TUTORIAL by DEIBIZ_XXL

Víctima:	**CeeBot-A**
Protección:	**CD-Check**
Fabricante:	**EPSITEC**
Dificultad:	**Sorpresivamente facilísimo**
Risas:	**Muuuchas XDD**

1 - Descripción del problema...

Aclaración: este tutorial se basa en la versión completa del programa, que hace CD-Check. Si existe alguna versión trial o algo similar (cosa que desconozco), este tutorial no os servirá para convertirla en completa.

La cuestión es que tienes que tener el CD insertado para poder jugar. Tratándose de un juego relativamente moderno, esto no es de extrañar... pero aún así, se podría haber hecho mejor. Me considero un newbie recién inserto en la escena, pero, encontrándote cosas así, te dan ganas de seguir investigando.

Si os gusta el juego, compradlo, sólo así podremos hacer que los estudios sigan desarrollando. Gracias.

2 - A por ello (método que he seguido)

Obviamente, lo primero que hay que hacer es instalar el pograma.

Tras ello, decidí identificar con el uso de PEiD 0.94 con qué herramienta se había compilado el programa y si estaba comprimida o como dicen algunos de mis compañeros, "empacada". Esto lo aprendí del tito Narvaja, el cual me lo inculcó como una costumbre muy sana al encarar los programas para investigar sus métodos de protección. Otra de las costumbres que introdujo en mí fue la de hacer una copia de seguridad del ejecutable, por si las cosas venían mal dadas, cosa que también hice.

Pues resulta que la referencia que me daba PEiD era la archiconocida "Microsoft Visual C++ 5.0". Ni comprimido ni nada. La cosa, de momento, pintaba fácil.

Me decidí por OllyDbg, ya que la formación que me habeís dado entre todos en el uso de este depurador ha sido excelente sobre todo gracias a los tutoriales de Ricardo Narvaja.

Tras abrirlo, eché un vistazo a la lista de API que se importaban, a ver si había alguna que diera el cante para poner un "breakpoint on import". La sorpresa fue mayúscula al ver la mítica función "GetDriveTypeA". Todo un lujo. Ni tan siquiera se obtenía la dirección de inicio de la API en tiempo de ejecución.

Después de poner el breakpoint, ejecuté el programa. Cual fue mi sorpresa, cuando vi que saltaba. No sólo te dejaban la API a huevo si no que encima la utilizaban de verdad XD.

Subiendo unas cuantas líneas para arriba detecté que el inicio de la rutina de comprobación del CD-ROM comenzaba en 437980. La ventaja de usar OllyDbg es que te indica donde comienzan y donde acaban las rutinas mediante una especie de paréntesis gigante XD. Puse un breakpoint en esa línea y volví a ejecutar el programa. A continuación os pongo un listado del código de la función, para después explicar las cosas más interesantes:

```
00437980  /$  55           PUSH EBP
00437981  |.  8BEC         MOV EBP,ESP
00437983  |.  83EC 7C      SUB ESP,7C
```

```
00437986  |.  894D 84       MOV DWORD PTR SS:[EBP-7C],ECX
00437989  |.  C745 88 00000>MOV DWORD PTR SS:[EBP-78],0
00437990  |.  8D45 94       LEA EAX,DWORD PTR SS:[EBP-6C]
```

RegOpenKeyExA - Apertura de una clave del registro

- Como OllyDbg nos indica, aquí se realiza una llamada a RegOpenKeyExA. Esto supone que se está accediendo al registro para leer algo. Se puede observar que se abre la clave **HKEY_LOCAL_MACHINE** y la subclave **Software\Epsitec\CeeBot-A\Setup**

```
00437993  |.  50            PUSH EAX
00437994  |.  68 19000200   PUSH 20019
00437999  |.  6A 00         PUSH 0
0043799B  |.  68 D8F35400   PUSH ceebot-a.0054F3D8
004379A0  |.  68 02000080   PUSH 80000002
004379A5  |.  FF15 08205400 CALL DWORD PTR DS:[<&ADVAPI32.RegOpenKey>
```

- Fíjense como nos ayuda OllyDbg con lo que pone a la derecha del código.

```
 ; / pHandle
 ; | Access = KEY_READ
 ; | Reserved = 0
 ; | Subkey = "Software\Epsitec\CeeBot-A\Setup"
 ; | hKey = HKEY_LOCAL_MACHINE
 ; \ RegOpenKeyExA
```

- En esto consiste el análisis que hace OllyDbg del código, pero... ¿qué leera del registro? no os preocupeís, enseguida saldremos de dudas.

```
004379AB  |.  8945 90       MOV DWORD PTR SS:[EBP-70],EAX
004379AE  |.  837D 90 00    CMP DWORD PTR SS:[EBP-70],0
004379B2  |.  74 0A         JE SHORT ceebot-a.004379BE
004379B4  |.  B8 14000000   MOV EAX,14
004379B9  |.  E9 B9000000   JMP ceebot-a.00437A77
004379BE  |>  C745 98 01000>MOV DWORD PTR SS:[EBP-68],1
004379C5  |.  C745 8C 64000>MOV DWORD PTR SS:[EBP-74],64
004379CC  |.  8D4D 8C       LEA ECX,DWORD PTR SS:[EBP-74]
```

RegQueryValueExA - Lectura de un valor de una clave específica del registro

- Realmente, no nos hace falta saber mucho acerca de qué hace el código completo, como se puede ver, si no sólo donde guarda los valores de retorno y qué hace cada API. Si os fijaís en la ayudita que nos da OllyDbg, podeís ver que lee el valor de la clave que se corresponde a **CDpath** ¿no os resulta un nombre sospechoso?.

```
004379CF  |.  51            PUSH ECX
004379D0  |.  8B55 84       MOV EDX,DWORD PTR SS:[EBP-7C]
004379D3  |.  81C2 AC000000 ADD EDX,0AC
004379D9  |.  52            PUSH EDX
004379DA  |.  8D45 98       LEA EAX,DWORD PTR SS:[EBP-68]
004379DD  |.  50            PUSH EAX
004379DE  |.  6A 00         PUSH 0
004379E0  |.  68 F8F35400   PUSH ceebot-a.0054F3F8
004379E5  |.  8B4D 94       MOV ECX,DWORD PTR SS:[EBP-6C]
004379E8  |.  51            PUSH ECX
004379E9  |.  FF15 00205400 CALL DWORD PTR DS:[<&ADVAPI32.RegQueryVa>
```

- De nuevo la ayudita...

```
; / pBufSize
; |
; |
; | Buffer
; |
; | pValueType
; | Reserved = NULL
; | ValueName = "CDpath"
; |
; \ RegQueryValueExA
```

- Como se puede observar, también se pasan a la API algunos parámetros más, que para los efectos no nos interesan. Pero hay una que si... y es la variable buffer, que es donde se insertarán los datos leídos de dicha clave. En ejecución, aparece al lado de "Buffer" la dirección donde se insertarán los datos leídos. En este programa, se escriben en la pila del sistema, en la dirección **0012FBD4**.

- La siguiente comprobación es para ver si la función se ha ejecutado con éxito por su valor de retorno en EAX.

```
004379EF  |.  8945 90       MOV DWORD PTR SS:[EBP-70],EAX
004379F2  |.  837D 90 00    CMP DWORD PTR SS:[EBP-70],0
004379F6  |.  75 06         JNZ SHORT ceebot-a.004379FE
004379F8  |.  837D 98 01    CMP DWORD PTR SS:[EBP-68],1
004379FC  |.  74 07         JE SHORT ceebot-a.00437A05
004379FE  |>  B8 14000000   MOV EAX,14
00437A03  |.  EB 72         JMP SHORT ceebot-a.00437A77
```

GetDriveTypeA - Dada una ruta, te devuelve qué tipo de unidad es (disco duro, cdrom...)

```
00437A05  |>  8B55 84       MOV EDX,DWORD PTR SS:[EBP-7C]
00437A08  |.  8A82 AC000000 MOV AL,BYTE PTR DS:[EDX+AC]
```

- Mete en la pila los valores correspondientes al retorno de la función anterior (sólo la letra que dicta la unidad, leída del registro, que está en AL), la barra y los dos puntos, para conformar la ruta de la unidad, que posteriormente se carga en ECX, que es lo que se pasa a la función como primer parámetro.

```
00437A0E  |.  8845 9C       MOV BYTE PTR SS:[EBP-64],AL
00437A11  |.  C645 9D 3A    MOV BYTE PTR SS:[EBP-63],3A
00437A15  |.  C645 9E 5C    MOV BYTE PTR SS:[EBP-62],5C
00437A19  |.  C645 9F 00    MOV BYTE PTR SS:[EBP-61],0
00437A1D  |.  8D4D 9C       LEA ECX,DWORD PTR SS:[EBP-64]
00437A20  |.  51            PUSH ECX
00437A21  |.  FF15 DC215400 CALL DWORD PTR DS:[<&KERNEL32.GetDriveTy>
00437A27  |.  8945 90       MOV DWORD PTR SS:[EBP-70],EAX
```

- La ayuda:

```
; / RootPathName
; \ GetDriveTypeA
```

- Este valor hay que cambiarlo, ya que 5 es lo correspondiente a un CD-ROM (Constante DRIVE_CDROM). Esto se puede ver el Win32 API reference. (WIN32.HLP). Hay que sustituirlo por un 3, que es lo que corresponde al disco duro (Constante DRIVE_FIXED).

00437A2A |. 837D 90 03   CMP DWORD PTR SS:[EBP-70],5

```
00437A2E  |.  74 07         JE SHORT ceebot-a.00437A37
00437A30  |.  B8 15000000   MOV EAX,15
00437A35  |.  EB 40         JMP SHORT ceebot-a.00437A77
```

- Aquí vemos el nombre del archivo que se lee "install.ini", que casualmente se encuentra en la raíz del CD-ROM de instalación del programa.

```
00437A37  |>  68 00F45400   PUSH ceebot-a.0054F400                ;  ASCII "install.ini"
00437A3C  |.  8D55 9C       LEA EDX,DWORD PTR SS:[EBP-64]
00437A3F  |.  52            PUSH EDX
00437A40  |.  E8 6B8E0F00   CALL ceebot-a.005308B0
00437A45  |.  83C4 08       ADD ESP,8
```

- Si sabeís C, este es el parámetro que se pasa a fopen, significa que se abre para lectura en modo binario.

```
00437A48  |.  68 0CF45400   PUSH ceebot-a.0054F40C                ;  ASCII "rb"
```

- Aquí hará algo con el archivo que no nos interesa, puesto que ya le hemos "engañado", y el archivo que hemos copiado en C:\ es "original", así que ni nos importa.

```
00437A4D  |.  8D45 9C       LEA EAX,DWORD PTR SS:[EBP-64]
00437A50  |.  50            PUSH EAX
00437A51  |.  E8 DA9F0F00   CALL ceebot-a.00531A30
00437A56  |.  83C4 08       ADD ESP,8
00437A59  |.  8945 88       MOV DWORD PTR SS:[EBP-78],EAX
00437A5C  |.  837D 88 00    CMP DWORD PTR SS:[EBP-78],0
00437A60  |.  75 07         JNZ SHORT ceebot-a.00437A69
00437A62  |.  B8 15000000   MOV EAX,15
00437A67  |.  EB 0E         JMP SHORT ceebot-a.00437A77
00437A69  |>  8B4D 88       MOV ECX,DWORD PTR SS:[EBP-78]
00437A6C  |.  51            PUSH ECX
00437A6D  |.  E8 0E9F0F00   CALL ceebot-a.00531980
00437A72  |.  83C4 04       ADD ESP,4
00437A75  |.  33C0          XOR EAX,EAX
00437A77  |>  8BE5          MOV ESP,EBP
00437A79  |.  5D            POP EBP
00437A7A  \.  C3            RETN
```

3 - Solución

La solución consiste en copiar el archivo install.ini en la raíz del disco duro C:, cambiar la comprobación marcada en negrita, sustituyendo el 5 por un 3, y cambiar esa clave del registro, poniendo "C:\" en lugar de "D:\" o lo que tengaís en esa clave.

El crack para este programa consistiría en un crack que cambiaría el byte, un archivo de importación de claves del registro, que podría tener el siguiente código y se guardaría como "CB.reg"

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Epsitec\CeeBot-A\Setup]
"CDpath"="C:\"
```

Y posteriormente habría que copiar el archivo "install.ini" del raíz del CD-ROM de instalación al raíz del disco duro "C:\". Con esto funcionaría perfectamente.

Habría más soluciones, como parchear directamente toda la rutina para que funcionara sin

necesidad de leer ningún archivo ni nada por el estilo o evitar la llamada a esa rutina parcheando directamente el call y las comprobaciones de retorno sucesivas. Sin embargo, esta me ha parecido la mejor forma, ya que se aprende algo acerca de las API utilizadas para leer el registro y apertura de archivos. Otra experiencia adicional sería investigar qué comprueba en el archivo para verificar que es el real y veraz CD-ROM de CeeBot-A.

4 - Agradecimientos

Al equipo de Cracks Latinos, especialmente a **Ricardo Narvaja** y **RedHawk**. Aunque hace tiempo que me apunté a la lista de correo, hasta ahora no había tenido tiempo de meterme en el rollito. Pese a este agadecimiento especial, extiendo el agradecimiento a todas las personas que participan en esa lista de correo por el tiempo que emplean haciendo que newbies como yo nos metamos en el ajo.