

BÁO CÁO ĐỒ ÁN ĐỒ ÁN 3 – CRACK PHẦN MỀM

KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

Giáo viên hướng dẫn thực hành: Nguyễn Thanh Quân

Giáo viên lý thuyết: Chung Thùy Linh

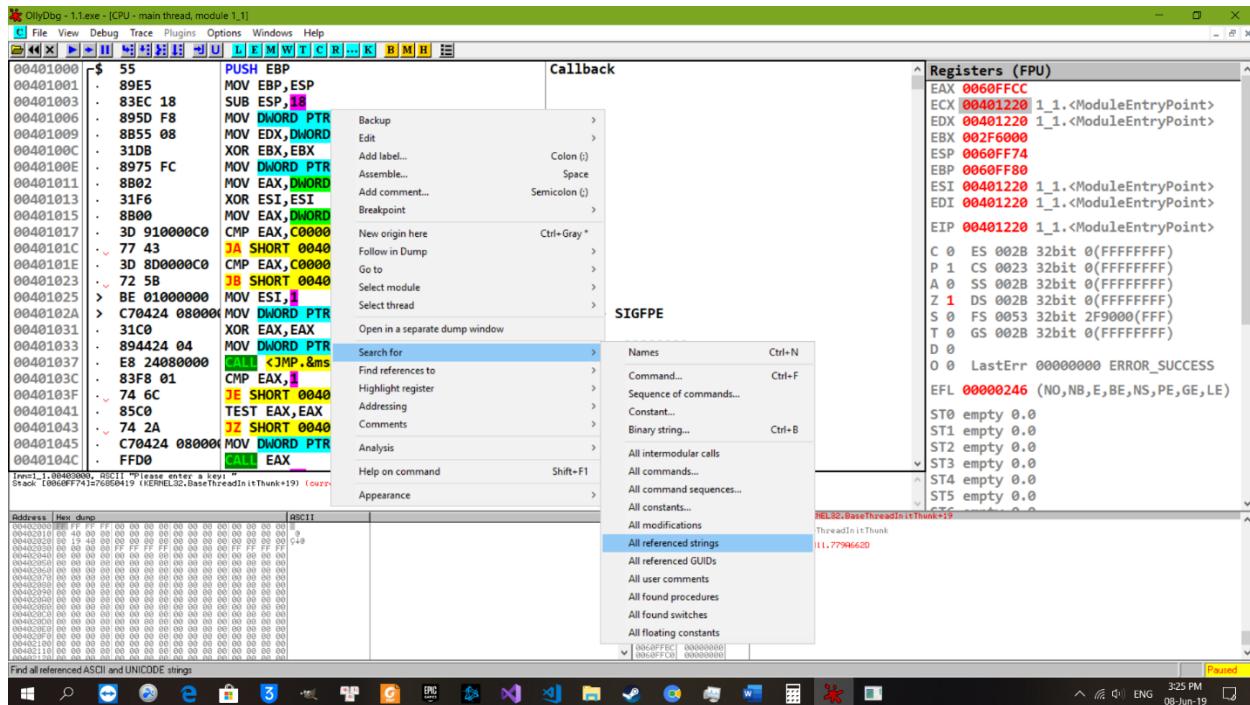
Nhóm: 1712358 – 1712369 – 1712379

Đề 01

Phân công công việc

Họ tên	MSSV	Đảm nhận
Nguyễn Minh Đức	1712358	1.3
Phạm Quốc Dũng	1712369	1.4 1.5
Đặng Thành Duy	1712379	1.1 1.2

1.1



OlyDbg - 1.1.exe - [Search - Text strings referenced in 1.1]

Address	Command	Comments
00401302	MOV DWORD PTR SS:[ESP],OFFSET 00403000	ASCII "Please enter a key: "
00401316	MOV DWORD PTR SS:[ESP],OFFSET 00403015	ASCII "%d"
00401330	MOV DWORD PTR SS:[ESP],OFFSET 00403018	ASCII "Congratulations! You are successful."
0040133E	MOV DWORD PTR SS:[ESP],OFFSET 00403040	ASCII "Better luck next time! You are unsuccessful."
00401517	MOV ECX,OFFSET 00403094	ASCII "w32_shreadptr->size == sizeof(W32_EH_SHARED)"
00401529	MOV DWORD PTR SS:[ESP],OFFSET 004030C1	ASCII "%s:%u: failed assertion `\\s'"
00401530	MOV EAX,OFFSET 004030E0	ASCII "./../gcc/gcc/config/i386/w32-shared-ptr.c"
0040153E	MOV EAX,OFFSET 0040310C	ASCII "GetAtomNameA (atom, s, sizeof(s)) != 0"

Found 8 strings and references:
Module <Mod_7793> (anonymous)

Windows Taskbar:

- File
- View
- Debug
- Trace
- Plugins
- Options
- Windows
- Help
- Pause
- Minimize
- Maximize
- Close
- 3:26 PM
- ENG
- 08-Jun-19

OlyDbg - 1.1.exe - [CPU - main thread, module 1.1]

Address	Command	Comments	Registers (FPU)
004012E6	. 83C0 0F ADD EAX,BF		EAX 0000FFCC
004012E9	. 83C0 0F ADD EAX,BF		ECX 00401220 1_1.<ModuleEntryPoint>
004012EC	. C1E8 04 SHR EAX,4		EDX 00401220 1_1.<ModuleEntryPoint>
004012EF	. C1E0 04 SHL EAX,4		EBX 002F6000
004012F2	. 8945 FC MOV DWORD PTR SS:[LOCAL.1],EAX		ESP 0060FF74
004012F5	. 8B45 FC MOV EAX,DWORD PTR SS:[LOCAL.1]		EBP 0060FF80
004012F8	. E8 A3040000 CALL 004017A0	Allocates 16. bytes on stack	ESI 00401220 1_1.<ModuleEntryPoint>
004012FD	. E8 3E010000 CALL 00401440	[1_1.00401440]	EDI 00401220 1_1.<ModuleEntryPoint>
00401302	. C70424 003040 MOV DWORD PTR SS:[LOCAL.10],OFFSET 0040:	Format => "Please enter a key: "	EIP 00401220 1_1.<ModuleEntryPoint>
00401309	. E8 A2050000 CALL <JMP.&msvcrt.printf>	MSVCRT.printf	C 0 ES 0028 32bit 0(FFFFFFFF)
0040130E	. C74424 04 104 MOV DWORD PTR SS:[LOCAL.9],OFFSET 004040:	[%d] => 1_1.004010 -> 0	P 1 CS 0023 32bit 0(FFFFFFFF)
00401316	. C70424 153040 MOV DWORD PTR SS:[LOCAL.10],OFFSET 0040:	format => "%d"	A 0 SS 0028 32bit 0(FFFFFFFF)
0040131D	. E8 7E050000 CALL <JMP.&msvcrt.scantf>	MSVCRT.scantf	Z 1 DS 0028 32bit 0(FFFFFFFF)
00401322	. E8 60FFFFF CALL 00401290	[1_1.00401290]	S 0 FS 0053 32bit 2F9000(F)
00401327	. E8 833D 10404000 CMP DWORD PTR DS:[004010],1		T 0 GS 0028 32bit 0(FFFFFFFF)
0040132E	. v 75 0E JNE SHORT 0040133E		D 0 LastErr 00000000 ERROR_SUCCESS
00401330	. C70424 183040 MOV DWORD PTR SS:[LOCAL.10],OFFSET 0040:	format => "Congratulations! You are successful!"	EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
00401337	. E8 74050000 CALL <JMP.&msvcrt.printf>	MSVCRT.printf	ST0 empty 0.0
0040133C	. EB 0C JMP SHORT 004013AA		ST1 empty 0.0
> 0040133E	> C70424 403040 MOV DWORD PTR SS:[LOCAL.10],OFFSET 0040:	format => "Better luck next time! You are unsuccessful."	ST2 empty 0.0
00401343	. E8 66050000 CALL <JMP.&msvcrt.printf>	MSVCRT.printf	ST3 empty 0.0
0040134A	. E8 D1040000 CALL <JMP.&msvcrt._getch>	MSVCRT._getch	ST4 empty 0.0
0040134F	. BB 00000000 MOV EAX,B		ST5 empty 0.0
00401354	. C9 LEAVE		ST6 empty 0.0
00401355	. C3 RETN		ST7 empty 0.0

MSVCRT._scanf returned ERR = 69569498.

Dest1_1.00401290

Windows Taskbar:

- File
- View
- Debug
- Trace
- Plugins
- Options
- Windows
- Help
- Pause
- Minimize
- Maximize
- Close
- 3:26 PM
- ENG
- 08-Jun-19

OlyDbg - 1.1.exe - [CPU - main thread, module 1_1]

Registers (FPU)

```

EAX 0000FFCC
ECX 00401220 1_1.<ModuleEntryPoint>
EDX 00401220 1_1.<ModuleEntryPoint>
EBX 002F6000
ESP 0060FF74
EBP 0060FF80
ESI 00401220 1_1.<ModuleEntryPoint>
EDI 00401220 1_1.<ModuleEntryPoint>
EIP 00401220 1_1.<ModuleEntryPoint>
C 0 ES 002B 32bit 0(FFFFFFFF)
P 1 CS 0023 32bit 0(FFFFFF)
A 0 SS 002B 32bit 0(FFFFFF)
Z 1 DS 002B 32bit 0(FFFFFF)
S 0 FS 0053 32bit 2F9000(F)
T 0 GS 002B 32bit 0(FFFFFF)
D 0
O 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0

```

Stack (0060FF7C)=76050400 (KERNEL32.BaseThreadInitInThunk) (current registers)

Module <Mod_77D> (anonymous)

OlyDbg - 1.1.exe - [CPU - main thread, module 1_1]

Registers (FPU)

```

EAX 0000FFCC
ECX 00401220 1_1.<ModuleEntryPoint>
EDX 00401220 1_1.<ModuleEntryPoint>
EBX 002F6000
ESP 0060FF74
EBP 0060FF80
ESI 00401220 1_1.<ModuleEntryPoint>
EDI 00401220 1_1.<ModuleEntryPoint>
EIP 00401220 1_1.<ModuleEntryPoint>
C 0 ES 002B 32bit 0(FFFFFFFF)
P 1 CS 0023 32bit 0(FFFFFF)
A 0 SS 002B 32bit 0(FFFFFF)
Z 1 DS 002B 32bit 0(FFFFFF)
S 0 FS 0053 32bit 2F9000(F)
T 0 GS 002B 32bit 0(FFFFFF)
D 0
O 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0

```

Stack (0060FF7C)=76050400 (KERNEL32.BaseThreadInitInThunk) (current registers)

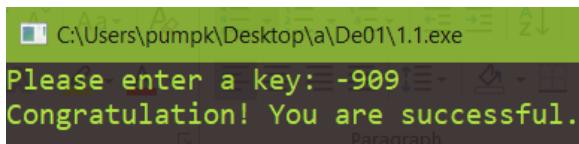
Module <Mod_77D> (anonymous)

Đoạn phát sinh key:

```
MOV DWORD PTR SS:[LOCAL.1],0C8
MOV EDX,DWORD PTR SS:[LOCAL.1]
MOV EAX,EDX
SHL EAX,2
ADD EAX,EDX
MOV DWORD PTR SS:[EBP-4],EAX
LEA EAX,[EBP-4]
XOR DWORD PTR DS:[EAX],00000064
LEA EAX,[EBP-4]
NOT DWORD PTR DS:[EAX]
```

*[LOCAL.1] = 200
EDX = *[LOCAL.1]
EAX = EDX
EAX <= 2 = 800
EAX += EDX = 1000
*[EBP-4] = EAX
[EAX] = [EBP-4]
*[EAX] ^= 0x64 = 908
[EAX] = [EBP-4]
[EAX] = ~([EAX]) = -909

Ý nghĩa:



1.2

DLL Debug - 1.2.exe - [CPU - main thread, module 1_2]

Registers (FPU)

EAX	00401000	1_2.<ModuleEntryPoint>
ECX	00401000	1_2.<ModuleEntryPoint>
EDX	00401000	1_2.<ModuleEntryPoint>
EBX	0022C000	
ESP	0019FF74	
EBP	0019FF80	
ESI	00401000	1_2.<ModuleEntryPoint>
EDI	00401000	1_2.<ModuleEntryPoint>
EIP	00401000	1_2.<ModuleEntryPoint>
C	0	ES 002B 32bit 0(FFFFFFFF)
P	1	CS 0023 32bit 0(FFFFFFFF)
A	0	SS 002B 32bit 0(FFFFFFFF)
Z	1	DS 002B 32bit 0(FFFFFFFF)
S	0	FS 0053 32bit 22F000(FFFF)
T	0	GS 002B 32bit 0(FFFFFFFF)
D	0	
O	0	LastErr 00000000 ERROR_SUCCESS
EFL	00000246	(NO_NB,E,BE,NS,PE,GE,LE)
ST0	empty 0.0	
ST1	empty 0.0	
ST2	empty 0.0	
ST3	empty 0.0	
ST4	empty 0.0	
ST5	empty 0.0	
ST6	empty 0.0	

Module Name = NULL

InitCommonControls

- .4010AF
- 5
- ParamA
- sed void)

Search for

- Names Ctrl+N
- Find references to
- Highlight register
- Addressing
- Comments
- Binary string... Ctrl+B
- Analysis
- Help on command Shift+F1
- Appearance

Names

- All intermodular calls
- All commands...
- All command sequences...
- All constants...
- All modifications
- All referenced strings
- All referenced GUIDs
- All user comments
- All found procedures
- All found switches
- All floating constants

Address Hex dump

Address

Call 1.2

Entry point of main module

OllyDbg - 1.2.exe - [CPU - main thread, module 1_2.]

File View Debug Trace Plugins Options Windows Help

Registers (FPU)

1_2.0040109B(guessed void)

```

0040109B $ 83C3 4C ADD EBX,4C
0040109E . 83C2 03 ADD EDX,3
004010A1 . 43 INC EBX
004010A2 . 81C3 8803000 ADD EBX,38B
004010A8 . 03DB ADD EBX,EBX
004010A4 . 0FAFDA IMUL EBX,EDX
004010AD . 4B DEC EBX
004010AE C3 RETN
004010AF $ 55 PUSH EBP
004010B0 . 8BEC MOV EBP,ESP
004010B2 . 8B45 0C MOV EAX,WORD PTR SS:[ARG.2]
004010B5 . 3D 10010000 CMP EAX,110
004010B8 . 75 18 JNE SHORT 004010D7
004010BC . 6A 00 PUSH 0
004010BE . 6A 04 PUSH 4
004010C0 . 68 C5000000 PUSH 0C5
004010C5 . 68 EC030000 PUSH 3EC
004010CA . FF75 08 PUSH DWORD PTR SS:[ARG.1]
004010CD . E8 44010000 CALL < JMP.&User32.SendDlgItemMessageA>
004010D2 . E9 EB000000 JMP 004011C2
004010D7 > 3D 11010000 CMP EAX,110
004010D8 . 0F85 D100000 JNE 004011B3
004010E2 . 8B45 10 MOV EAX,WORD PTR SS:[ARG.3]
004010E5 . 3D EA030000 CMP EAX,3EA
004010E8 . 75 19 JNE SHORT 00401105

```

lParam = 0
wParam = 4
Msg = EM_LISTTEXT
ItemID = 1004.
hDialog => [ARG.1]
USER32.SendDlgItemMessageA

Registers (FPU)

EAX 0019FFCC
ECX 00401000 1_2.<ModuleEntryPoint>
EDX 00401000 1_2.<ModuleEntryPoint>
EBX 0022C000
ESP 0019FF74
EBP 0019FF80
ESI 00401000 1_2.<ModuleEntryPoint>
EDI 00401000 1_2.<ModuleEntryPoint>
EIP 00401000 1_2.<ModuleEntryPoint>

C 0 ES 002B 32bit 0(FFFFFFFF)
P 1 CS 0023 32bit 0(FFFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFFF)
Z 1 DS 002B 32bit 0(FFFFFFFF)
S 0 FS 0053 32bit 22F000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0 O 0 LastErr 00000000 ERROR_SUCCESS
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0

Registers (FPU)

0019FF74 76898919 RETURN to KERNEL32.BaseThreadInitchunk+19
0019FF7C 00000000 KERNEL32.BaseThreadInitchunk
0019FF80 0019FF7C RETURN to nt!dll.77994620

Address Hex dump ASCII

Entry point of main module

Paused 5:36 PM 08-Jun-19

Đoạn phát sinh key:

CALL <JMP.&user32.GetDlgItemInt>
MOV EBX,EAX

CALL <JMP.&user32.GetDlgItemInt>
MOV ECX,EAX

ADD EBX,4C
ADD EDX,3
INC EBX
ADD EBX,38B
ADD EBX,EBX
IMUL EBX,EDX
DEC EBX

Ý nghĩa:

EDX = 1701722 + [strlen(personal ID)-1]*2

EAX = personal ID

EBX = EAX

EDX = 1701722 + [strlen(Serial)-1]*2

EAX = Serial

ECX = EAX

EBX += 76

EDX += 3

EBX += 1

EBX += 907

EBX += EBX

EBX *= EDX

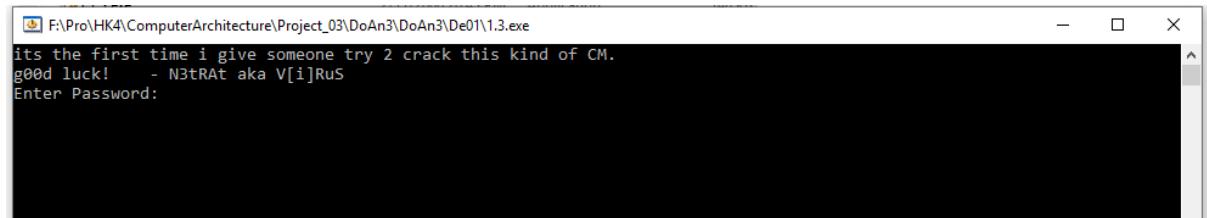
EBX -= 1



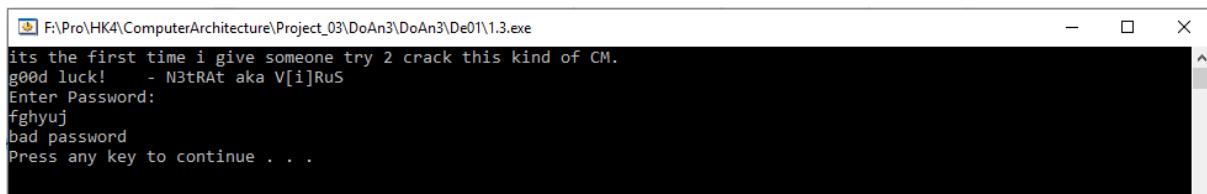
1.3

Khởi đầu

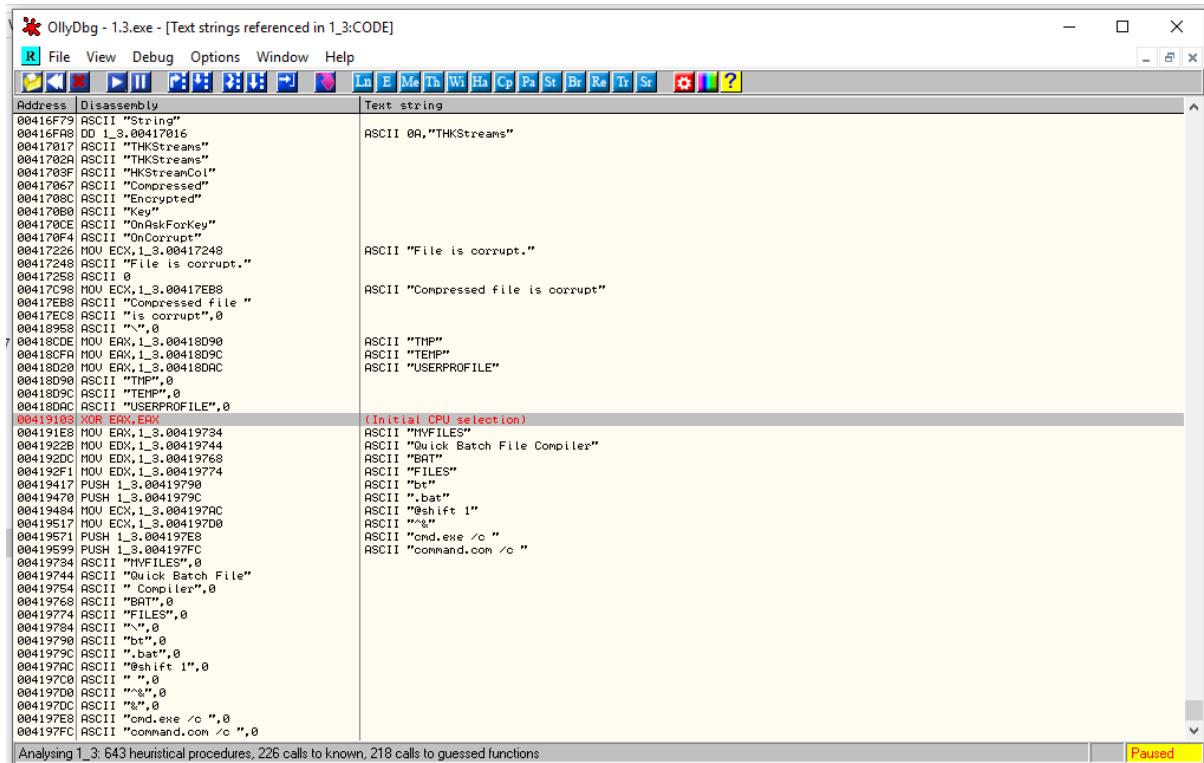
Khi chạy file 1.3.exe, ta được kết quả như sau:



Thử nhập 1 password ngẫu nhiên, được kết quả là:



Như đã thấy, console màn hình hiện ra 1 vài chuỗi có nghĩa, ta có thể tìm các chuỗi này trong source string khi phân tích 1.3.exe bằng Olly Dbg, từ đó tìm manh mối dẫn tới phần thuật toán phát sinh password:



Bằng cách chuột phải -> search for -> all referenced text strings, thử tìm tất các chuỗi nhưng không có chuỗi nào giống trong console mà thay vào đó là các chuỗi liên quan tới file. Đặc biệt xuất hiện chuỗi "cmd.exe /c", chuỗi này chuyên dùng để chạy command line thực thi các file.

Khi debug, em cố gắng dò tìm trong stack chương trình, cuối cùng tìm được nhiều chuỗi chứa 1 đường dẫn rất khả nghi:

0019FDFC	004196ED	SE handler
0019FE00	0019FF70	
0019FE04	0019FF64	Pointer to next SEH record
0019FE08	0041971F	SE handler
0019FE0C	0019FF70	
0019FE10	002BD000	
0019FE14	00000000	
0019FE18	00000000	
0019FE1C	00000000	
0019FE20	021E0A70	ASCII "C:\Users\Admin\AppData\Local\Temp\"
0019FE24	3A462245	
0019FE28	6F72505C	
0019FE2C	344B485C	
0019FE30	6D6F435C	
0019FE34	65747570	
0019FE38	63724172	
0019FF20	00000000	
0019FF24	021E1510	ASCII ""F:\Pro\HK4\ComputerArchitecture\Project_03\DoAn3\DoAn3\De01\1.3.exe""
0019FF28	021E14D4	ASCII "C:\Users\Admin\AppData\Local\Temp\bt5862.bat"
0019FF2C	021E1498	ASCII "C:\Users\Admin\AppData\Local\Temp\bt5862.bat"
0019FF30	021E1458	
0019FF34	A21F144A	

Quan sát trong đường dẫn "C:\Users\Admin\AppData\Local\Temp", đúng là trong quá trình chạy 1.3.exe có phát 1 file "bt5862.bat":

This PC > WINDOWS 10 (C:) > Users > Admin > AppData > Local > Temp				
	Name	Date modified	Type	Size
Mozilla	assistant_installer_2019060230325.log	6/6/2019 23:03 PM	Text Document	1 KB
MSfree Inc	assistant_installer_20190607230325.log	7/6/2019 23:03 PM	Text Document	1 KB
NCSOFT	assistant_installer_20190607230338.log	7/6/2019 23:03 PM	Text Document	2 KB
Notepad++	assistant_installer_20190607230339.log	7/6/2019 23:03 PM	Text Document	1 KB
Nox	BIT5751.tmp	1/6/2019 17:36 PM	TMP File	0 KB
NVIDIA	bt3441.bat	8/6/2019 20:05 PM	Windows Batch File	1 KB
NVIDIA Corporation	chrome_installer.log	6/6/2019 18:09 PM	Text Document	17 KB
oald8	C:\Users\Admin\AppData\Local\Programs\Op...	5/6/2019 23:03 PM	LOCK File	0 KB
Opera Software	dd_vs_setup_bootstrapper_decompressio...	4/6/2019 16:46 PM	Text Document	1 KB
Package Cache	ida16717.tmp	6/6/2019 21:32 PM	TMP File	3 KB

Khi chạy file .bat này, máy tính thực thi 1 chương trình console giống y như chạy 1.3.exe:

```

C:\WINDOWS\system32\cmd.exe
its the first time i give someone try 2 crack this kind of CM.
g00d luck! - N3tRAT aka V[i]RuS
Enter Password:
fd5afadss
bad password
Press any key to continue . .

```

Em đoán là 1.3.exe đã tự tạo file .bat này, sau đó thực thi nó, để file .bat làm toàn bộ công việc, còn 1.3.exe chỉ việc tạo file và chạy file. Để chứng thực cho phán đoán trên, em tìm lại trong các dòng lệnh của 1.3.exe trong Olly thì đã tìm được nơi 1.3.exe tạo chuỗi đường dẫn tạo file btxxxx.bat:



Ở trên ta thấy 1 di chuyển giá trị từ EBP-44 về thanh ghi EDX, nãy coi trong EBP-44 có gì:

\$-54	00000000
\$-50	00000000
\$-4C	02181510 ASCII "F:\Pro\HK4\ComputerArchitecture\Project_03\DoAn3\DoAn3\De01\1.3.exe"
\$-48	021814D4 ASCII "C:\Users\Admin\AppData\Local\Temp\bt6780.bat"
\$-44	02181498 ASCII "C:\Users\Admin\AppData\Local\Temp\bt6780.bat"
\$-40	02181458
\$-3C	02181448
\$-38	02181438
\$-34	02181428

Đúng là chứa chuỗi chỉ đường dẫn tạo file .bat.

Sau đó 1 lệnh mov và 1 lệnh gọi hàm quan trọng được thực hiện:

```

0041E8D1: . 6A 00      PUSH 0
0041E8D3: . 6A 00      PUSH 0
0041E8D5: . A1 00E84100 MOV EAX,DWORD PTR [41E800]
0041E8D9: . E8 10B1FEFF CALL 1_3.004046FC
0041E8DF: . 50          PUSH EAX
0041E8E0: . 6A 00      PUSH 0
0041E8E2: . E8 49CAFEEF CALL <JMP.&kernel32.CreateProcessA>
0041E8E7: . 85C0        TEST EAX,EAX

```

Ở đây giá trị trong địa chỉ 41E8D0 được gán vào EAX, hãy coi trong địa chỉ chứa giá trị gì:

Address	Hex dump	ASCII
0041E8D0	B8 15 05 02 7C 0A 05 02!...
0041E8D8	7C 0A 05 02 68 14 05 02	!...h...
0041E8E0	AC 0A 05 02 00 00 00 00
0041E8E8	00 00 00 00 00 00 00 00
0041E8F0	00 00 00 00 00 00 00 00

Đọc ngược lại ta được 020515B8, dò trong stack, đây là địa chỉ 1 chuỗi chứa 1 lệnh thực thi:

\$-2C	00000000	00000000
\$-28	020515B8	ASCII "cmd.exe > C:\Users\Admin\AppData\Local\Temp\bt8301.bat "F:\Pro\HK4\ComputerArchitecture\Project_03\DoAn3\De01\1.3.exe""
\$-24	00000000	
\$-20	00000000	
\$-16	00000000	

Sau dòng lệnh mov quan trọng này chương trình thực thi hàm CreateProcessA để tạo command line.

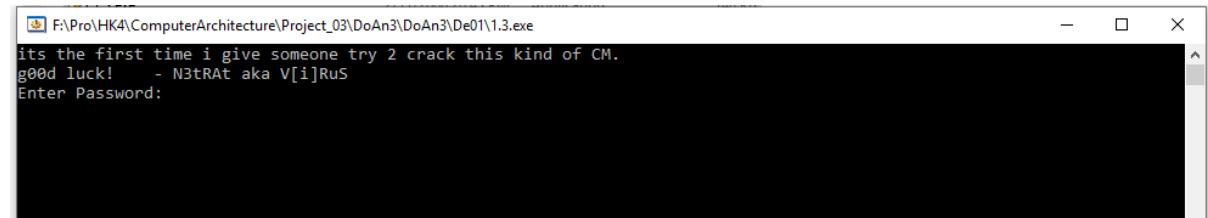
Sau đó chương trình chạy 1 hàm mang tên WaitForSingleObject

```

0041E8E1: . B8 FF      PUSH -1
0041E8E5: . A1 C0E84100 MOV EAX,DWORD PTR [41E800]
0041E8E9: . 50          PUSH EAX
0041E8F3: . E8 98CAFEEF CALL <JMP.&kernel32.WaitForSingleObject>
0041E8F8: . E8 E8E84100 PUSH 1_3.0041E8E8
0041E8FD: . A1 C0E84100 MOV EAX,DWORD PTR [41E800]
0041E902: . 50          PUSH EAX

```

Sau đó thì 1 cửa sổ console hiện lên cho user nhập password, vậy hàm này đã đang chờ file .bat thực thi nốt phần việc còn lại:



Kết luận:

1.3.exe tạo ra 1 file thực thi tại thư mục /Local/Temp với định dạng btxxxx.bat và sau đó dùng command line thực thi file này, chờ file .bat thực thi xong sẽ kết thúc, file .bat chờ người dùng nhập password và thực thi 1 chương trình kiểm tra mật khẩu rất đơn giản.

Keygen:

Ta dùng lại code trong file .bat và dùng lệnh echo để in key ra màn hình bằng command line. Mở command và dán đoạn code sau rồi enter:

set r=o

set o=t

set llo=he

```

set t=y
set h=u
set j=w
set he=llo
echo %o%%llo%%he%%h%%t%%windir%billgates..2006

```

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.503]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Admin>set r=o
ine
leNaC:\Users\Admin>set o=t
oces
C:\Users\Admin>set llo=he
= IN
=> NC:\Users\Admin>set t=y
ins
= =
C:\Users\Admin>set h=u
=>
odeP
C:\Users\Admin>set j=w
=>
dle
C:\Users\Admin>set he=llo

C:\Users\Admin>echo %o%%llo%%he%%h%%t%%windir%billgates..2006
thellouyC:\WINDOWSbillgates..2006
C:\Users\Admin>

```

Vậy key là thellouyC:\WINDOWSbillgates..2006, với phần windir thì tùy máy sẽ khác nhau, nhập key này vào 1.3.exe ta được kết quả:

```

1.2.exe
30/4/2011 19:12 PM Application 118 KB

F:\Pro\HK4\ComputerArchitecture\Project_03\DoAn3\DoAn3\De01\1.3.exe

D:) its the first time i give someone try 2 crack this kind of CM.
) g00d luck! - N3tRAT aka V[i]RuS
Enter Password:
) thellouyC:\WINDOWSbillgates..2006
good password
Press any key to continue . . .

```

Vậy chúng ta đã crack thành công.

1.4

Thuật toán kiểm tra:

B1: Đảo ngược chuỗi username

B2: Gọi A là chuỗi username bị đảo ngược

B3: Vòng lặp để tính $0 - (A_0 - 20) - (A_1 - 20) - \dots - (A_n - 20)$ với n là độ dài chuỗi, sau đó lưu địa chỉ chuỗi kq trong ECX

B4: Đảo ngược chuỗi serial người dùng nhập vào, gán địa chỉ của nó vào EDX

B5: Gán EAX là 4 byte đầu của EDX

B6: So sánh thanh ghi AL với byte đầu của ECX, nếu không = nhau thì xuất ra password không đúng

B7: Xét kết quả phép OR của AL và AL, nếu = 0 thì xuất ra password đúng

B8: So sánh thanh ghi AH với byte thứ hai của ECX, nếu không = nhau thì xuất ra password không đúng

B9: Xét kết quả phép OR của AH và AH, nếu = 0 thì xuất ra password đúng

B10: Shift Right EAX đi 10 bit

B11: So sánh thanh ghi AL với byte thứ ba của ECX, nếu không = nhau thì xuất ra password không đúng

B12: Xét kết quả phép OR của AL và AL, nếu = 0 thì xuất ra password đúng

B13: So sánh thanh ghi AH với byte thứ tư của ECX, nếu không = nhau thì xuất ra password không đúng

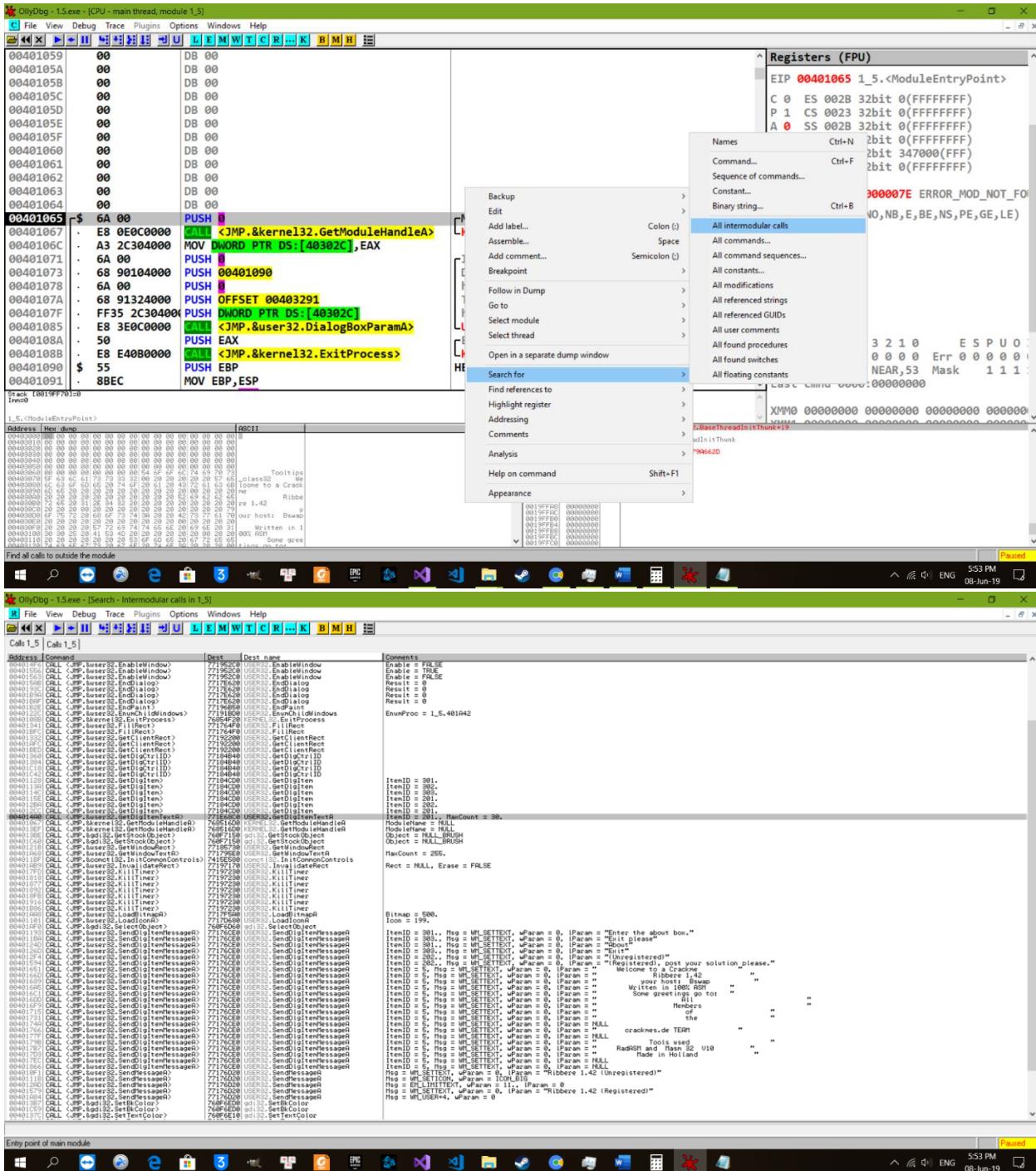
B14: Xét kết quả phép OR của AH và AH, nếu = 0 thì xuất ra password đúng

B15: Tăng ECX lên 4 đơn vị, mục đích là để xét 4 byte tiếp theo trong chuỗi A

B16: Tăng EDX lên 4 đơn vị, mục đích là để xét 4 byte tiếp theo trong chuỗi password bị đảo ngược

B17: Quay lại B5

1.5



OllDbg - 1.exe - [CPU - main thread, module 1_5]

File View Debug Trace Plugins Options Windows Help

Registers (FPU)

EIP 00401065 1_5.<ModuleEntryPoint>

C 0 ES 0028 32bit 0xFFFFFFF
P 1 CS 0023 32bit 0xFFFFFFF
A 0 SS 0028 32bit 0xFFFFFFF
Z 1 DS 0028 32bit 0xFFFFFFF
S 0 FS 0053 32bit 347000(F)
T 0 GS 002B 32bit 0xFFFFFFF
D 0
O 0 LastErr 0000007E ERROR_MOD_NOT_F
EFL 00000246 (NO_NB,E_BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

3 2 1 0 E S P U O
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR,53 Mask 1 1 1
Last cmd 0000:00000000

MaxCount = 30.
String
ItemID = 201.
hDialog = NULL

User32.GetDlgItemTextA

0040147A ADD ECX, EBX
0040147C 66:0FACC2 03 SHRD DX, AX,**3**
00401481 FEC2 INC DL
00401483 32D6 XOR DL,DH
00401485 88143B MOV BYTE PTR DS:[EDI+EBX],DL
00401488 . 43 INC EBX
00401489 . 83FB 04 CMP EBX, 4
0040148E . ^ 75 E5 JNE SHORT 00401473
00401490 . 6A 1E PUSH 1E
00401490 . 68 1C334000 PUSH OFFSET 0040331C
00401495 . 68 C9000000 PUSH 0C9
0040149A . FF35 00304000 PUSH DWORD PTR DS:[403000]
0040149B E8 59080000 CALL C3MP.&user32.GetDlgItemTextA
004014A5 . 83F8 01 CMP EAX, 1
004014A8 . 7D 56 JGE SHORT 00401500
004014A9 . ^ EB 2F JMP SHORT 004014DB
004014AC . ^ EB 25 JMP SHORT 004014D3
004014AE . 59 6F 75 72 ASCII "Your name must b"
004014BE . 65 20 61 74 ASCII "e at least one b"
004014CE . 79 74 65 21 ASCII "yle!"
004014D3 . > EB 06 JMP SHORT 004014DB
004014D5 . > 45 72 72 6F ASCII "Error"
004014DB . > 6A 40 PUSH 40
004014D6 . 68 D5144000 PUSH 004014D5
004014D7 . 68 AE144000 PUSH 004014AE

ASCII "Your name must be at least one byte!"
ASCII "Error"
ASCII "Error"
ASCII "Error"
ASCII "Error"!
Destination: 004014A0 - Jumps to User32.GetDlgItemTextA

0040150E ADD ECX, EBX
0040150F 6A 40 PUSH 40
00401510 . 68 D5144000 PUSH 004014D5
00401511 . 68 AE144000 PUSH 004014AE
00401512 . 6A 00 PUSH 1
00401513 . E8 3A080000 CALL C3MP.&user32.MessageBoxA
00401514 . 6A 00 PUSH 1
00401515 . FF35 60304000 PUSH DWORD PTR DS:[403060]
00401516 . E8 D3070000 CALL C3MP.&user32.EnableWindow
00401517 . 00 99000000 JMP 00401599
00401518 . > 50 PUSH EAX
00401519 . E8 55040000 CALL 00401958
00401520 . A1 3D334000 MOV EAX, DWord PTR DS:[403330]
00401521 . 8B1D 41334000 MOV EBX, DWord PTR DS:[403341]
00401522 . A3 E4324000 MOV DWord PTR DS:[4032E4],EAX
00401523 . 891D E8324000 MOV DWord PTR DS:[4032E8],EBX
00401524 . E8 C0040000 CALL 004019E1
00401525 . 33C0 XOR EAX,EAX
00401526 . 33DB XOR EBX,EBX
00401527 . 66:A1 3B3340 MOV AX,WORD PTR DS:[403338]
00401528 . 66:8B5F 08 MOV BX,WORD PTR DS:[EDI+8]
00401529 . 66:2BC3 SUB AX,BX
00401530 . 35 F1B00000 XOR EAX,00001B3F
00401531 . 2D 23010000 SUB EAX,123
00401532 . EB 03 JMP SHORT 00401514
00401533 . 01 DB 01

ASCII "Error"
ASCII "Your name must be at least one byte!"
Enable = FALSE
hWnd = NULL
User32.EnableWindow

0040153E ADD ECX, EBX
Stack: 00401503 (current registers)
Stack: 00401503 (current registers)
Jump from 403060

Registers (FPU)

EIP 00401065 1_5.<ModuleEntryPoint>

C 0 ES 0028 32bit 0xFFFFFFF
P 1 CS 0023 32bit 0xFFFFFFF
A 0 SS 0028 32bit 0xFFFFFFF
Z 1 DS 0028 32bit 0xFFFFFFF
S 0 FS 0053 32bit 347000(F)
T 0 GS 002B 32bit 0xFFFFFFF
D 0
O 0 LastErr 0000007E ERROR_MOD_NOT_F
EFL 00000246 (NO_NB,E_BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

3 2 1 0 E S P U O
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR,53 Mask 1 1 1
Last cmd 0000:00000000

MaxCount = 30.
String
ItemID = 201.
hDialog = NULL

User32.GetDlgItemTextA

0040153E ADD ECX, EBX
Stack: 00401503 (current registers)
Stack: 00401503 (current registers)
Jump from 403060

Registers (FPU)

EIP 00401065 1_5.<ModuleEntryPoint>

C 0 ES 0028 32bit 0xFFFFFFF
P 1 CS 0023 32bit 0xFFFFFFF
A 0 SS 0028 32bit 0xFFFFFFF
Z 1 DS 0028 32bit 0xFFFFFFF
S 0 FS 0053 32bit 347000(F)
T 0 GS 002B 32bit 0xFFFFFFF
D 0
O 0 LastErr 0000007E ERROR_MOD_NOT_F
EFL 00000246 (NO_NB,E_BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0

3 2 1 0 E S P U O
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 FCW 027F Prec NEAR,53 Mask 1 1 1
Last cmd 0000:00000000

MaxCount = 30.
String
ItemID = 201.
hDialog = NULL

User32.GetDlgItemTextA

DLLMainCRT01()

```

0040195B $ 55 PUSH EBP
0040195C . 8BEC MOV EBP,ESP
0040195E . BE 1C334000 MOV ESI,OFFSET 0040331C
00401963 . BF 3B334000 MOV EDI,OFFSET 0040333B
00401968 . B9 10000000 MOV ECX,10
0040196D > 0FB606 -MOVZX EAX,BYTE PTR DS:[ESI]
00401970 . 51 PUSH ECX
00401971 . 50 PUSH EAX
00401972 . E8 08000000 CALL 0040197F
00401977 . 8907 MOV DWORD PTR DS:[EDI],EAX
00401979 . 47 INC EDI
0040197A . 46 INC ESI
0040197B ^ E2 F0 LOOP SHORT 0040196D
0040197D . C9 LEAVE
0040197E . C3 RETN
0040197F $ 55 PUSH EBP
00401980 . 8BEC MOV EBP,ESP
00401982 . 83C4 FC ADD ESP,-4
00401985 . 33C0 XOR EAX,EAX
00401987 . BB 56141200 MOV EBX,121456
0040198C . 05 11100001 ADD EAX,100101
00401991 . C645 F2 2D MOV BYTE PTR SS:[LOCAL.1],2D
00401995 . 8A5D FC MOV BL,BYTE PTR SS:[LOCAL.1]
00401998 . 02C3 ADD AL,AL
0040199A . C645 FD 3F MOV BYTE PTR SS:[LOCAL.1+1],3F

```

Entry point of main module

DLLMainCRT01()

```

0040195B $ 8D3D E4324000 LEA EDI,[4032E4]
0040195C . 6A 00 PUSH 0
0040195D . 6A 00 PUSH 0
0040195E . 50 PUSH EAX
0040195F . DF2C24 FILD QWORD PTR SS:[LOCAL.2]
00401960 . DF3424 FBSTP TBYTE PTR SS:[LOCAL.2]
00401961 . 59 POP ECX
00401962 . 58 POP EAX
00401963 . 8BD1 MOV EDX,ECX
00401964 . 8BD8 MOV EBX,EAX
00401965 . C1E9 04 SHR ECX,4
00401966 . C1E8 04 SHR EAX,4
00401967 . 83E3 0F AND EBX,0000000F
00401968 . 81E2 0F0F0F01 AND EDX,0F0F0F0F
00401969 . 81E1 0F0F0F01 AND ECX,0F0F0F0F
0040196A . 81C2 3030303 ADD EDX,30303030
0040196B . 81C1 3030303 ADD ECX,30303030
0040196C . 83C0 30 ADD EAX,30
0040196D . 83C3 30 ADD EBX,30
0040196E . 8807 MOV BYTE PTR DS:[EDI],AL
0040196F . 885F 01 MOV BYTE PTR DS:[EDI+1],BL
00401970 . 884F 08 MOV BYTE PTR DS:[EDI+8],CL
00401971 . 8857 09 MOV BYTE PTR DS:[EDI+9],DL
00401972 . 886F 06 MOV BYTE PTR DS:[EDI+6],CH
00401973 . 8877 07 MOV BYTE PTR DS:[EDI+7],DH

```

Entry point of main module

Đoạn phát sinh key:

```
PUSH OFFSET 0040331C  
CALL <JMP.&user32.GetDlgItemText  
  
MOV ESI,OFFSET 0040331C  
MOV EDI,OFFSET 0040333B  
MOV ECX,10  
ΓMOVZX EAX,BYTE PTR DS:[ESI]  
  
ADD EAX,ECX  
CMP EAX,21  
JAE SHORT 004019D2  
ADD EAX,21  
CMP EAX,7B  
JLE SHORT 004019D9  
SHR EAX,1  
  
MOV DWORD PTR DS:[EDI],EAX  
INC EDI  
INC ESI  
LOOP SHORT 0040196D  
  
MOV EAX,WORD PTR DS:[40333D]  
  
MOV DWORD PTR DS:[4032E4],EAX  
  
PUSH EAX  
FIOLD QWORD PTR SS:[LOCAL.2]  
FBSTP TBYTE PTR SS:[LOCAL.2]  
POP ECX  
  
MOV EDX,ECX  
  
SHR ECX,4  
  
MOV BYTE PTR DS:[EDI+8],CL  
MOV BYTE PTR DS:[EDI+9],DL  
  
XOR EAX,EAX  
XOR EBX,EBX  
MOV AX,WORD PTR DS:[40333B]  
MOV BX,WORD PTR DS:[EDI+8]  
SUB AX,BX  
XOR EAX,00001B3F  
SUB EAX,123
```

Ý nghĩa:

*[40331C] = Key

[ESI] = [40331C]
[EDI] = [4033B]
ECX = 16
EAX = *[ESI]
EAX += ECX
IF (EAX < 33)
 EAX += 33
IF (EAX > 123)
 EAX >>= 1

*[EDI] = EAX
EDI += 1
ESI += 1
WHILE (ECX != 0)
 EAX = *[40333D] = *[40333B + 2]

*[4032E4] = EAX

Push EAX vào stack
Covert stack thành int
Pop stack vào ECX

EDX = ECX

ECX >>= 4

[EDI][8] = CL
[EDI][9] = DL

EAX ^= EAX = 0
EBX ^= EAX = 0
AX = *[40333B]
BX = *[EDI+8]
AX -= BX
EAX ^= 6975
EAX -= 291

OR EAX, EAX

EAX |= EAX



