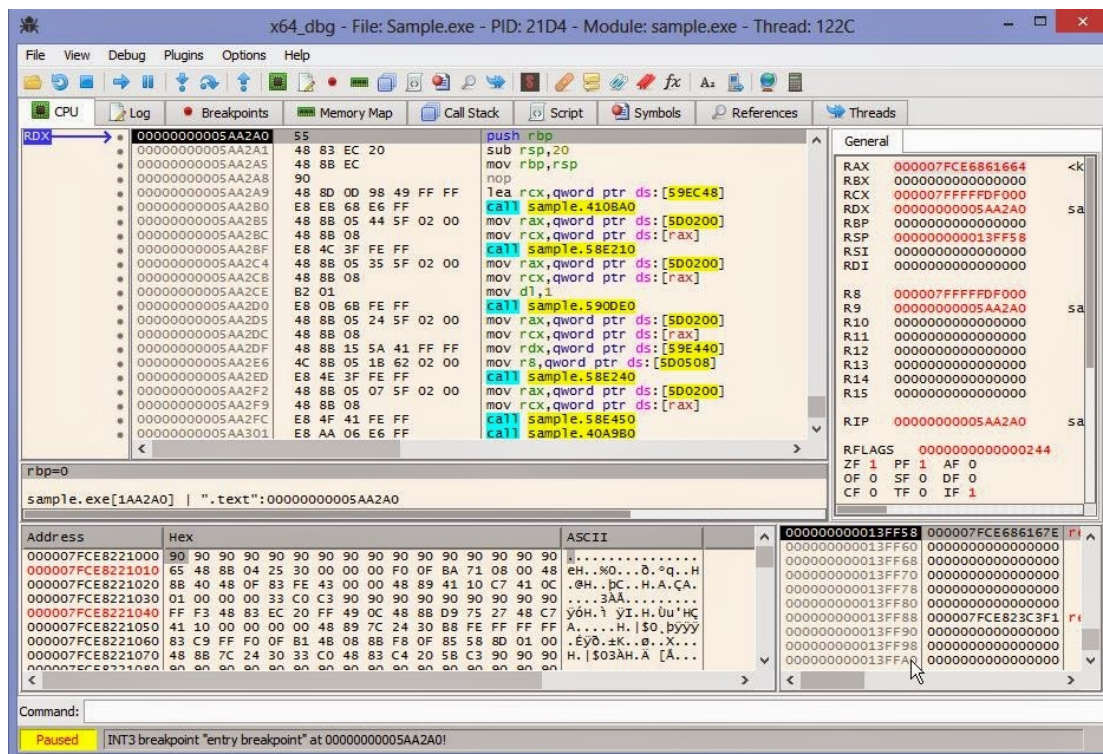
Autocomments only on CIP: حذف تمامی کامنت ها



این قسمت این قابلیت را به ما میدهد که x64_dbg را به برای Just in Time Debugger تنظیم کنیم

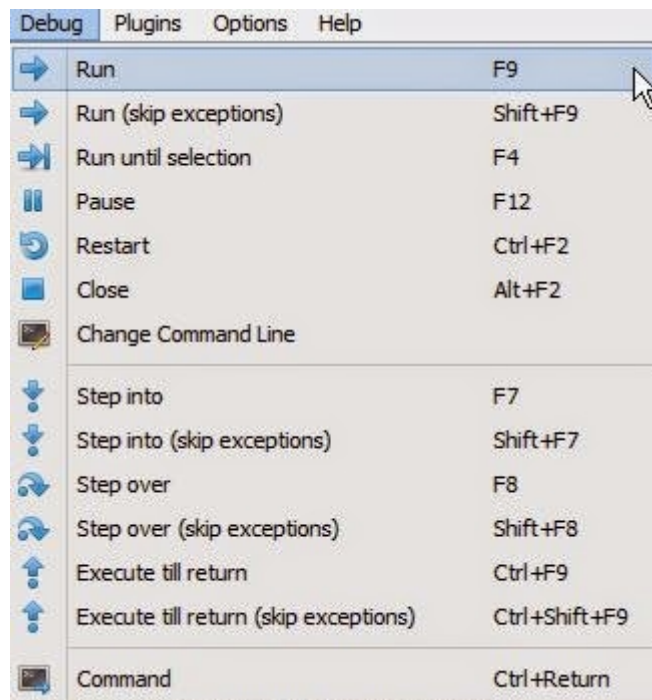
به منظور درک بیشتر با محیط این دیباگر یک برنامه کوچک را مورد بررسی قرار خواهیم داد، این برنامه را میتوانید از آدرس زیر دریافت کنید:
sourceforge.net/projects/x64dbg/files/extra

نمای بعد از یاز کردن برنامه توسط دیباگر :



اگر پیکر بندی پیش فرض دیباگر انتخاب شده باشد شما باید بر روی ماژول اصلی (entry point) متوقف شوید

اکثر گزینه ها ی (اِهبردی) (کلید ها) (دیباگینگ شبیه ollydbg هستند:



به صورت مختصر :

Run(F9): شروع یا از سر گرفته شدن پروس در حالت معمولی

Run(skip exceptions) (Shift+F9): عبور از استثنا ها

Pause(F12): تعلیق روند فعلی

Restart(Ctrl+F2): پایان روند دیباگ و اجرای مجدد آن

Close(Alt+F4): پایان روند دیباگ و عدم بارگیری مجدد

Step Into(F7): ورود به یک روال یا اجرای یک ملقه

Step Into(skip exceptions)(Shift+F7): مانند دستور قبلی با قابلیت چشم پوشی از استثنا ها

Step Over(F8): اجرای یک زیر روال یا ملقه به صورت یکجا

Step Over(skip exceptions)(Shift+F8): مانند دستور قبل قبلی با قابلیت چشم پوشی از

استثنا ها

Execute Till Return(Ctrl+F9): از سرگیری پروس هنگامی که دستور العمل بازگشتی اجرا

میشود

Execute Till Return(skip exceptions)(Ctrl+Shift+F9): مانند دستور قبلی با قابلیت

چشم پوشی از استثنا ها

Breakpoints:

هنگامی که شما در مین عمل دیبگینگ هستید یکی از اصلی ترین ویژه گیها Breakpoints

هستند، که شامل سه نوع میشوند که به صورت مختصر توضیح داده میشوند:

Execution Breakpoint: رایجترین نوع نقاط توقف هستند که با تخییر در بایت اول opcode با

مقدار 0xCC باعث توقف میشود.

که با کلید F2 قابل گذاردن است

Memory Breakpoint: توقف یک منطقه ی خاص از حافظه

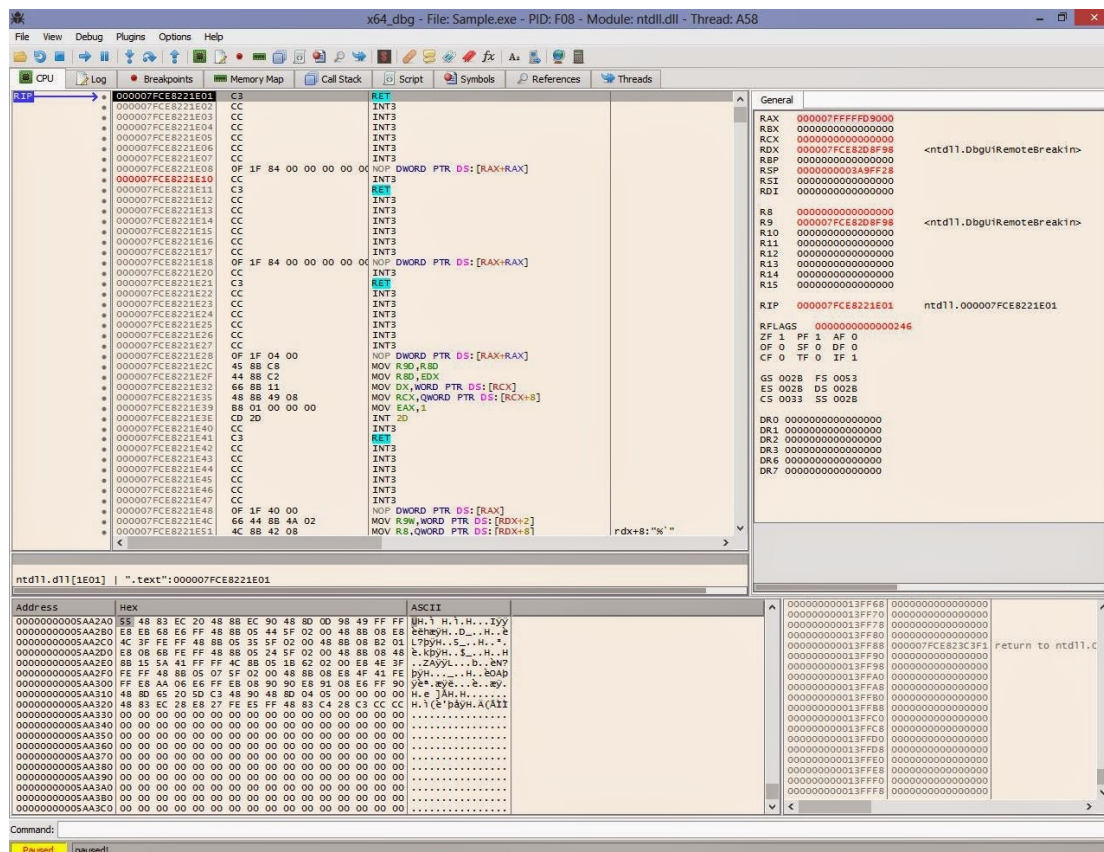
Hardware Breakpoint: باعث توقف برنامه هنگام خواندن یا نوشتن یا اجرا شدن در یک آدرس

میشود .

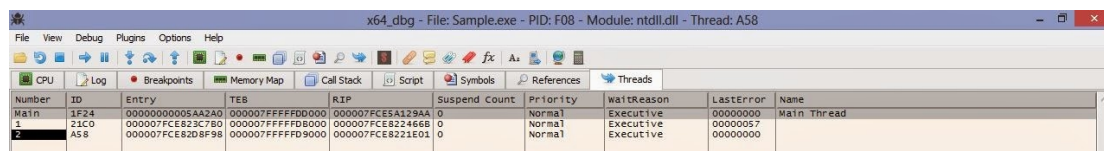
شروع:

بعد از بارگذاری برنامه برنامه را با کلید F9 اجرا کنید ، و بعد از اجرا برنامه را با کلید F12 متوقف

کنید



سپس بر روی Threads > Tab بروید ،همانطور که در تصویر میبینید بر روی Thread اصلی متوقف شده ایم



زمانی که ما بر روی Threads اصلی هستیم میتوانیم به صورت مرحله ایی روتین ها را تحلیل و همینطور پشته را آنالیز کنیم

پنجره ی Call Stack لیست توابع روتین جاری و کلا اطلاعات مربوطه را نشان میدهد که شما با دابل کلیک بر روی آن میتوانید به محل اجاعات سوئیچ کنید.که لازمه درک بهتر آن ،آنالیز برنامه ی هدف است،با دابل کلیک بر روی هرکدام در پنجره ی disassembler قابل رویت هست

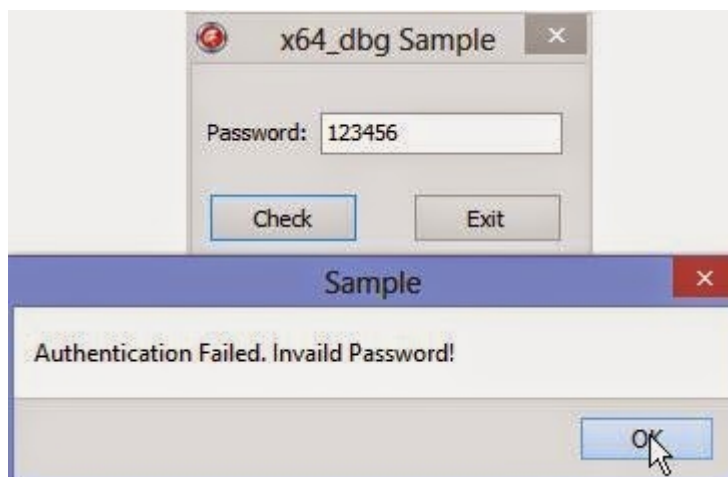
Address	To	From	Comment
0000000000013FE08	0000000000058F5B7	00000000000416F00	return to sample.0000000000058F5B7 from sample.00000000000416F00
0000000000013FE78	0000000000058E088	0000000000058F450	return to sample.0000000000058E088 from sample.0000000000058F450
0000000000013FE8E	0000000000058E544	0000000000058E090	return to sample.0000000000058E544 from sample.0000000000058E090
0000000000013FF28	000000000005AA301	0000000000058E450	return to sample.000000000005AA301 from sample.0000000000058E450
0000000000013FF58	000007FCE686167E		return to kernel32.000007FCE686167E from ???
0000000000013FF88	000007FCE823C3F1		return to ntdll.000007FCE823C3F1 from ???

پنجره ی Memory Map حاوی سکشن ها وهمچنین مناطق اختصاص داده شده در حافظه و تمامی منابع لازم برای اجرای پروس است.

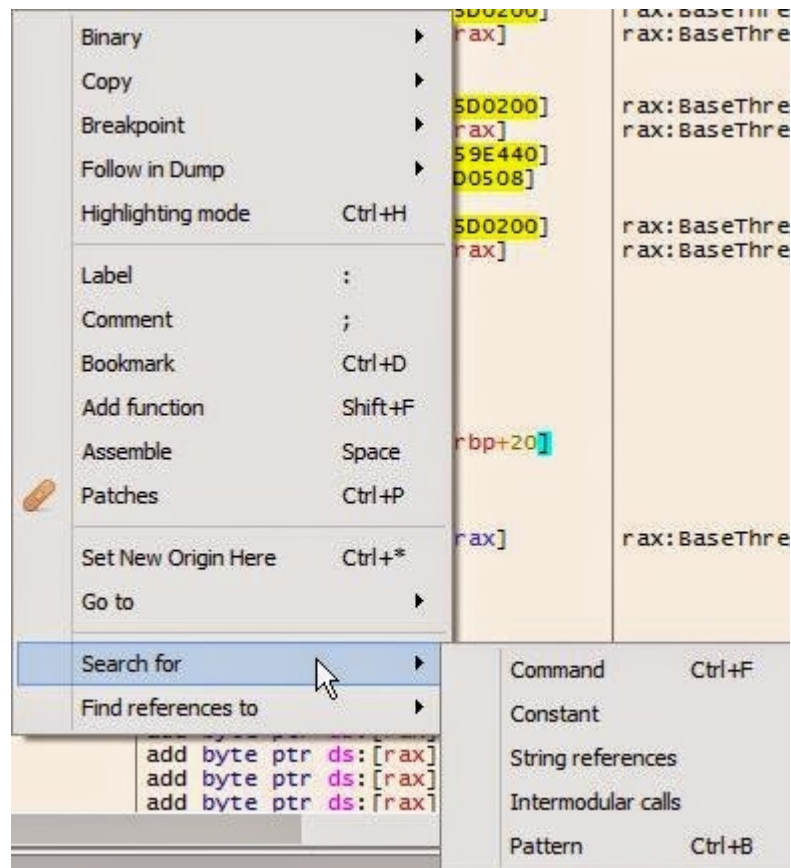
ADDR	SIZE	INFO	TYPE	PROT	APROT
0000000000001000	000000000000010000		MAP	-RW--	-RW--
0000000000002000	000000000000010000		PRV	-RW-	-RW-
0000000000003000	000000000000090000		MAP	-R--	-R--
0000000000004000	000000000000080000		PRV	-RW-G	-RW--
0000000000005000	000000000000030000		MAP	-R--	-R--
0000000000006000	000000000000020000		MAP	-R--	-R--
0000000000007000	000000000000020000		PRV	-RW-	-RW-
0000000000008000	000000000000075000		MAP	-R--	-R--
0000000000009000	000000000000020000		PRV	-RW-	-RW--
000000000000A000	000000000000010000		PRV	-RW-	-RW-
000000000000B000	000000000000010000		PRV	-RW-	-RW--
000000000000C000	000000000000080000		PRV	-RW-	-RW--
000000000000D000	000000000000040000		MAP	-R--	-R--
000000000000E000	000000000000020000		PRV	-RW-	-RW-
000000000000F000	000000000000010000		PRV	ERW-	ERW--
0000000000000000	000000000000000000	sample.exe	IMG	-R--	ERW-
0000000000000100	00000000000001AA000	"text"	IMG	ER--	ERW-
0000000000000200	000000000000006000	"data"	IMG	-RW-	ERW-
0000000000000300	00000000000000A000	"bss"	IMG	-RW-	ERW-
0000000000000400	000000000000005000	".idata"	IMG	-RW-	ERW-
0000000000000500	000000000000001000	".idata"	IMG	-RW-	ERW-
0000000000000600	000000000000001000	".tls"	IMG	-RW-	ERW-
0000000000000700	000000000000001000	".rdata"	IMG	-R--	ERW-
0000000000000800	000000000000017000	".pdata"	IMG	-R--	ERW-
0000000000000900	00000000000004F000	".rsrvc"	IMG	-R--	ERW-
0000000000000A00	000000000000033000		MAP	-R--	-R--
0000000000000B00	000000000000002000		PRV	-RW-	-RW--
0000000000000C00	000000000000003000		MAP	-R--	-R--
0000000000000D00	000000000000004000		PRV	-RW-	-RW--
0000000000000E00	000000000000001000		MAP	-R--	-R--
0000000000000F00	000000000000007800		MAP	-R--	-R--
0000000000001000	000000000000008000		MAP	-R--	-R--
0000000000001100	000000000000008000		MAP	-R--	-R--
0000000000001200	000000000000008000		MAP	-R--	-R--
0000000000001300	000000000000008000		MAP	-R--	-R--
0000000000001400	000000000000008000		MAP	-R--	-R--
0000000000001500	000000000000008000		MAP	-R--	-R--
0000000000001600	000000000000008000		MAP	-R--	-R--
0000000000001700	000000000000008000		MAP	-R--	-R--
0000000000001800	000000000000008000		MAP	-R--	-R--
0000000000001900	000000000000008000		MAP	-R--	-R--
0000000000001A00	000000000000008000		MAP	-R--	-R--
0000000000001B00	000000000000008000		MAP	-R--	-R--
0000000000001C00	000000000000008000		MAP	-R--	-R--
0000000000001D00	000000000000008000		MAP	-R--	-R--
0000000000001E00	000000000000008000		MAP	-R--	-R--
0000000000001F00	000000000000008000		MAP	-R--	-R--
0000000000002000	000000000000008000		MAP	-R--	-R--
0000000000002100	000000000000008000		MAP	-R--	-R--
0000000000002200	000000000000008000		MAP	-R--	-R--
0000000000002300	000000000000008000		MAP	-R--	-R--
0000000000002400	000000000000008000		MAP	-R--	-R--
0000000000002500	000000000000008000		MAP	-R--	-R--
0000000000002600	000000000000008000		MAP	-R--	-R--
0000000000002700	000000000000008000		MAP	-R--	-R--
0000000000002800	000000000000008000		MAP	-R--	-R--
0000000000002900	000000000000008000		MAP	-R--	-R--
0000000000002A00	000000000000008000		MAP	-R--	-R--
0000000000002B00	000000000000008000		MAP	-R--	-R--
0000000000002C00	000000000000008000		MAP	-R--	-R--
0000000000002D00	000000000000008000		MAP	-R--	-R--
0000000000002E00	000000000000008000		MAP	-R--	-R--
0000000000002F00	000000000000008000		MAP	-R--	-R--
0000000000003000	000000000000008000		MAP	-R--	-R--
0000000000003100	000000000000008000		MAP	-R--	-R--
0000000000003200	000000000000008000		MAP	-R--	-R--
0000000000003300	000000000000008000		MAP	-R--	-R--
0000000000003400	000000000000008000		MAP	-R--	-R--
0000000000003500	000000000000008000		MAP	-R--	-R--
0000000000003600	000000000000008000		MAP	-R--	-R--
0000000000003700	000000000000008000		MAP	-R--	-R--
0000000000003800	000000000000008000		MAP	-R--	-R--
0000000000003900	000000000000008000		MAP	-R--	-R--
0000000000003A00	000000000000008000		MAP	-R--	-R--
0000000000003B00	000000000000008000		MAP	-R--	-R--
0000000000003C00	000000000000008000		MAP	-R--	-R--
0000000000003D00	000000000000008000		MAP	-R--	-R--
0000000000003E00	000000000000008000		MAP	-R--	-R--
0000000000003F00	000000000000008000		MAP	-R--	-R--
0000000000004000	000000000000008000		MAP	-R--	-R--
0000000000004100	000000000000008000		MAP	-R--	-R--
0000000000004200	000000000000008000		MAP	-R--	-R--
0000000000004300	000000000000008000		MAP	-R--	-R--
0000000000004400	000000000000008000		MAP	-R--	-R--
0000000000004500	000000000000008000		MAP	-R--	-R--
0000000000004600	000000000000008000		MAP	-R--	-R--
0000000000004700	000000000000008000		MAP	-R--	-R--
0000000000004800	000000000000008000		MAP	-R--	-R--
0000000000004900	000000000000008000		MAP	-R--	-R--
0000000000004A00	000000000000008000		MAP	-R--	-R--
0000000000004B00	000000000000008000		MAP	-R--	-R--
0000000000004C00	000000000000008000		MAP	-R--	-R--
0000000000004D00	000000000000008000		MAP	-R--	-R--
0000000000004E00	000000000000008000		MAP	-R--	-R--
0000000000004F00	000000000000008000		MAP	-R--	-R--
0000000000005000	000000000000008000		MAP	-R--	-R--
0000000000005100	000000000000008000		MAP	-R--	-R--
0000000000005200	000000000000008000		MAP	-R--	-R--
0000000000005300	000000000000008000		MAP	-R--	-R--
0000000000005400	000000000000008000		MAP	-R--	-R--
0000000000005500	000000000000008000		MAP	-R--	-R--
0000000000005600	000000000000008000		MAP	-R--	-R--
0000000000005700	000000000000008000		MAP	-R--	-R--
0000000000005800	000000000000008000		MAP	-R--	-R--
0000000000005900	000000000000008000		MAP	-R--	-R--
0000000000005A00	000000000000008000		MAP	-R--	-R--
0000000000005B00	000000000000008000		MAP	-R--	-R--
0000000000005C00	000000000000008000		MAP	-R--	-R--
0000000000005D00	000000000000008000		MAP	-R--	-R--
0000000000005E00	000000000000008000		MAP	-R--	-R--
0000000000005F00	000000000000008000		MAP	-R--	-R--
0000000000006000	000000000000008000		MAP	-R--	-R--
0000000000006100	000000000000008000		MAP	-R--	-R--
0000000000006200	000000000000008000		MAP	-R--	-R--
0000000000006300	000000000000008000		MAP	-R--	-R--
0000000000006400	000000000000008000		MAP	-R--	-R--
0000000000006500	000000000000008000		MAP	-R--	-R--
0000000000006600	000000000000008000		MAP	-R--	-R--
0000000000006700	000000000000008000		MAP	-R--	-R--
0000000000006800	000000000000008000		MAP	-R--	-R--
0000000000006900	000000000000008000		MAP	-R--	-R--
0000000000006A00	000000000000008000		MAP	-R--	-R--
0000000000006B00	000000000000008000		MAP	-R--	-R--
0000000000006C00	000000000000008000		MAP	-R--	-R--
0000000000006D00	000000000000008000		MAP	-R--	-R--
0000000000006E00	000000000000008000		MAP	-R--	-R--
0000000000006F00	000000000000008000		MAP	-R--	-R--
0000000000007000	000000000000008000		MAP	-R--	-R--
0000000000007100	000000000000008000		MAP	-R--	-R--
0000000000007200	000000000000008000		MAP	-R--	-R--
0000000000007300	000000000000008000		MAP	-R--	-R--
0000000000007400	000000000000008000		MAP	-R--	-R--
0000000000007500	000000000000008000		MAP	-R--	-R--
0000000000007600	000000000000008000		MAP	-R--	-R--
0000000000007700	000000000000008000		MAP	-R--	-R--
0000000000007800	000000000000008000		MAP	-R--	-R--
0000000000007900	000000000000008000		MAP	-R--	-R--
0000000000007A00	000000000000008000		MAP	-R--	-R--
0000000000007B00	000000000000008000		MAP	-R--	-R--
0000000000007C00	000000000000008000		MAP	-R--	-R--
0000000000007D00	000000000000008000		MAP	-R--	-R--
0000000000007E00	000000000000008000		MAP	-R--	-R--
0000000000007F00	000000000000008000		MAP	-R--	-R--
0000000000008000	000000000000008000		MAP	-R--	-R--
0000000000008100	000000000000008000		MAP	-R--	-R--
0000000000008200	000000000000008000		MAP	-R--	-R--
0000000000008300	000000000000008000		MAP	-R--	-R--
0000000000008400	000000000000008000		MAP	-R--	-R--
0000000000008500	000000000000008000		MAP	-R--	-R--
0000000000008600	000000000000008000		MAP	-R--	-R--
0000000000008700	000000000000008000		MAP	-R--	-R--
0000000000008800	000000000000008000		MAP	-R--	-R--
0000000000008900	000000000000008000		MAP	-R--	-R--
0000000000008A00	000000000000008000		MAP	-R--	-R--
0000000000008B00	000000000000008000		MAP	-R--	-R--
0000000000008C00	000000000000008000		MAP	-R--	-R--
0000000000008D00	000000000000008000		MAP	-R--	-R--
0000000000008E00	000000000000008000		MAP	-R--	-R--
0000000000008F00	000000000000008000		MAP	-R--	-R--
0000000000009000	000000000000008000		MAP	-R--	-R--
0000000000009100	000000000000008000		MAP	-R--	-R--
0000000000009200	000000000000008000		MAP	-R--	-R--
0000000000009300	000000000000008000		MAP	-R--	-R--
0000000000009400	000000000000008000		MAP	-R--	-R--
0000000000009500	000000000000008000		MAP	-R--	-R--
0000000000009600	000000000000008000		MAP	-R--	-R--
0000000000009700	000000000000008000		MAP	-R--	-R--
0000000000009800	000000000000008000		MAP	-R--	-R--
0000000000009900	000000000000008000		MAP	-R--	-R--
0000000000009A00	000000000000008000		MAP	-R--	-R--
0000000000009B00	000000000000008000		MAP	-R--	-R--
00000					

فوب، برنامه را با F9 اجرا کنید و در کادر درخواست پسورد عدد یا مروف دلفواه را وارد کنید، ما در این مثال اعداد:

۱۲۳۴۵۶ را وارد کرده و روی دکمه ی check کلیک میکنیم:



همان طور که در تصویر بالا مشاهده میکنید رمز وارد شده معتبر نیست، فب حال ما پیام خطا مبنی بر اشتباه بودن رمز را میدانیم، اجازه دهید از روش های پایه ای استفاده کنیم (قبلا توضیح دادیم که این نمونه بصورت معمولی و بدون پروتکتور است) از طریق referenced strings، پیام مورد نظر را مستخرج میکنیم



همانطور که در تصویر زیر میبینید پس از جستجوی کوتاه رشته مورد نظر و همینطور رشته های جالب دیگر را پیدا کرده ایم

000000000059EA4A	lea rdx,qword ptr ds:[59EAF8]	L"10db8e415b857a61e18ef5d4db8e4f38"
000000000059EA5A	lea rcx,qword ptr ds:[59EB48]	L"Valid Password. Welcome, Authorized User."
000000000059EA68	lea rcx,qword ptr ds:[59EB48]	L"Authentication Failed. Invalid Password!"

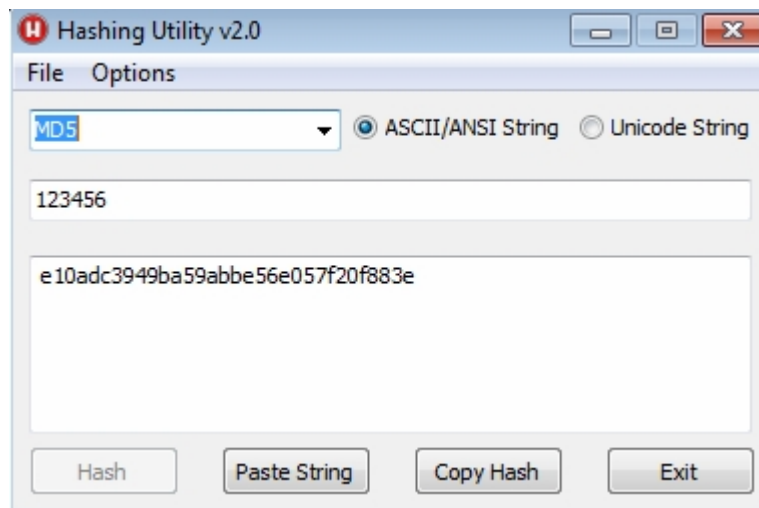
با دابل کلیک برروی عبارت دلفواه، شما به محل مورد نظر سوئیچ فواید کرد
حال با ایجاد یک نقطه ی توقف از نوع معمولی با کلید F2 روند کار خود را ادامه میدهیم

* 000059E9ED	CC	int3	
* 000059E9EE	CC	int3	
* 000059E9EF	CC	int3	
* 000059E9F0	55	push rbp	
* 000059E9F1	48 83 EC 40	sub rsp,40	
* 000059E9F5	48 88 EC	mov rbp,rsp	
* 000059E9F8	48 89 4D 20	mov qword ptr ss:[rbp+20],rcx	
* 000059E9FC	48 C7 45 38 00 00 00 00	mov qword ptr ss:[rbp+38],0	
* 000059EA04	48 C7 45 30 00 00 00 00	mov qword ptr ss:[rbp+30],0	
* 000059EA0C	48 C7 45 28 00 00 00 00	mov qword ptr ss:[rbp+28],0	
* 000059EA14	90	nop	
* 000059EA15	48 88 45 20	mov rax,qword ptr ss:[rbp+20]	
* 000059EA19	48 88 88 58 06 00 00	mov rcx,qword ptr ds:[rax+658]	
* 000059EA20	48 8D 55 30	lea rdx,qword ptr ss:[rbp+30]	
* 000059EA24	E8 77 53 F2 FF	call sample.4C3DA0	Gets Edit1.text
* 000059EA29	48 8D 4D 28	lea rcx,qword ptr ss:[rbp+28]	
* 000059EA2D	48 88 55 30	mov rdx,qword ptr ss:[rbp+30]	
* 000059EA31	4D 33 C0	xor r8,r8	
* 000059EA34	E8 C7 D2 E6 FF	call sample.40BD00	
* 000059EA39	48 8D 4D 38	lea rcx,qword ptr ss:[rbp+38]	
* 000059EA3D	48 8B 55 28	mov rdx,qword ptr ss:[rbp+28]	
* 000059EA41	E8 CA FE FF FF	call sample.59E910	Hashes the value
* 000059EA46	48 8B 4D 38	mov rcx,qword ptr ss:[rbp+38]	
* 000059EA4A	48 8D 15 A7 00 00 00	lea rdx,qword ptr ds:[59EAF8]	59EAF8:L"10db8e415b857a61e18ef5d4db8e4f38"
* 000059EA51	E8 2A DD E6 FF	call sample.40C780	
* 000059EA56	85 C0	test eax,eax	
* 000059EA58	75 0E	jnz sample.59EA68	
* 000059EA5A	48 8D 0E E7 00 00 00	lea rcx,qword ptr ds:[59EB48]	59EB48:L"Valid Password. Welcome, Authoriz
* 000059EA61	E8 1A 58 F6 FF	call sample.504280	
* 000059EA66	EB 0C	jmp sample.59EA74	
* 000059EA68	48 8D 0D 39 01 00 00	lea rcx,qword ptr ds:[59EBA8]	59EBA8:L"Authentication Failed. Invalid Pas
* 000059EA6F	E8 0C 58 F6 FF	call sample.504280	
* 000059EA74	90	nop	
* 000059EA75	48 8D 4D 28	lea rcx,qword ptr ss:[rbp+28]	
* 000059EA79	E8 42 C4 E6 FF	call sample.40AEC0	
* 000059EA7E	48 8D 4D 30	lea rcx,qword ptr ss:[rbp+30]	
* 000059EA82	E8 E9 C3 E6 FF	call sample.40AE70	
* 000059EA87	48 8D 4D 38	lea rcx,qword ptr ss:[rbp+38]	
* 000059EA88	E8 E0 C3 E6 FF	call sample.40AE70	
* 000059EA90	48 8D 65 40	lea rsp,qword ptr ss:[rbp+40]	
* 000059EA94	5D	pop rbp	
* 000059EA95	C3	ret	
* 000059EA96	48 90	nop	

با اجرای مجدد و وارد کردن دوباره رمز عبور(۱۲۳۴۵۶) و فشردن دکمه ی check متوقف شده و روند دیباگینگ را با کلید F8 ادامه می‌دهیم همانطور که می‌بینیم می‌فهمیم الگوریتم hash شده را بیابیم

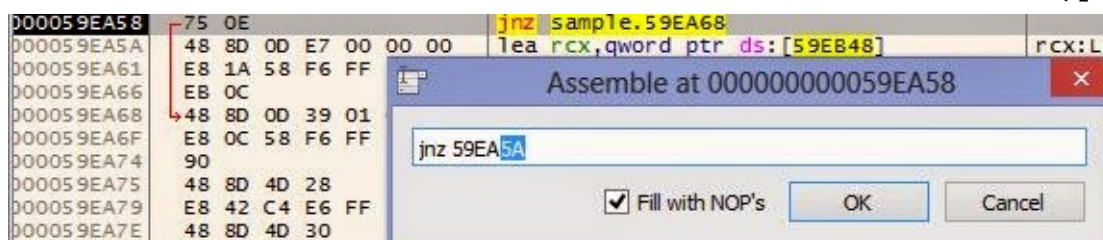
* 000059E9ED	CC	int3	
* 000059E9EE	CC	int3	
* 000059E9EF	CC	int3	
* 000059E9F0	55	push rbp	
* 000059E9F1	48 83 EC 40	sub rsp,40	
* 000059E9F5	48 88 EC	mov rbp,rsp	
* 000059E9F8	48 89 4D 20	mov qword ptr ss:[rbp+20],rcx	
* 000059E9FC	48 C7 45 38 00 00 00 00	mov qword ptr ss:[rbp+38],0	[rbp+38]: e10adc3949ba59abbe56e057f20f883e
* 000059EA04	48 C7 45 30 00 00 00 00	mov qword ptr ss:[rbp+30],0	[rbp+30]: 123456
* 000059EA0C	48 C7 45 28 00 00 00 00	mov qword ptr ss:[rbp+28],0	[rbp+28]: 123456
* 000059EA14	90	nop	
* 000059EA15	48 88 45 20	mov rax,qword ptr ss:[rbp+20]	
* 000059EA19	48 88 88 58 06 00 00	mov rcx,qword ptr ds:[rax+658]	rcx:L"10db8e415b857a61e18ef5d4db8e4f38"
* 000059EA20	48 8D 55 30	lea rdx,qword ptr ss:[rbp+30]	[rbp+30]: 123456
* 000059EA24	E8 77 53 F2 FF	call sample.4C3DA0	Gets Edit1.text
* 000059EA29	48 8D 4D 28	lea rcx,qword ptr ss:[rbp+28]	[rbp+28]: 123456
* 000059EA2D	48 88 55 30	mov rdx,qword ptr ss:[rbp+30]	[rbp+30]: 123456
* 000059EA31	4D 33 C0	xor r8,r8	
* 000059EA34	E8 C7 D2 E6 FF	call sample.40BD00	
* 000059EA39	48 8D 4D 38	lea rcx,qword ptr ss:[rbp+38]	[rbp+38]: e10adc3949ba59abbe56e057f20f883e
* 000059EA3D	48 8B 55 28	mov rdx,qword ptr ss:[rbp+28]	[rbp+28]: 123456
* 000059EA41	E8 CA FE FF FF	call sample.59E910	Hashes the value
* 000059EA46	48 8B 4D 38	mov rcx,qword ptr ss:[rbp+38]	[rbp+38]: e10adc3949ba59abbe56e057f20f883e
* 000059EA4A	48 8D 15 A7 00 00 00	lea rdx,qword ptr ds:[59EAF8]	59EAF8:L"10db8e415b857a61e18ef5d4db8e4f38"
* 000059EA51	E8 2A DD E6 FF	call sample.40C780	
* 000059EA56	85 C0	test eax,eax	
* 000059EA58	75 0E	jnz sample.59EA68	
* 000059EA5A	48 8D 0E E7 00 00 00	lea rcx,qword ptr ds:[59EB48]	rcx:L"10db8e415b857a61e18ef5d4db8e4f38", ss
* 000059EA61	E8 1A 58 F6 FF	call sample.504280	
* 000059EA66	EB 0C	jmp sample.59EA74	
* 000059EA68	48 8D 0D 39 01 00 00	lea rcx,qword ptr ds:[59EBA8]	rcx:L"10db8e415b857a61e18ef5d4db8e4f38", ss
* 000059EA6F	E8 0C 58 F6 FF	call sample.504280	
* 000059EA74	90	nop	
* 000059EA75	48 8D 4D 28	lea rcx,qword ptr ss:[rbp+28]	[rbp+28]: 123456
* 000059EA79	E8 42 C4 E6 FF	call sample.40AEC0	
* 000059EA7E	48 8D 4D 30	lea rcx,qword ptr ss:[rbp+30]	[rbp+30]: 123456
* 000059EA82	E8 E9 C3 E6 FF	call sample.40AE70	
* 000059EA87	48 8D 4D 38	lea rcx,qword ptr ss:[rbp+38]	[rbp+38]: e10adc3949ba59abbe56e057f20f883e
* 000059EA88	E8 E0 C3 E6 FF	call sample.40AE70	
* 000059EA90	48 8D 65 40	lea rsp,qword ptr ss:[rbp+40]	
* 000059EA94	5D	pop rbp	
* 000059EA95	C3	ret	
* 000059EA96	48 90	nop	

همانطور که در بالا می‌بینید که hash شده برای رمز ما(۱۲۳۴۵۶) این است:
e10adc3949ba59abbe56e057f20f883e



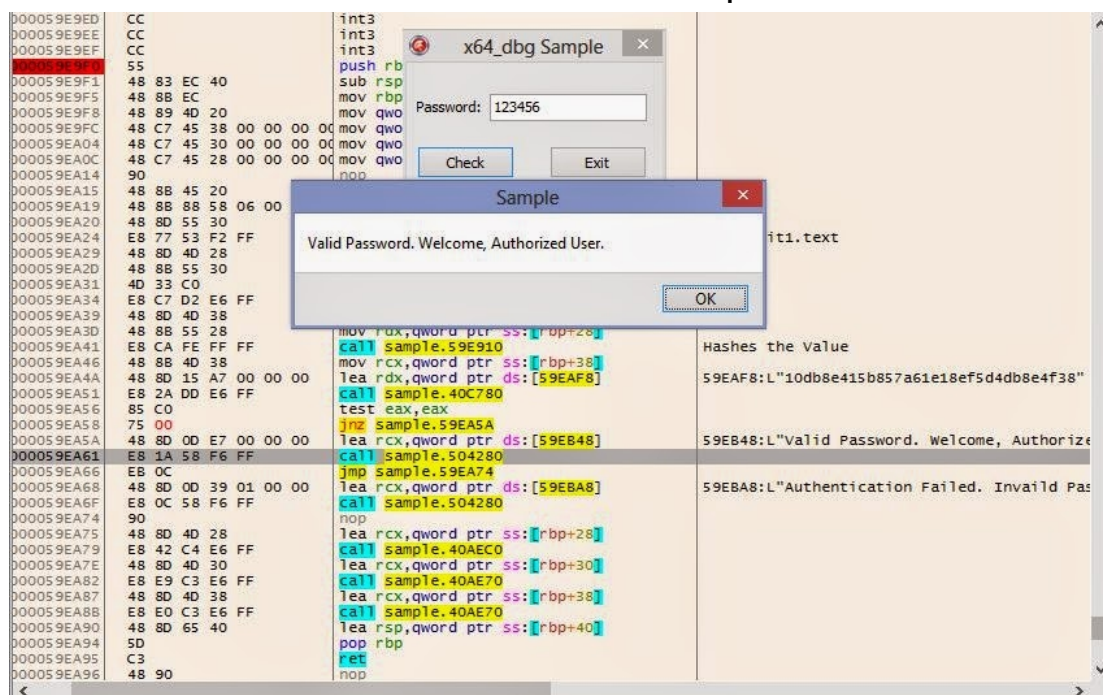
به نظر می‌رسد که الگوریتم مورد استفاده MD5 است. که برای رشته های طولانی پروت فورس کردن بسیار زمان بر خواهد بود، اما میتوان با تغییر آدرس پرش شرط روند را به نفع خودمان تغییر

دهیم

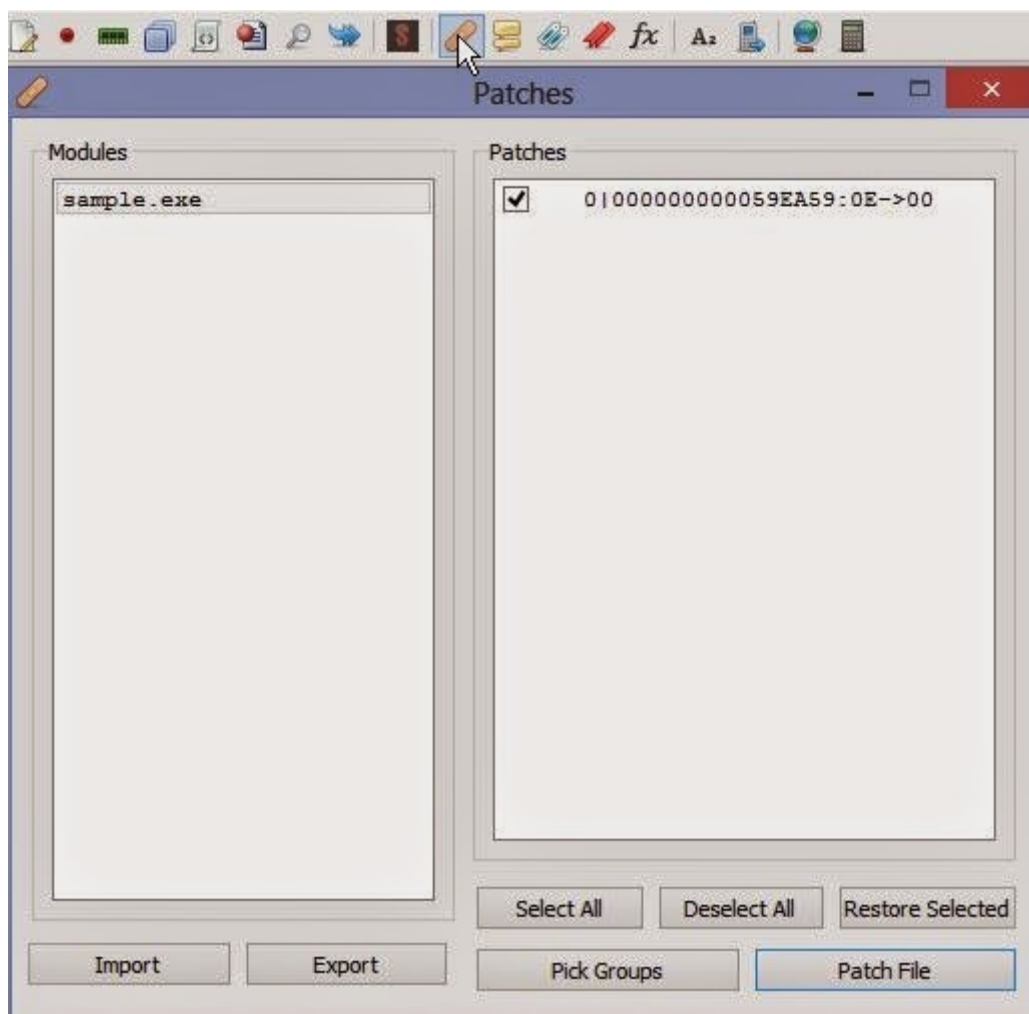


که در تصویر زیر ملاحظه میکنید

نکته: از آنجایی که هدف این مقاله چیز دیگری است از توضیحات ماشیه ایی خودداری میکنیم

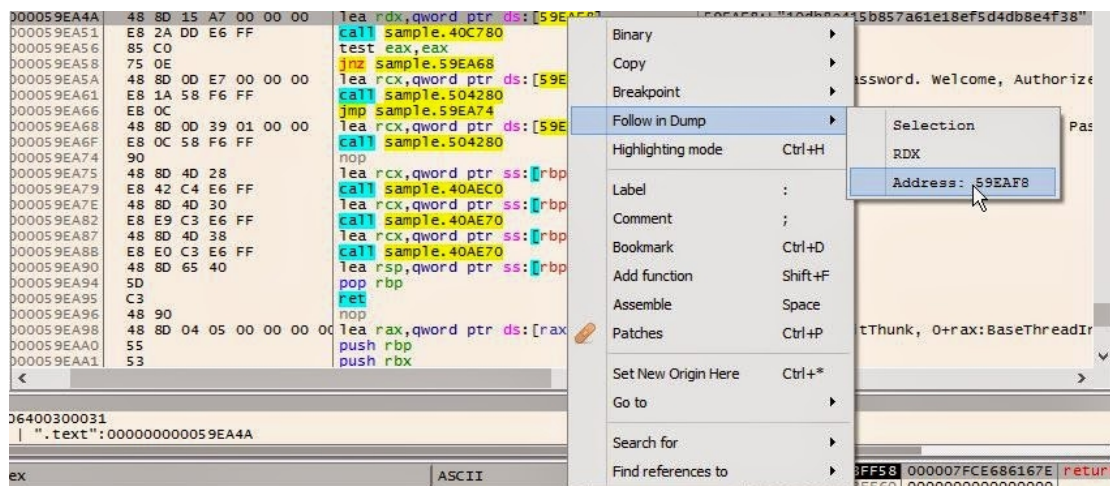


همانطور که میبینید هر رمزی مکمل رمز صمیع را دارد ولی این تغییرات فقط در حافظه است. برای پچ کردن دائمی میتوان با انتخاب گزینه ی Patche اقدام به سافت فایل پچ شده برروی رسانه ی ذخیره سازی کرد

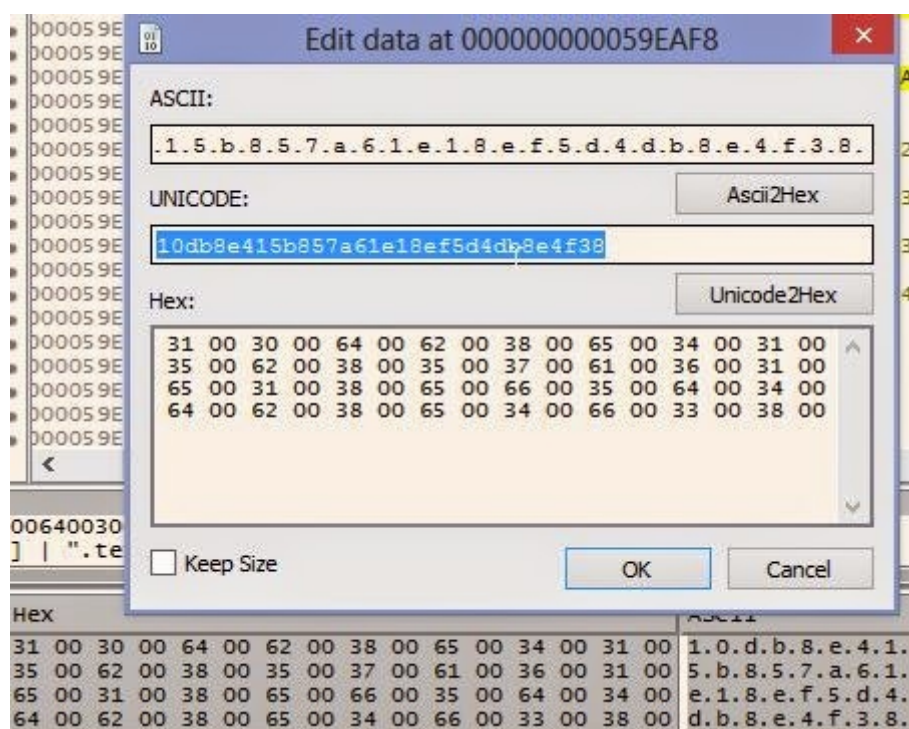


خب، کار تمام شد

همانطور که قبلا دیدیم ما میدانیم الگوریتم رمزنگاری چیست و همینطور رمز کد شده را هم میدانیم بیا باید با استفاده از سایت های آنلاین که قابلیت جستجو در دیتا بیس خود را دارند امتحان کنیم



با استفاده دامپ مقدار کد شده در آدرس موردنظر که همان رمز (۱۳۱۴۵۶) است را درون کلیپ برد ذخیره کرده



در این مثال ما از سایت hashkiller.co.uk استفاده کرده ایم که نتیجه را میبینید.

