Target : crackme2 by massh
Level : 1/10
Lang. : asm32
Tools : OllyDBG, PEiD
Cracker : allko

Learn Reverse Engineering Series- lesson three

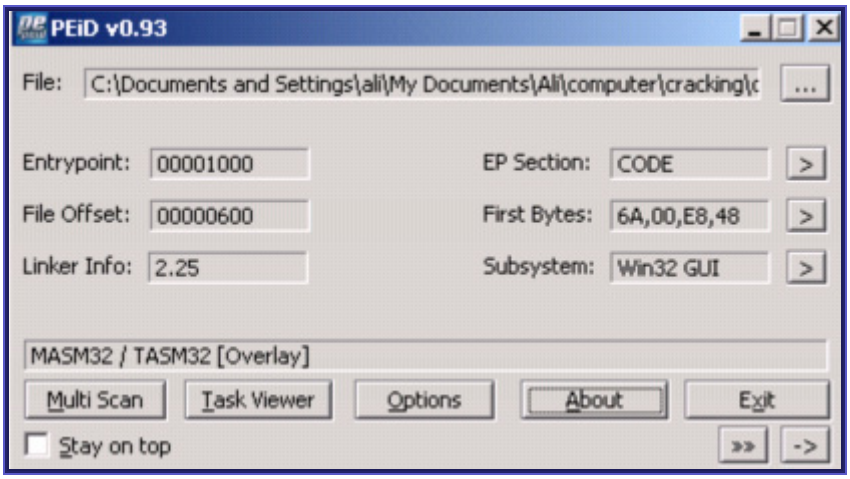
Introduction:

ها قد وصلنا إلى ثالث درس لنا في هذه السلسلة...في الدرسين الماضيين تعلمنا أمورا كثيرة...وها نحن نكمل المشوار مع هذا الدرس...الهدف لهذا الدرس لا يحتوي على خوارزمية تشفير مختلفة عما تعلمناه...اقصد أنها ليست مميزة أو معقدة...لكن الطريق للوصول إليها يجعلنا نمر بعدة أمور مهمة سيتم شرحها ، لذلك فمع نهاية الدرس ستجد نفسك – عزيزي الكراكر – وقد تعلمت العديد من الطرق والأساليب والمفاهيم الجديدة...حظا جيدا.
قبل البدء ،يفضل قراءة الملاحظات بالأسفل في قسم notes .

Here we go :

000 – كالعادة ، شغل الملف الضحية...لا شيء مميز...مكان لكتابة الـ username وآخر للـ serial number وعند إدخال السيريال الخطأ تخرج لنا شاشة مزعجة...

001– قم بعمل فحص للبرنامج باستخدام PEiD 0.9 والنتيجة :



010– والآن حان دور OLLYDBG ...
ابدأ بالبحث – في قسم الـ comments – عن دالة تهمنا...حسنا هناك MessageBoxA على عنوان 40119D كما تلاحظ فان بارامتر text فارغ لكن يوجد قبل الدالة two strings الأولى للسيريال الصحيح والأخرى للخاطئ

```
00401186  .  BB 8A204000   MOV EBX,Crackme2.0040208A        ASCII "Great Work!"
0040118B  .v EB 05         JMP SHORT Crackme2.00401192
0040118D  >  BB 71204000   MOV EBX,Crackme2.00402071        ASCII "Nope, but keep trying :)"
00401192  >  6A 00         PUSH 0                           Style = MB_OK|MB_APPLMODAL
00401194  .  68 68204000   PUSH Crackme2.00402068           Title = "CrackMe2"
00401199  .  53            PUSH EBX                         Text
0040119A  .  FF75 08       PUSH DWORD PTR SS:[EBP+8]        hOwner
0040119D  .  E8 9B020000   CALL <JMP.&USER32.MessageBoxA>   MessageBoxA
```

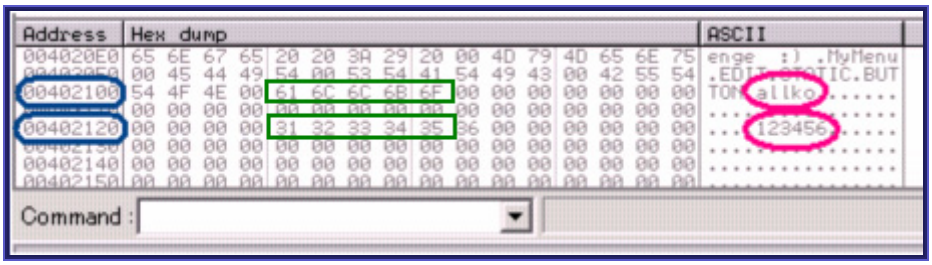
allko

ملاحظة : إن الـ string كما ترى تنقل إلى EBX ومقابل البارامتر text هناك أمر PUSH EBX ...اعتقد أن الصورة أصبحت واضحة.
تابع حتى تصل إلى ما كنا نبحث عنه...GetWindowTextA ...هناك دالتان من هذا النوع الأولى لليوزرنيم والأخرى للباسوورد. هذا مكان مناسب لوضع BreakPoint ضع واحدة عند العنوان 40120F كما بالصورة

```
004011FD  .  6A 1F         PUSH 1F                          Count = 1F (31.)
004011FF  .  68 04214000   PUSH Crackme2.00402104           Buffer = Crackme2.00402104
00401204  .  FF35 4C20400  PUSH DWORD PTR DS:[40204C]       hWnd = NULL
0040120A  .  E8 1C020000   CALL <JMP.&USER32.GetWindowTextA> GetWindowTextA
0040120F  .  85C0          TEST EAX,EAX
00401211  .v 74 4D         JE SHORT Crackme2.00401260
00401213  .  A3 60204000   MOV DWORD PTR DS:[402060],EAX
00401218  .  6A 1F         PUSH 1F                          Count = 1F (31.)
0040121A  .  68 24214000   PUSH Crackme2.00402124           Buffer = Crackme2.00402124
0040121F  .  FF35 5020400  PUSH DWORD PTR DS:[402050]       hWnd = NULL
00401225  .  E8 01020000   CALL <JMP.&USER32.GetWindowTextA> GetWindowTextA
```

allko

هناك أمر TEST EAX,EAX يليه JE وهذان الأمران مترابطان فغالبا ما يكونان معا...والمعنى : " إذا كانت قيمة EAX صفرا فاقفز إلى العنوان..." وذلك يعني انك إذا لم تدخل شيئا في خانة اليوزرنيم أو الباسوورد(لاحظ انه يوجد أمران آخران أسفل دالة السيريال) فانه لن يكمل عملية التحقق بل سيذهب إلى 401260 ولا يهمنا ألان ماذا يوجد هناك...
أين يخزن اليوزرنيم والباسوورد؟ كما علمت من الأمثلة السابقة ، انظر إلى البارامتر buffer لتعرف ان اليوزر يخزن في 402104 ام الباسوورد فيخزن في 402124 ، والان شغل الملف بالضغط على F9 ثم اكتب allko و 123456 أو أي شيء آخر...اضغط OK ها قد عدنا إلى Olly ، تتبع – باستخدام F8 – حتى تصل الاى الامر الذي يلي الدالة الثانية مباشرة , أي إلى العنوان 40122A والان انظر إلى محتويات الذاكرة أي إلى القسم Data



كما تلاحظ فان الاسم خزن في 402104 (لاحظ بجانب 402100 ، اول 4 بايتات ليست للاسم أي ان الاسم يبدا من 402104 وكما تلاحظ هناك الرقم 61 الذي يقابل اول حرف وهو a). وكذلك الحال بالنسبة للسيريال نمبر.
بعد ذلك هناك دالتان من نوع SendMessageA ...ليستا مفيدتنا لنا لذا اتركهما وتتبع حتى تصل إلى أمر RETN انظر إلى الاسفل... يخبرك Olly بانه : Return to 0040116F لا بأس تتبع...نحن الآن عند العنوان المذكور و هناك أمر TEST EAX,EAX يليه JE...

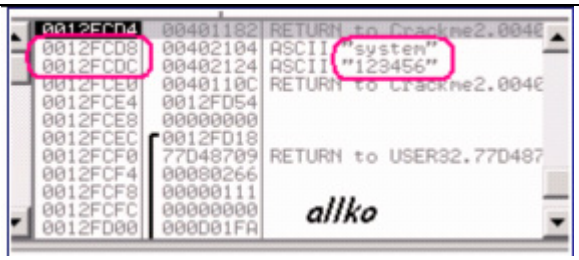
```
0040116F  .  85C0          TEST EAX,EAX
00401171  .v 74 2F         JE SHORT Crackme2.004011A2
00401173  .  68 24214000   PUSH Crackme2.00402124        ASCII "123456"
00401178  .  68 04214000   PUSH Crackme2.00402104        ASCII "allko"
0040117D  .  E8 DF000000   CALL Crackme2.00401261
```

هناك امران PUSH لنقل اليوزر والباسوورد إلى الذاكرة عند العناوين المبينة(لاحظ ان العناوين هي نفسها التي كانت بجانب بارامتر buffer) والان وصلنا إلى call اضغط انتر وضع BreakPoint حتى يتسنى لنا تتبع الاوامر الموجودة بداخل الـ call ...كما ترى هناك loop inside loop ...
011- فلنبدأ بفك التشفير...
لاحظ ان هناك أمر مقارنة مع القيمة 6 لذا ، اعد تشغيل البرنامج واتبع نفس الخطوات السابقة لكن ادخل اسما من 6 احرف وليكن system ونفس كلمة السر 123456 ...

```
00401261  $  A1 60204000   MOV EAX,DWORD PTR DS:[402060]
00401266  .  3905 6420400  CMP DWORD PTR DS:[402064],EAX
0040126C  .v 75 2F         JNZ SHORT Crackme2.0040129D
0040126E  .  33C0          XOR EAX,EAX
00401270  .  8B7C24 04     MOV EDI,DWORD PTR SS:[ESP+4]
00401274  .  8B7424 08     MOV ESI,DWORD PTR SS:[ESP+8]
00401278  >  0FB61F        MOVZX EBX,BYTE PTR DS:[EDI]
0040127B  .  0FB616        MOVZX EDX,BYTE PTR DS:[ESI]
0040127E  .  80FA 30       CMP DL,30
00401281  .v 7C 1A         JL SHORT Crackme2.0040129D
00401283  .  80FA 39       CMP DL,39
00401286  .v 7F 15         JG SHORT Crackme2.0040129D
00401288  .  80EA 30       SUB DL,30
0040128B  .  83E3 0F       AND EBX,0F      allko
0040128E  .  D0EB          SHR BL,1
00401290  .  2ADA          SUB BL,DL
00401292  .v 75 09         JNZ SHORT Crackme2.0040129D
00401294  .  47            INC EDI
00401295  .  46            INC ESI
00401296  .  803F 00       CMP BYTE PTR DS:[EDI],0
00401299  .^ 75 DD         JNZ SHORT Crackme2.00401278
0040129B  .v EB 04         JMP SHORT Crackme2.004012A1
0040129D  >  33C0          XOR EAX,EAX
0040129F  .v EB 05         JMP SHORT Crackme2.004012A6
004012A1  >  B8 01000000   MOV EAX,1
004012A6  >  C3            RETN
```

ينقل طول الاسم (6) من الذاكرة إلى **EAX**
يتم مقارنة طول السيريال مع القيمة 6
اقفز إلى **40129D** إذا لم يكونا متساويين
صفر **EAX**
ضع اليوزرنيم في **EDI** *(ملاحظة 1)*
ضع الباسوورد في **ESI**
انقل اول حرف من اليوزرنيم إلى **EBX** *(ملاحظة 2)*
انقل اول خانة في السيريال(فلنسمها **s1**) إلى **EDX**
قارن **s1** مع 30 (الاعداد في **ASCII** تبدا من 30 حتى39)
اقفز إذا كان **s1** اقل من30(أي انه ليس برقم ولا بحرف)
قارن **s1** مع 39
اقفز إذا كان **s1** اكبر من 39 (أي انه ليس رقما)
اطرح 30 من قيمة الحرف الاول
هذه العملية تبقي الخانة الأولى فقط *(ملاحظة 3)*
اقسم على 2 *(ملاحظة 4)*
اطرح القيمتين من بعض...
اقفز إلى **40129D** إذا لم تكونا متساويتين
زود قيمة **EDI** بمقدار 1 *(ملاحظة 5)*
زود قيمة **ESI** بمقدار 1
قارن **EDI** مع 0 ، أي هل انتهت احرف اليوزرنيم؟
إذا لم تنتهي فاقفز لبداية الـ**loop** للانتقال للحرف التالي
اقفز إلى **4012A1**
صفر **EAX**
اقفز إلى **4012A6**
انقل القيمة 1 إلى **EAX**
عد إلى **401182**

ملاحظة 1 : القيمة [ESP+4] تساوي 12FCD4+4=12FCD8 أي ان SS:[ESP+4] تشير إلى عنوان في الـ stack (SS=Stack Segment) وكما ترى في الاسفل (قسم التعليقات السفلي) فهذا العنوان يحتوي على اليوزرنيم وللتأكد انظر إلى قسم الـ stack لتتأكد ، ونفس الشيء بالنسبة للامر التالي



ملاحظة 2 : لاحظ الامر جيدا MOVZX EBX , BYTE PTR DS:[EDI] هناك كلمة BYTE وحيث ان اليوزرنيم تخزن على هيئة DWORD (انظر 401270)ونعلم ان DWORD=Double Word=32 byte إذا اول بايت يشير إلى اول حرف من اليوزرنيم ، ونفس الشيء بالنسبة للامر التالي (أي بالنسبة للباسوورد).

ملاحظة 3 : عملية AND تعني الضرب وبالتالي عندما نضرب بـ 0F أي 00001111 فان كل شيء سيصبح صفرا ما عدا الخانة الأولى لانها تضرب بـ 1111 وبالتالي تبقى كما هي ، في مثالنا فان EBX يحوي الحرف الاول من كلمة system أي حرف s والذي يقابل القيمة 73 وعند تحويلها لـ Binary فانها تساوي 0111 0011 ، اضربها بـ 1111 0000 ستجد ان الخانة الثانية – 7 – قد اختفت (اصبحت 0) بينما بقيت الخانة الأولى 3 كما هي.

ملاحظة 4 : SHR BL,1 تعني ازاحة محتويات BL لليمين لمرة واحدة...بعد عملية AND اصبح BL=03 أي 0011 وعند عمل الازاحة فان الخانة اليمنى تختفي (تحديدا ، تذهب إلى CF:Carry Flag) بينما يضاف 0 من اليسار. أي ان 0011 تصبح 0001
مثال : مثال : لو انBL=07أي 0111 ثم عملنا SHR BL,1فانه سيصبح 0011
مثال : لو ان BL=06 أي 0110 ثم عملنا SHL BL,2 فانه سيصبح 1000
ومن الامثلة أعلاه يمكن استنتاج الآتي : كل ازاحة لليمين تعني قسمة على 2 وكل ازاحة لليسار تعني ضرب بـ 2 اذاSHR BL,3 يعني قسمة BL على 8 (كل ازاحة لليمين تعني قسمة على 2 اذا 3 ازاحات تعني قسمة على 8) SHL BL,2 يعني ضرب BL في 4 (كل ازاحة لليسار تعني ضرب بـ2 اذا ازاحتان تعني ضرب بـ 4) والباقي في حالة القسمة يضيع.

لاحظت ان كل خانة من خانات السيريال تمت مقارنتها : هل هي اصغر من 30 ؟ هل هي اكبر من 39؟ واذا تحقق احد هذين الشرطين فاننا سنذهب إلى bad boy massage وذلك يعني ان السيريال مكون من ارقام فقط...
ملاحظة : انظر جدول ASCII المصغر المرفق...الارقام تبدا من 30 حتى 39، ال small letters تبدا من 61 حتى 7A بينما capital letters تبدا من 41 حتى 5A
دعنا الآن نفك الشيفر...اول حرف من اليوزرنيم ولنسمه u1 يتم عمل AND له أي انه الخانة الثانية من قيمته تختفي وتبقى الأولى ، ثم يتم عمل SHR له...ام السيريال فيطرح منه 30 ...ثم تتم عملية المقارنة بين القيمتين...من ذلك يمكن وضع الجدول التالي

operation	s	y	s	t	e	m
in ASCII	73	79	73	74	65	6D
AND 0F	3	9	3	4	5	D
In Binary	0011	1001	0011	0100	0101	1101
SHR BL,1	0001	0100	0001	0010	0010	0110
In decimal	1	4	1	2	2	6
ADD 30	31	34	31	32	32	36
In ASCII	1	4	1	2	2	6

إذا الباسوورد الخاصة بكلمة system هي **141226**

Do you know ?

1- هل تعلم أن أول أمين عام للأمم المتحدة هو "ترنجف لي" وهو نرويجي الجنسية.
2- هل تعلم أن العالم نيوتن هو الذي ابتدع علم التفاضل والتكامل.
3- هل تعلم أن القرآن الكريم نزل على مرحلتين: الأولى في ليلة القدر حيث نزل إلى السماء الدنيا والثانية على مدى 23 سنة حيث كان يتنزل على محمد – صلى الله عليه وسلم -.

Notes :

1 – you can add an arrows to show the direction of the jumps in OLLY :
Options ➔ Debugging options ➔ CPU ➔ mark these :
Show direction of jumps , Show jump path
Show grayed path if jump is not taken and Show jumps to selected commands
2 – If you want to restart OLLY quickly: CTRL+F2
3- open PEiD : options ➔ register shell extensions
Now you can easily right click the crackme and select : scan with PEiD
4- If you are not familiar with the number systems (binary , hex , decimal , etc…) feel free to mail me and I will help you as much as I can , nevertheless you are not obligated to convert between hex and decimal by paper and pen!! , just use the calculator companioned with windows and switch to" scientific mode".

Outroduction :

I'm still a newbie; I looked Longley for an easy crackme, though all of them are "level 1" crackmes. okay that's not too bad because this is just the beginning . My study began last week so I might not complete this series regularly although I'd like to. If you have any comments, problems, suggestions or you just want to say "hello cracker" feel free to mail me: allko34~hotmail.com

Greetings :

Forums: Coding and protection forum, Arab team's coding forum
Extra greetngz : DiDi