melted MeltICE

(SoftIce 3.xx detection and another lesson for shareware programmers)

by Frog's Print

(22 August 1997, slightly edited by Fravia)
With an important addition by Kox! (27 August 1997)
[Picture]Courtesy of fravia's page of reverse engineering
Well, Frog's Print finding are indeed interesting. I'm pretty sure that we are going to assist, in the next future, to an explosion of many little anti-softice (and anti-wdasm) tricks. I may open an extra section of my site to this if necessary... btw, to-day ReZiDeNt signalled an "anti-BRW" trick inside Ultraedit...

Note that since the programmers keep programming in "high" languages, all this can (at most) defeat the stupid lamers... the good-guy=1 and beggar off=0 flags are always the same... poor programmers... how long should we repeat it to you? YOU HAVE TO PROGRAM PROTECTIONS IN ASSEMBLER ON YOUR OWN, you do not have to:

1) Use Visual Basic made protections;
2) Use ready-made third party protections (if the people selling them

   would put their money where their mouths are you would see many more working demos of their protections on the net, btw);

3) Use Visual C++ made protections;

Write your own small protections routines in assembler using some forgotten dos call and some empty bits inside the file header for Ginger Rogers' sake, and you'll fend off more than 80% of all attacks on your software!

Well here you go: Meltice melted away under the touch of Frog's Print...

MeltICE
(SoftIce 3.xx detection)

by Frog's Print

I found today at http://www.window95.com a file named MeltICE. This is an updated version of "ICEcream" whose only purpose was to detect if a version of SoftIce was loaded and "to make shareware developers a little bit easier about the safety of their software" as it's author (David Eriksson) wrote.

The file contains a source code that (lazy) protectionists can add to their programs.

MelICE was written specially for the new versions of SoftIce: v3.0 and 3.01 (Win 95/NT).

How it works:

The program will open the VxD driver named SICE (Windows 95) or NTICE (for Windows NT) with CreateFile. It will then check the file's Handle (in EAX) in order to see if SoftIce Win95 or WinNT is loaded or not.

Below are the source code of MeltICE and a disassembly listing that I did with W32dasm of an .exe file compiled with this code:

MeltICE - SoftICE '95 version 3 detection - Made by David Eriksson

```
#include <stdio.h>
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
```

```
// See if SoftICE version 3.x for Windows 95 is loaded
BOOL IsSoftIce95Loaded()

{

HANDLE hFile;
hFile = CreateFile( "\\\\.\\SICE",
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,
NULL);


if( hFile != INVALID_HANDLE_VALUE )

        {

CloseHandle(hFile);
return TRUE;

        }

return FALSE;


}

// See if SoftICE version 3.x for Windows NT is loaded
BOOL IsSoftIceNTLoaded()

{

HANDLE hFile;
hFile = CreateFile( "\\\\.\\NTICE",
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,
NULL);


if( hFile != INVALID_HANDLE_VALUE )

        {

CloseHandle(hFile);
return TRUE;

        }
```

```
return FALSE;

}

// Example code for calling these functions
int main(void)

{

if( IsSoftIce95Loaded() )
printf("SoftICE for Windows 95 is active!\n");
else if( IsSoftIceNTLoaded() )
printf("SoftICE for Windows NT is active!\n");
else
printf("Can't find SoftICE with this method!\n");


return 0;

}
```

And now, the Dead Listing of an .exe file using that code:

Referenced by a CALL at Address :004011DE

```
:00401080 E87BFFFFFF      call 00401000    ; first, check for S-Ice Win95

:00401085 85C0            test eax, eax    ; check if loaded...

:00401087 7410            je 00401099      ; No, jump to check_NT, if yes:

:00401089 6894604000      push 00406094    ;->"SoftICE for Windows 95 is active!"

:0040108E E83D000000      call 004010D0

:00401093 83C404          add esp, 4

:00401096 33C0            xor eax, eax

:00401098 C3              ret              ; S-Ice Win95 detected. Bye_bye.

:Check_NT
:00401099 E8A2FFFFFF      call 00401040    ; Now, check for S-Ice WinNT

:0040109E 85C0            test eax, eax    ; check if loaded...

:004010A0 7410            je 004010B2      ; jump if NOT loaded to can't_find, else

:004010A2 6870604000      push 00406070    ;->"SoftICE for Windows NT is active!"

:004010A7 E824000000      call 004010D0

:004010AC 83C404          add esp, 4
```

```
:004010AF 33C0          xor eax, eax

:004010B1 C3            ret             ; S-Ice WinNT detected. Bye_bye.

:can't_find
:004010B2 6848604000    push 00406048   ;->"Can't find SoftICE with this method!"

:004010B7 E814000000    call 004010D0

:004010BC 83C404        add esp, 4

:004010BF 33C0          xor eax, eax

:004010C1 C3            ret             ; S-Ice not found.
```

**********End of detection**********

The detection/CreateFileA routine for S-Ice Win95:

```
:00401000 6A00          push 00000000           ; CreateFileA parameters

:00401002 6880000000    push 00000080           ; ...

:00401007 6A03          push 00000003           ; ...

:00401009 6A00          push 00000000           ; ...

:0040100B 6A03          push 00000003           ; ...

:0040100D 68000000C0    push C0000000           ; ...
```

Possible StringData Ref from Data Obj ->"\\.\SICE" ; VxD driver for S-Ice Win95

```
:00401012 6830604000    push 00406030
```

Reference To: KERNEL32.CreateFileA, Ord:0031h

```
:00401017 FF15BCA04000  Call dword ptr [0040A0BC]  ; CreateFileA

:0040101D 83F8FF        cmp eax, FFFFFFFF       ; Handle= -1 ?

:00401020 740D          je 0040102F             ; Yes, jump otherwise...

:00401022 50            push eax                ; SoftIce Win95 IS loaded!
```

Reference To: KERNEL32.CloseHandle, Ord:0018h

```
:00401023 FF15F8A04000  Call dword ptr [0040A0F8]  ; Close file's handle
```

:00401029 B801000000 mov eax, 00000001 ; Eax:=1

:0040102E C3 ret !

; Back to the caller
Referenced by a ©onditional Jump at Address :00401020
:0040102F 33C0 xor eax, eax ; Eax:=0 (not loaded)

:00401031 C3 ret !

; Back to the caller

...

The detection/CreateFileA routine for S-Ice WinNT:

...

Referenced by a CALL at Address :00401099
:00401040 6A00 push 00000000 ; CreateFileA parameters

:00401042 6880000000 push 00000080 ; ...

:00401047 6A03 push 00000003 ; ...

:00401049 6A00 push 00000000 ; ...

:0040104B 6A03 push 00000003 ; ...

:0040104D 68000000C0 push C0000000 ; ...

Possible StringData Ref from Data Obj ->"\\.\NTICE"; VxD driver for S-Ice WinNT
:00401052 683C604000 push 0040603C
Reference To: KERNEL32.CreateFileA, Ord:0031h

:00401057 FF15BCA04000 Call dword ptr [0040A0BC] ; CreateFileA

:0040105D 83F8FF cmp eax, FFFFFFFF ; Handle= -1 ?

:00401060 740D je 0040106F ; Yes, jump otherwise...

:00401062 50 push eax ; SoftIse WinNT IS loaded!

Reference To: KERNEL32.CloseHandle, Ord:0018h

:00401063 FF15F8A04000 Call dword ptr [0040A0F8] ; Close file's handle

:00401069 B801000000 mov eax, 00000001 ; Eax:=1

:0040106E C3 ret !

```
; Back to the caller
Referenced by a ©onditional Jump at Address :00401060
:0040106F 33C0          xor eax, eax            ; Eax:=0 (not loaded)

:00401071 C3            ret                     !

; Back to the caller
```

OK, we see that this new simple and 'ready-to-use' protection will probably please a huge amount of unexperimented shareware programmers. Since it is available at Windows95.com, I assume we may have to face it very soon.

But is S-T-U-P-I-D because we now will be able to check if any program is detecting Soft-Ice even before it will have the time to do so: just with a BPX CreateFile(A).

Anyway, this will make shareware crackers a little bit easier about the safety of the software they want to reverse engineer too.

Here is the important addition by Kox (27 August 1997):

Defeating MeltedICE for good in 10 Seconds.
They sure can't be serious for publishing such MeltedICE.
You can never rely on a constant string comparison to detect SoftICE.

Here is how to defeat it in 10 seconds:

(I guess everybody knows this, but just in case someone didn't think of it..)

Just replace the string "SICE" with "KICE" (or whatever you want) in the files

"Winice.exe" and "nmtrans.dll"

(Please do not use "KICE" ,just use a unique string... cause those MeltedICE people may update it by checking for "KICE" too :)

And gone is the check for VxD name "SICE".
You can check the VxD names with many tools.. (Infospy for example)
This goes for win95 version and i guess would work for NT too..
(although for NT you have to recalculate the checksums as in +HCU Project 2)
This way of detecting SoftICE is the same one used in the "nmtrans.dll" Function "DevIO_ConnectToSoftICE" so i guess they reversed engineer that function.. (You see the pun,they do reverse engineer too, i guess no one can just live without Reverse Engineering :-) well,except for zombies ..