Virusinfiziert?

Life ... auch in Ihrem Amiga!

Ist Ihnen das auch schon passiert? Statt ordentlich durchzubooten meldete sich der Amiga mit einer seltsamen Nachricht: "Something wonderful has happened...". Oder Ihr Lieblingsbootsektor ist auf einmal verschwunden. Wenn Sie eine dieser Fragen mit "Ja" beantworten können, dann "lebt" in Ihrem Amiga ein **Virus**!

Nun mögen Sie sagen: "Na und, was geht mich das an?". Doch halt! Wie auch für uns Menschen Virusinfektionen durchaus gefährlich sein können, birgt auch ein Computervirus eine nicht zu unterschätzende Gefahr in sich.

Doch zunächst zur Arbeitsweise des einzigen (bisher!) bekannten Virus, der von der Schweizer Gruppe SCA geschrieben und in Umlauf gebracht worden ist.

Das Virusprogramm wird auf einer Diskette im Boot-Bereich (Block 0 und 1) abgelegt. Wenn diese Diskette gebootet wird, lädt der Amiga dieses Programm in den Speicher und startet es. Danach ist es fast nur durch Ausschalten oder Löschen des gesamten Speichers wieder zu entfernen.

Wenn der Computer nach einem Reset wieder von einer anderen Disk booten soll, so verhindert der Virus dies und versucht sich statt dessen auf die Bootblocks zu schreiben und sich somit zu verbreiten.

Zwar handelt es sich bei dem SCA Virus um ein verhältnismäßig harmloses Programm, das allerhöchstens Bootblocks zerstört, doch können nach dem selben Prinzip auch viel gefährlichere geschrieben werden, die nach einigen erfolgreichen Kopierversuchen jede eingelegte Disketten zerstören oder sogar Festplattendaten.

Bei dem vorliegenden Virus handelt es sich jedoch glücklicherweise um einen recht einfachen Vertreter, den man leicht ausfindig machen und ihn somit auch bekämpfen kann.

Zu diesem Zweck finden Sie zwei Listings in Roeshe's Computer Mag.: Das erste ist in reinem Maschinencode geschrieben und sollte mit dem Seka-Assembler assembliert werden (es müßte allerdings auch mit anderen klappen), das zweite ist mit dem Manx Aztec C Compiler geschrieben und läuft mit der 3.20 Version, müßte allerdings auch mit der 3.40 Version arbeiten.

Zur Funktionsweise:

Das erste Programm duchsucht den in Frage kommenden Speicher nach einer Textfolge, die im Virus vorkommt. Wenn diese gefunden wurde, wird man mittels eines Alerts deutlich darauf hingewiesen.

Dieses Programm sollte man auf alle oft benutzten Disketten kopieren, von denen man weiß, daß man sie auch ohne Schreibschutz bootet oder booten muß.

Man sollte etwa folgendermaßen vorgehen:

- den Sourcecode in den Seka Assembler eingeben.

```
;************
; Anti-Virus
;   S C A
;************
; (C) 1987 by Roeske's Computer Mag.
; written 1987 by G. Glendown
alnum = 0
height = 40

; Exec Routinen:
Forbid=             -132
Permit=             -138
OpenLibrary=        -552
CloseLibrary=       -414

; Intuition Routinen:
DisplayAlert=       -90

          move.l    4,a6                    ; Exec Base
          jsr       Forbid(a6)              ; Multitask. sperren
          move.l    #$70000,a0              ; suchen ab
1:
          cmp.b     #$41,(a0)+              ; Zeichen = "A"
          beq.s     foundb                  ; Ja
          cmp.l     #$80000,a0              ; Schlu_ ?
          bne.s     1                       ; Nein
          bra.s     exit                    ; Ja
foundb:                                     ; "A" gefunden, weitersuchen
          lea       data,a1                 ; Tabellenadr. nach a1
          move.l    #8,d0                   ; Anzahl Zeichen
11:
          move.b    (a1,d0),d1              ; Inhalt der Adr. (a1+d0) nach d1
          cmp.b     (a0)+,d1          ; mit Tabelle vergleichen
          bne.s     1                       ; stimmt nicht, zur/ck
          dbf       d0,11                   ; weitermachen
;********* Virus gefunden ! **********
gef:
          move.l    4,a6                    ; Exec Base
          jsr       Permit(a6)              ; Multitasking erlauben
          move.l    4,a6                    ; Exec Base
          lea       intuiname,a1            ; Adresse von "intuition.library"
          clr.l     d0                      ; d0 l vschen
          jsr       OpenLibrary(a6)         ; Intuition vffnen
```

```
b1:
          cmp.l     #0,d0                         ; Offen ?
          beq.s     nointui                       ; nein, PowerLED aus
          move.l    d0,intuibase                  ; IntuitionBase sichern
          move.l    d0,a6                         ; und nach a6
          move.l    #alnum,d0           ; Alerttype
          lea       alstr,a0            ; Adresse Alertstring
          move.l    #height,d1                    ; Hvhe Alert
          jsr       DisplayAlert(a6)    ; und anzeigen
          move.l    4,a6                          ; Exec Base
          move.l    intuibase,a1                  ; IntuiionBase
          jsr       CloseLibrary(a6)    ; Intuition schlie_en
          rts                                     ; und zur/ck
nointui:
          or.b      #2,$bfe001                    ; PowerLED auf dunkel schalten
          rts                                     ; zur/ck
exit:
          move.l    4,a6                          ; Exec Base
          jmp       Permit(a6)                    ; MultiTasking erlauben


alstr:                                            ; AlertString
          dc.w      100
          dc.b      20,"Red Alert ! Virus in System !",0,0
data:                                             ; Daten Tabelle
          dc.b      "!ACS!ACS!"
even
intuibase:                                        ; Speicher f/r Intuition Base
          dc.l      0
intuiname:                                        ; Name der Intui-Library
          dc.b      "intuition.library",0
```

```
/*****************************************************************************
              A             V                 C
                  utomatic  irus               hecker
        (C)   1987 by       Roeske's Computer Mag.
                            Written by G.Glendown
*****************************************************************************/

#include <exec/types.h>
#include <graphics/gfxbase.h>
#include <intuition/intuition.h>
#include <devices/trackdisk.h>
#include <exec/ports.h>
#include <exec/nodes.h>
#include <exec/io.h>
#include <exec/devices.h>

#define TD_READ CMD_READ;

struct IntuitionBase         *IntuitionBase;
struct GfxBase               *GfxBase;
struct Window                *win, *OpenWindow();
struct IntuiMessage*msg,*GetMsg();
struct MsgPort               *diskport,*CreatePort();
struct IOExtTD               *diskreq;
struct IORequest   *CreateExtIO();

BYTE                         Buffer[TD_SECTOR+1];
ULONG                        dcc;

char                  text[]={"       Diskette in DF0: ist infiziert ! Bitte sofort INSTALLen ! \0\0"};
             /* Achtung !      ----------------------  Genau abtippen ! */

struct NewWindow windef  =  {
         100, 50, 240, 100, 0, 1,
         CLOSEWINDOW|DISKINSERTED,
         WINDOWDEPTH|WINDOWSIZING|WINDOWDRAG|WINDOWCLOSE|RMBTRAP,
         NULL, NULL, NULL, NULL, NULL, 30, 15, 240, 100, WBENCHSCREEN };
int                          lncnt;

char data[] =           {" y a VIRUS"};
       /* Achtung!       ----------  Genau abtippen ! */


main ()
{
         register struct RastPort     *rp;
         long                                 pen,t;
         long                                 class,*l;
         int                                  code;

         text[0]=(char)0;
         text[1]=(char)6;
         text[2]=(char)15;
```

- den Assembler mit "a" starten und nach dem Entfernen aller Tippfehler den erzeugten Code mit "wo" auf Disk abspeichern (z.B. unter dem Namen "av")

- das File auf alle Disketten kopieren.

- in den jeweiligen "Startup-Sequence"n als erste Zeile "av" (bzw. den jeweiligen Namen) einfügen.

Das zweite Programm arbeitet etwas anders. Es durchsucht direkt die eingelegten Disketten nach dem Virus. Dabei sollte man es entweder mit run aus dem CLI starten oder von der Workbench, sodaß es jederzeit auf einen eventuell vorhandenen Virus hinweist. Dabei wird ausgenutzt, daß Intuition bei einem Diskettenwechsel eine Message über den Useport eines Windows schickt und somit ein Programm darüber benachrichtigt.

Dies wird bei dem Programm ausgenutzt, damit, sobald diese Message kommt, die Laufwerke überprüft werden. Beenden kann man das Programm mit dem Close-Gadget des Windows.

Natürlich ist es in diesem Fall kein großes Kunsstück den Virus zu entdecken. Doch bei sich selbst modifizierenden Virusprogrammen kann man sich nicht mehr so leicht, möglicherweise sogar überhaupt nicht mehr schützen.

Daher an dieser Stelle eine Bitte im Namen aller Computerfans an die "Profis", die in der Lage sind, solche Viren zu programmieren (was nicht schwer ist!): Nutzt Euer Wissen zu friedlichen Zwecken. Denkt an die armen Leute, denen vielleicht ein Programm drauf geht, das ein paar hundert Mark gekostet hat und nicht nur deren Raubkopien...

```
        openstuff();
        rp = win->RPort;

        SetAPen(rp,1L);
        SetDrMd(rp,JAM1);

        printw(win," ");
        printw(win,"*************************");
        printw(win,"      Find VIRUS");
        printw(win,"*************************");
        printw(win,"  written by G. Glendown");
        printw(win," ");
        printw(win,"Funktioniert gegen  ");
        printw(win,"den SCA Virus (Something ");
        printw(win,"wonderful ...)");

        for (t = 1;t;) {
                WaitPort(win->UserPort);

                while (msg = GetMsg(win->UserPort)) {
                        class =   msg->Class;
                        code  =   msg->Code;
                        l     =   (long *)msg->IAddress;
                        ReplyMsg (msg);

                        if (class==DISKINSERTED) CheckDisk();
                        if (class==CLOSEWINDOW)  t = 0;
                }
        }
        ex ();
}

CheckDisk()
{
int                     error;
long                    g,n,m,u;
long                    drive;
for (m = 0;m<4;m++) {
        diskport = CreatePort (0L,0L);

        if (diskport) {
                diskreq = (struct IOExtTD *) CreateExtIO (diskport,(LONG)sizeof (struct IOExtTD));

                if (diskreq) {
                        error = OpenDevice(TD_NAME,(long)m,diskreq,0L);

                        if (!error) {
                                for (u = 0;u<TD_SECTOR;u+=5) Buffer[u] = ' ';
                                diskreq->iotd_Req.io_Command = TD_CHANGENUM;
                                DoIO(diskreq);
                                dcc = diskreq->iotd_Req.io_Actual;
                                MotorOn();
                                ReadBlock(1);
                                MotorOff();

                                for (g = 0;g<(TD_SECTOR-7);g++) {

                                        if ((int)(Buffer[g]&255)==(int)'b') {

                                                for (n = 1;n<8;n++) {
                                                        if ((int)(Buffer[g+n]&255)!=(int)data[n]) n = 10;
                                                }

                                                if (n<10) {
                                                        text[21] = m+0x30;
                                                        DisplayAlert (0L,text,28L);
                                                        g = TD_SECTOR;
                                                }
                                        }
                                }
                                CloseDevice (diskreq);
                        }
                        DeleteExtIO (diskreq,(long)sizeof(struct IOExtTD));
                }
                DeletePort (diskport);
        }
}
}

openstuff ()
{
        char *OpenLibrary();

        if (!(IntuitionBase = (struct IntuitionBase *)
                OpenLibrary ("intuition.library", 1L)))
                        ex();

        if (!(GfxBase = (struct GfxBase *)
                OpenLibrary ("graphics.library", 1L)))
                        ex();

        if (!(win = OpenWindow (&windef)))
                        ex ();
}

ex()
{
if (win)            CloseWindow (win);
if (GfxBase)                CloseLibrary(GfxBase);
if (IntuitionBase)  CloseLibrary (IntuitionBase);
exit(111);
}

closestuff ()
{
        CloseWindow         (win);
        CloseLibrary        (GfxBase);
        CloseLibrary        (IntuitionBase);
}


printw(win,text)
char                        text[];
struct Window *win;
{
        int                                         flag,code;
        register struct RastPort     *rp;

        rp = win -> RPort;
        SetAPen(rp,1L);
        Move(rp,5L,(long)(lncnt*9+9));
        Text(rp,text,(long)(strlen(text)));
        lncnt++;
}

ReadBlock(bn)
int bn;
{
        short                                       cyl,sec,hd;
        long                                        offset;

        cyl =     bn/22;
        bn -=     cyl*22;
        hd =      (bn>10);
        sec =     bn-hd*11;
        diskreq->iotd_Req.io_Length =          TD_SECTOR;
        diskreq->iotd_Req.io_Data =  (APTR)Buffer;
        diskreq->iotd_Req.io_Command =         ETD_READ;
        diskreq->iotd_Count =                  dcc;
        offset = TD_SECTOR*(sec+NUMSECS*hd+NUMSECS*NUMHEADS*cyl);
        diskreq->iotd_Req.io_Offset =          offset;
        DoIO (diskreq);
        return();
}

MotorOn()
{
        diskreq->iotd_Req.io_Length =          1L;
        diskreq->iotd_Req.io_Command =         TD_MOTOR;
        DoIO (diskreq);
}

MotorOff()
{
        diskreq->iotd_Req.io_Length =          0L;
        diskreq->iotd_Req.io_Command =  TD_MOTOR;
        DoIO (diskreq);
}
```