Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

Moderní počítačová infiltrace

Bakalářská práce

AUTOR: **Igor Hák**

Obecná informatika

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně, s použitím uvedené literatury.

V Hradci Králové dne 20.7.2003 **Igor Hák**

Poděkování

Speciální poděkování patří Petru „Baďas“ Kurtinovi ze společnosti Alwil Software, který byl neustále k dispozici prostřednictvím služby ICQ do pozdních nočních (nebo-li brzkých ranních) hodin a ochotně komentoval veškeré nové odstavce, které postupně tvořily tuto bakalářskou práci.

Dále pak Petru Odehnalovi ze společnosti Grisoft, s.r.o, který celý dokument okomentoval a navrhl řadu nových nápadů a změn. Kromě toho objevil několik dalších slohových nedostatků.

Úvod

Výběr tématu „Moderní počítačová infiltrace“ nebyl náhodný. O „viry“ se zajímám od vánoc 1993, kdy jsem společně s novým PC 386SX získal nevědomky virus Yankee_Doodle.TP-44. Zmiňovaný virus na sebe upozorňoval pravidelně každý den v 17:00, kdy na interní PC speaker hrával melodii „Yankee Doodle“. Tato, ale i následující skutečnosti (proces dezinfekce, příchod viru One_Half.3544.A atd.) mě natolik fascinovaly, že se o tuto problematiku zajímám dodnes.

Počátkem roku 1998 jsem vytvořil první web stránky o virech, které se postupně vyvíjely až po dnešní verzi, tak jak ji známe z www.viry.cz. Potěšující je podpora ze strany tuzemských AV společností a distributorů, za co bych jim chtěl tímto poděkovat.

Stránky viry.cz s podtitulkem „Igiho stránka o virech“ za ta léta značně zpopulárněly. Ač jsem jakožto autor stránek viry.cz potěšen obrovskou návštěvností v dobách největších virových incidentů, na druhé straně zamrzí rozdíl mezi průměrnou návštěvností „v dobách klidu“ a v těchto extrémech – je i 13x vyšší[1] ! Lidé tak ve většině případů přicházejí na stránky až v době po infekci PC a hledají poslední naději v podobě jednoúčelových antivirů.

Autor se účastní řady seminářů a konferencí ať už jako host nebo vystupující (především akce Security v Praze – www.security2003.cz) a soukromých akcí společností Grisoft software („Prase“) a Eset („Žranica“).

Poslední úprava: 21.8.2003

[1] K tomuto došlo během úspěšného šíření viru Win32/BugBear.B.

Obsah

Počítačová infiltrace

Počítačovou infiltrací nazveme jakýkoliv neoprávněný vstup do počítačového systému. Jde o termín s velice širokým významem, účel této publikace omezí působení jen na některé z nich: viry, červy, trojské koně, backdoory. Výše uvedené typy lze označit i názvem MALWARE – MALicious softWARE, škodlivý software.

Pojem „virus“ se zapsal do podvědomí lidí nejvíce a proto jsou takto často označovány veškeré typy infiltrací, bez ohledu na to, zda jde opravdu o virus, trojského koně nebo červa. Ani některé pasáže této publikace se tohoto označení „všeho“ nevyvarovaly.

Pokud není jinak uvedeno, veškeré následující informace se týkají virů pro operační systémy společnosti Microsoft a programů s nimi souvisejících.

1 Základní dělení

1.1 Viry

Jde o nejčastější formu infiltrace, přičemž název je odvozen díky jistým podobnostem od biologických originálů. Virus je schopen sebe-replikace, tedy množení sebe sama, ovšem za přítomnosti vykonatelného hostitele k němuž je připojen. Hostitelem mohou být například spustitelné (executable) soubory, systémové oblasti disku, popřípadě soubory, které nelze vykonat přímo, ale za použití specifických aplikací (dokumenty Microsoft Wordu, skripty Visual Basicu apod.). Jakmile je tento hostitel spuštěn (vykonán), provede se rovněž kód viru. Během tohoto okamžiku se obvykle virus pokouší zajistit další sebe-replikaci a to připojením k dalším vhodným vykonatelným hostitelům.

Párkrát se již stalo, že novináři jásali nad objevením prvního viru na světě, který dokáže infikovat soubory formátu JPG či MP3. V tomto případě nelze mluvit o infekci, ale o prostém připojení zcela nepoužitelného kódu (tělo viru) k výše uvedenému formátu. Jelikož tělo viru nedrží s původním kódem žádnou společnou strukturu, přehrávač (v případě MP3) či prohlížeč (JPG) považují tělo viru za „smetí“. Navíc, JPG i MP3 jsou datové formáty, kdežto tělo viru binární kód. Souvisejícím tématem je pochopení rozdílu mezi formátem souboru a jeho příponou. Pokud hovoříme o formátu, pak hovoříme o vnitřní struktuře souboru a zvolená přípona mu nemusí ve skutečnosti odpovídat. Proto mohou být k vidění případy, kdy je infikován například soubor s příponou .DAT (jinak zcela nezajímavý ze strany virů), ale to jen z důvodu, že jeho vnitřní struktura odpovídá specifikaci formátu EXE souboru.

Podle typu hostitele a způsobů infekce lze viry rozdělovat do dalších skupin.

1.2 Trojské koně (Trojani)

Narozdíl od virů není tento typ škodlivého kódu schopen sebe-replikace a infekce souborů. Trojský kůň nejčastěji vystupuje pod spustitelným souborem typu EXE, který neobsahuje nic jiného (užitečného), než samotné „tělo“ trojského koně. Odtud společně se skutečností, že trojan není připojen k žádnému hostiteli plyne, že jedinou formou dezinfekce je odmazání dotyčného souboru. Starší definice říkají, že trojan je program, vizuálně vypadající jako užitečný, ve skutečnosti však škodlivý. V daleké minulosti se tak několikrát objevil trojský kůň vydávající se za antivirový program McAfee VirusScan, ve skutečnosti likvidující soubory na pevném disku. Jako jeden z posledních dodržujících tuto tradici byl trojan Telefoon, který se vydával za komprimační program RAR 3.0 (jehož

oficiální verze 3.0 mimochodem vyšla až za několik let). V současnosti se tak můžeme setkat nejčastěji s následující formou trojanů, nikoliv však v takové četnosti jako v případě virů:

1.2.1 Password-stealing trojani (PWS)

Skupina trojských koní, která obvykle sleduje jednotlivé stisky kláves[2] (key-loggers) a tyto ukládá a následně i odesílá na dané e-mailové adresy. Majitelé těchto e-mailových adres (nejčastěji samotní autoři trojského koně) tak mohou získat i velice důležitá hesla.

1.2.2 Destruktivní trojani

Klasická forma, pod kterou je pojem trojských koní obecně chápán. Pokud je takový trojský kůň spuštěn, pak likviduje soubory na disku, nebo ho rovnou kompletně zformátuje. Do této kategorie můžeme zařadit i většinu BAT trojanů, tj. škodlivých dávkových souborů s příponou BAT. V tomto případě může překvapit snad jen občasné jednoduché kódování obsahu, díky čemuž není na první pohled zřejmé, co takový kód provádí.

1.2.3 Droppers

Dropper – „vypouštěč". Nese ve svém těle jiný škodlivý kód (například virus), který je vypuštěn po aktivaci trojanu do systému.

1.2.4 Backdoory

Speciální skupina trojských koní, detailněji popsána níže.

Z výše uvedené charakteristiky je zřejmé, že setkání s trojským koněm není běžnou záležitostí. Pravděpodobnost setkání je ovšem vyšší od chvíle, kdy řada virů začala nést takové trojské koně s sebou a během svého šíření je vypouští na jednotlivé počítače. Pokud je virus úspěšný, dojde jak k masivnímu rozšíření viru, tak i k vypuštění velké spousty trojských koní. Nejčastěji jsou vypouštěni password-stealing trojani a backdoory. Potom už záleží jen na autorovi daného viru, jak dokáže získané možnosti využít.

1.3 Backdoory

Jde o aplikace typu klient-server, které jsou schopnostmi velice podobné komerčním produktům jako pcAnyWhere, VNC či Remote Administrator. Narozdíl od nich ovšem vystupují anonymně, uživatel není schopen jejich přítomnost běžným způsobem vypozorovat a to je důvodem, proč jsou preventivně detekovány antiviry jako jeden z typů infiltrace.

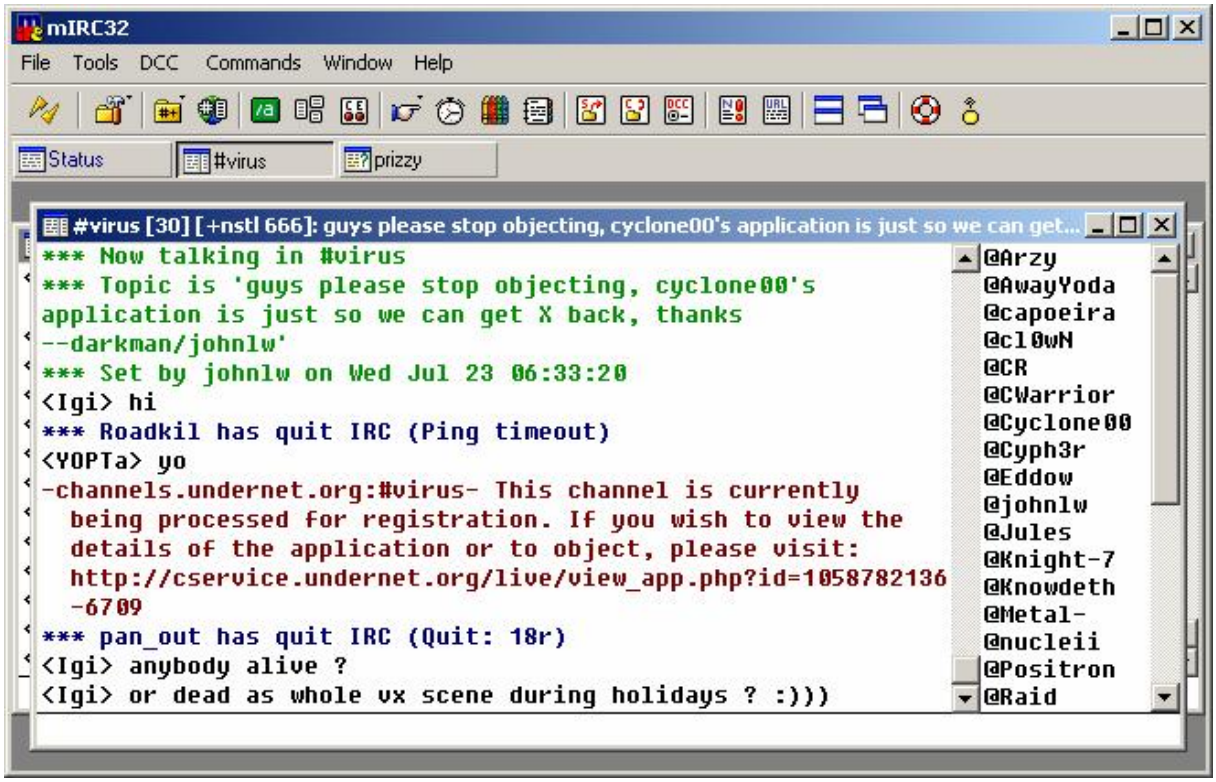
Backdoor je tak aplikace, sloužící pro vzdálenou správu PC a sama osobě nemusí být škodlivá. Záleží pouze na osobě, která tuto vzdálenou správu vykonává. Pokud půjde o činnost škodlivou, pak tuto osobu nazýváme vzdáleným útočníkem. Princip fungování backdooru je následující. Klientská část vysílá požadavky útočníka serverové části, ta tyto požadavky plní, popřípadě odesílá zpět klientu požadované informace. Z předchozího je zřejmé, že klientskou část aplikace by měl vlastnit útočník a serverová by měla být

[2] Uvedené programy mají v řadě případů problémy s českým jazykem a vedlejším efektem bývá, že například při pokusu napsat velké písmeno s háčkem dostáváme následující výsledek: chceme napsat „Č", dostáváme: „ˇC".

umístěna na počítači, kde lze očekávat kupříkladu důležitá data. Pokud je serverová část backdooru vypouštěna úspěšně se šířícím virem, má vzdálený útočník k dispozici tisíce počítačů, ke kterým může vzdáleně přistupovat[3]. Celá komunikace probíhá ve většině případů na bázi TCP/IP, která ve spojení s celosvětovou sítí Internet umožňuje, aby útočník byl vzdálen tisíce kilometrů od serverové části backdooru.

1.3.1 IRC

Zvláštní skupinou jsou pak backdoory (nemusí jít nutně o ně), komunikující s útočníkem skrze domluvený kanál v síti IRC. Jako příklad jmenujme virus Win32/Anarxy, který se snaží z infikovaného PC připojit ke kanálu #iworm_anarxy_channel. V něm vystupuje jako „bot", na první pohled jevící se jako skutečná osoba, chatující na IRC. Útočník tak má teoreticky pohromadě všechny instance viru běžících ve světě a k libovolné z nich se může zalogovat a domluvenými rozkazy ji vzdáleně ovládat.



Obrázek 1 Ukázka komunikace prostřednictvím aplikace mIRC32

1.4 Červi (worms)

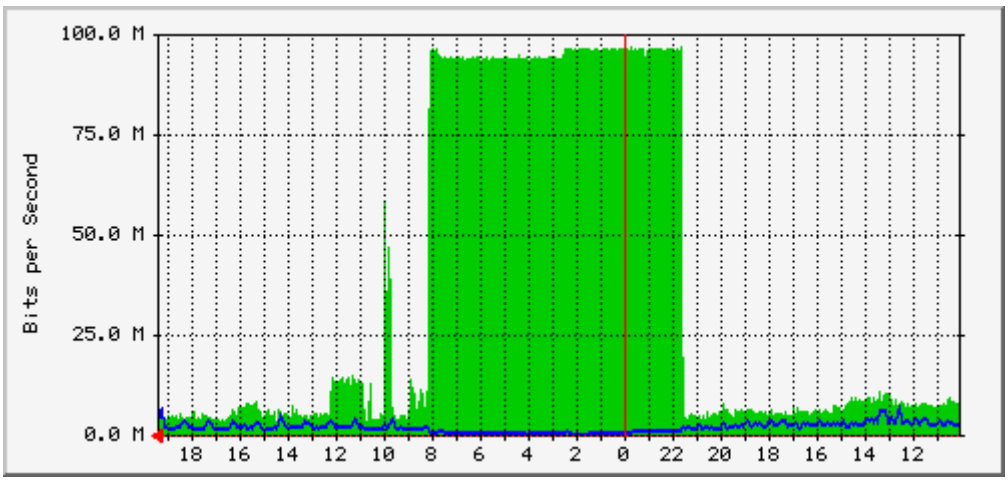
Pojmem červ (worm) byl prvně označen tzv. Morrisův červ, který v roce 1989 dokázal zahltit značnou část tehdejší sítě, ze které později vznikl Internet. Tento a další červi (z poslední doby třeba populární Code Red či SQL Slammer) pracují na nižší síťové úrovni nežli klasické viry. Nelze je tak spatřit ve formě infikovaných souborů, ale pouze jako síťové pakety. Jmenované pakety jsou směrovány již od úspěšně infikovaného systému na další systémy v síti Internet (ať už náhodně, nebo dle určitého klíče). Pokud takový paket dorazí k systému se specifickou bezpečností dírou, může dojít k jeho infekci a následně i k produkci dalších „červích" paketů. Šíření červa je tedy postaveno na zneužívaní konkrétních bezpečnostních děr, úspěšnost pak od rozšířenosti daného

[3] V praxi jsou tyto hodnoty nižší, například díky přítomnosti firewallů, popřípadě díky omezením plynoucích z použití privátních IP adres v rámci sítě LAN.

softwaru obsahující zneužitelnou bezpečnostní díru. Z výše uvedených charakteristik plyne, že červy nelze detekovat klasickou formou antivirového softwaru. Vedlejším efektem může být kompletní zahlcení sítě, podnikové LAN nevyjímaje.

Praktickým příkladem z poslední doby může být červ SQLSlammer, který zneužíval bezpečností díru v aplikaci Microsoft SQL Server. Pokud UDP pakety na portu 1433 o délce 376 bajtů (což je zároveň velikost celého červa SQLSlammer) dorazily k MS SQL Serveru s nezáplatovanou bezpečnostní dírou („Buffer Overruns in SQL Server 2000 Resolution Service Could Enable Code Execution"), došlo díky podtečení zásobníku (buffer underrun) k jeho infekci. SQLSlammer se usadil rezidentně v paměti a začal generovat a následně i rozesílat další spoustu UDP paketů na náhodné IP adresy.

Jmenovaný červ se masově rozšířil a bylo jen štěstí, že neprováděl žádnou destruktivní činnost. Jedinou viditelnou nepříjemností byla schopnost 100% zahltit celou LAN díky obrovské produkci UDP paketů – za 12 hodin bylo dokonce jedno infikované PC s dostatečně dobrým připojením schopno proskenovat všechny veřejné IP adresy celé sítě Internet ! Zajímavostí je, že záplata pro zneužitou bezpečnostní díru byla ze strany Microsoftu uvolněna již několik měsíců před incidentem, ale i tak došlo k úspěšnému rozšíření[4].



Obrázek 2 Graf ukazující extrémní vytížení LAN 100 Mbit/s sítě UDP pakety

Pojem „červ" je často spojován i s typem infiltrace šířící se elektronickou poštou. Zde tak můžou pojmy „virus" a „červ" splývat v jeden. Osobně budu v dalších kapitolách označovat infiltraci šířící se elektronickou poštou za virus.

2 Speciální případy

I když v následujících případech nemusí jít přímo o infiltraci, jde přinejmenším o nepříjemné záležitosti.

2.1 Spyware

Spyware je program, který využívá Internetu k odesílání dat z počítače bez vědomí jeho uživatele. Narozdíl od backdooru jsou odcizovány pouze „statistická" data jako přehled navštívených stránek či nainstalovaných programů. Tato činnost bývá odůvodňována snahou zjistit potřeby nebo zájmy uživatele a tyto informace využít pro

[4] I když je dnes SQLSlammer pro zkušenější správce mrtvou záležitostí, pohledem do reportu firewallů lze snadno zjistit, že pakety o délce 376 bajtů na portu 1433 příslušející červu, se stále šíří Internetem.

cílenou reklamu. Nikdo však nedokáže zaručit, že informace nebo tato technologie nemůže být zneužita. Proto je spousta uživatelů rozhořčena samotnou existencí a legálností spyware. Důležitým poznatkem je, že spyware se šíří společně s řadou sharewarových programů a jejich autoři o této skutečnosti vědí.

2.2 Hoax

Slovem Hoax označujeme poplašnou zprávu, která obvykle varuje před neexistujícím nebezpečným virem. Šíření je zcela závislé na uživatelích, kteří takovou zprávu e-mailem obdrží. Někteří se mohou pokusit varovat další kamarády či spolupracovníky a jednoduše jim poplašnou zprávu přeposlat (forwardovat). Tím vzniká proces šíření. Pokud budeme hovořit o poplašných zprávách týkajících se virů, pak je můžeme charakterizovat následně:

- **Popis nebezpečí (viru)**

  Smyšlené nebezpečí (virus) bývá stručně popsané, v případě viru bývá uváděn i způsob šíření.

- **Ničivé účinky viru**

  Zde záleží převážně na autorově fantazii. Ničivé účinky mohou být celkem obyčejné, třeba zformátování disku nebo už míň důvěryhodné - zběsilý útěk myši do ledničky, roztočení HDD opačným směrem, výbuch počítače... Autoři hororů zde mohou hledat inspiraci.

- **Důvěryhodné zdroje varují**

  Ve většině případů se pisatel poplašné zprávy snaží přesvědčit, že varování přišlo od důvěryhodných zdrojů ("IBM a FBI varují" nebo "Microsoft upozorňuje" atd...)

- **Výzva k dalšímu rozeslání**

  Tento bod HOAX vždy obsahuje! Mnoho nezkušených uživatelů se nechá zprávou napálit a bez přemýšlení výzvu uposlechnou. Díky tomu se tyto nesmysly lavinovitě šíří.

2.3 Dialer

Dialer je program, který změní způsob přístupu na Internet prostřednictvím modemu. Místo běžného telefonního čísla pro Internetové připojení přesměruje vytáčení na čísla se zvláštní tarifikací až 60 Kč / minutu. V některých případech se tak děje zcela nenápadně, zvlášť když oběť používá špatně nastavený Internetový prohlížeč. Dialer (nejčastěji soubor typu EXE a několik pomocných souborů) je obvykle na PC vypuštěn za využití technologie ActiveX, takže problémy mohou nastat uživatelům Internet Exploreru. Ve všech případech nemusí jít nutně o ilegální program. Můžou totiž sloužit jako způsob zpoplatnění určité služby (například přístup na porno stránky).

3 Souborové viry pod Win32

Viry pro Win32, tj. pro Windows 9x / Me / NT (2000, XP atd.) jsou nejčastější dnešní formou infiltrace. Funkčnost Win32 virů zajišťuje Win32 API, (Application Program Interface) tedy množina funkcí operačních systému Windows 9x a NT (včetně 2000, 2003, XP). Prvně se Win32 API objevilo v operačním systému Windows NT a v omezeném

rozsahu i ve Windows řady 9x. To je zároveň i jeden z důvodů, proč jsou některé Win32 viry nekompatibilní mezi řadou Windows 9x a NT. Nová éra 32-bitových Windows přinesla i nový spustitelný formát souborů – Portable Executable (PE).

3.1 Portable Executable

Struktura spustitelných souborů PE hraje při šíření Win32 virů významnou roli a proto ji je nutné podrobněji poznat. Struktura PE souborů se vůči DOS EXE souborům značně zkomplikovala.

3.1.1 PE hlavička

Hlavička (header) popisuje kompletně daný soubor, kterého je součástí. DOS stub je krátkým DOS EXE programem, který při aktivaci z OS DOS obvykle vypisuje „This program cannot be run in DOS mode". Pokud je PE soubor spuštěn z Windows 9x/NT, je DOS stub rovnou přeskočen a realizována 32-bitová část. V hlavičce PE souborů se nachází z pohledu infiltrace několik důležitých atributů:

- **NumberOfSections**

  Označuje počet sekcí v PE souboru. Každá sekce obsahuje navzájem podobné informace / data. Pokud virus využívá metodu infekce, kdy přidá novou sekci, musí zvýšit i tuto hodnotu

- **SizeOfCode**

  Obsahuje údaj o velikosti všech sekcí. Virus by měl tento údaj při infekci zvýšit, ale v řadě případů se tak neděje.

- **AddressOfEntryPoint**

  Adresa, na které začíná spustitelný kód. Je vyjádřena relativně (RVA – Relative Virtual Address) od níže uvedené hodnoty.

- **ImageBase**

  Při spuštění PE souboru je obraz namapován na místo v paměti, které je specifikováno právě virtuální adresou, uvedenou v ImageBase. Obvykle jde o hodnotu 0x400000. Viry využívají tento údaj pro přesné vypočítání pozice určitého prvku.

- **SectionAlignment**

  Každá sekce PE souboru začíná v paměti na virtuální adrese, která je násobkem tohoto atributu. Obvykle jde o 64 KB (0x10000). Většina virů využívá tento údaj pro odvození vhodného místa pro uložení vlastního těla.

- **FileAlignment**

  Jde o podobnou záležitost jako v předchozím případě. Na násobcích této hodnoty začínají „surová" data.

- **SizeOfImage**

  Velikost obrazu PE souboru v paměti, zaokrouhlena nahoru na nejbližší násobek SectionAlignment. Nesprávný přepočet této hodnoty po infekci souboru způsobí,

že virus nebude schopen provozu pod Windows NT. Operační systém řady Windows 9x/Me totiž tuto hodnotu narozdíl od NT nekontroluje.

- **Checksum**

  V tomto atributu by měl být správně uložen kontrolní součet souboru, ale v řadě případů je prázdný. Celý proces infekce PE souborů je navíc usnadněn tím, že Windows 9x/Me tento atribut opět nekontroluje.

Field Name	Data Value	Description	Field Name	Data Value	Description
Machine	014Ch	i386®	Section Alignment	00001000h	
Number of Sections	0003h		File Alignment	00000200h	
Time Date Stamp	3B7D8410h	17/08/2001 20:52:32	Operating System Version	00010005h	5.1
Pointer to Symbol Table	00000000h		Image Version	00010005h	5.1
Number of Symbols	00000000h		Subsystem Version	00000004h	4.0
Size of Optional Header	00E0h		Win32 Version Value	00000000h	Reserved
Characteristics	010Fh		Size of Image	0001F000h	126976 bytes
Magic	010Bh	PE32	Size of Headers	00000400h	
Linker Version	0007h	7.0	CheckSum	00024939h	
Size of Code	00012800h		Subsystem	0002h	Win32 GUI
Size of Initialized Data	00009400h		Dll Characteristics	8000h	Terminal Server aware
Size of Uninitialized Data	00000000h		Size of Stack Reserve	00040000h	
Address of Entry Point	00012475h		Size of Stack Commit	00001000h	
Base of Code	00001000h		Size of Heap Reserve	00100000h	
Base of Data	00014000h		Size of Heap Commit	00001000h	
Image Base	01000000h		Loader Flags	00000000h	Obsolete
			Number of Data Directories	00000010h	

Obrázek 3 Přehled atributů hlavičky PE souboru

3.1.2 Section Table

Mezi PE hlavičkou a programovými daty se nachází tzv. Section Table. Tato tabulka obsahuje základní informace o každé sekci daného PE obrazu. Jak již bylo řečeno, sekce oddělují navzájem odlišné části a naopak spojují podobné (spustitelný kód, data atd.). Modifikací Section Table virus specifikuje svou vlastní novou sekci, popřípadě zajistí možnost umístění se do některé stávající sekce. Z tohoto hlediska jsou pro virus nejdůležitější následující údaje:

- **VirtualSize**

  Udržuje velikost dané sekce před zaokrouhlením dle FileAlignment.

- **SizeOfRawData**

  Velikost dané sekce po zaokrouhlení na násobek FileAlignment.

- **Charakteristika (není přímo názvem atributu)**

  Uchovává příznaky, které indikují vlastnosti dané sekce. Příkladem mohou být: readable, writeable, code, data.

Nejvíce informací je soustředěno do sekce .text. Velice důležitou sekcí z pohledu virů je .idata, která obsahuje Import Table. Podobně důležitou je i sekce .edata, která obsahuje seznam všech API, které aplikace exportuje (nabízí) ostatním externím programům (Export Table).

Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
.text	000126B0h	01001000h	00012800h	00000400h	60000020h	Import Table; Debug data; Import ...
.data	0000101Ch	01014000h	00000A00h	00012C00h	C0000040h	
.rsrc	000089A0h	01016000h	00008A00h	00013600h	40000040h	Resource Table

Obrázek 4 Příklad seznamu jednotlivých sekcí a informací

3.1.3 Import Table

RVA	Name	RVA	Hint	Name
01012E42h	SHELL32.dll	01001020h	0067h	GetModuleHandleA
01012F60h	msvcrt.dll	01001024h	002Eh	LoadLibraryA
01012FFCh	ADVAPI32.dll	01001028h	0089h	GetProcAddress
01013104h	KERNEL32.dll	0100102Ch	00D8h	GlobalCompact
010132DCh	GDI32.dll	01001030h	00D7h	GlobalAlloc
010136A4h	USER32.dll	01001034h	00DEh	GlobalFree
		01001038h	00E5h	GlobalReAlloc
		0100103Ch	0093h	lstrcmpW
		01001040h	0029h	Sleep

Obrázek 5 Příklad import table - vlevo příslušný DLL, vpravo konkrétní funkce

Import Table v sekci .idata udržuje jména všech importovaných DLL včetně jmen funkcí z těchto souborů[5]. V praxi to znamená, že v Import Table jsou uloženy všechny externí funkce, které daná aplikace v aktuálním PE souboru potřebuje ke svému chodu. Import Table je aktualizována při každém spuštění PE souboru a na základě požadavků aplikace jsou jednotlivé externí funkce importovány. Pokud system loader, který se o tuto údržbu stará, nenalezne požadovanou externí funkci, pak se lze dočkat hlášení „Nanalezen soubor xxx.dll“ a aplikace nebude vykonána. Následující obrázek zachycuje Import Table souboru CALC.EXE, tedy klasické aplikace „Kalkulačka“ pod Windows. Na příkladě je zřejmé, že Kalkulátor ke svému chodu potřebuje dalších 6 DLL souborů, přičemž například z KERNEL32.DLL, jedné z nejdůležitějších DLL souborů operačního systému jich je voláno více než dost[6].

Jakmile aplikace volá externí funkci (tj. z jiného DLL souboru) funkcí CALL, není tato volána přímo, nýbrž přes instrukci JMP DWORD PTR [XXXXXXXXh], nacházející se nejčastěji v .text sekci. Aby toho nebylo málo, JMP se odkazuje na adresu v Import Table (do sekce .idata) a až tam dochází k přesměrování na vstupní adresu (entry) požadované funkce v externím DLL souboru.

3.1.4 Export Table

Z pohledu virů je Export Table důležitá především v případě souboru KERNEL32.DLL, tedy DLL knihovny, bez které by nebyl schopen provozu ani samotný operační systém Windows. KERNEL32.DLL obsahuje obrovské množství API funkcí operačního systému a jednou z velice důležitých je GetProcAddress. Tato funkce umožňuje získat vstupní adresy (entry address) k dalším libovolným funkcím. Z pohledu virů k dalším funkcím, které virus potřebuje pro svou úspěšnou replikaci (například FindFirstFileA, GetCurrentDirectoryA) a vykonávání všech ostatních činností. Export Table se skládá ze tří seznamů: Function Address Table, Function Name Table a Function Ordinal Table.

3.1.4.1 Zjišťování vstupních adres funkcí API

I když by na první pohled mohlo jít o jednoduchou činnost a aplikace či dokonce virus by si prostě po potřebné API funkci „sáhly“ na určitou, přesně danou adresu, ve skutečnosti je situace daleko komplikovanější. Umístění jednotlivých funkcí v paměti a tím i hodnoty vstupních adres pro každou funkci se totiž liší s každou novou verzí operačního systému Windows a to dokonce i v rámci jedné „řady“ (například starší Windows 95 a Windows 95 SR2). Výsledkem jsou základní dvě metody:

- **Hard-coded metoda**

[5] Import Table je významově přibližně to samé, jako Interrupt Vector Table v operačním systému DOS.
[6] Seznam v pravém sloupci je neúplný.

Tuto metodu využívají některé amatérské pokusy o vytvoření Win32 viru. Stejnou metodu využíval i naprosto první virus pro Windows 95 a tím byl virus Win95/Boza.A. Adresy externích volaných API funkcí jsou v tomto případě pevně nastaveny v těle viru (hard-coded) a díky výše uvedeným skutečnostem může být takový virus schopen provozu jen pod určitou skupinou operačních systému (v případě jmenovaného viru jen Windows 95 Beta). Příkladem může být viry hojně využívaná API funkce GetCurrentDirectoryA. Zatímco v anglické verzi Windows 95 byla vstupní adresa této API funkce 0x00007744, v maďarské 0x0000774C. Je zřejmé, že pokud virus „šáhne" vedle v případě strategicky důležitých funkcí, nebude schopen rozumného šíření.

- **Metoda využití API funkce GetProcAddress**

  Metoda využívaná všemi moderními viry a dalšími infiltracemi. API funkce GetProcAddress slouží k zjištění adresy požadované API funkce. Ta narozdíl od výše uvedeného případu vždy odpovídá realitě a tak nedochází k přehmatům. Počátečním problémem ovšem je, že je potřeba zjistit adresu samotné API funkce GetProcAddress. Možností, jak k ní přijít, je prohlídka Export Table souboru KERNEL32.DLL a vyhledání řetězce „GetProcAddress" v části Function Name Table a následně i odpovídající adresy v Function Address Table.

Podobné metody existují i při hledání pozice samotného, v paměti namapovaného souboru KERNEL32.DLL. Inteligentnější viry využijí výstup funkce API CreateProcess, která je automaticky volána při spuštění PE souboru a která volá (CALL) požadovanou aplikaci. Během posledně jmenované operace se do zásobníku uloží adresa následující instrukce API funkce CreateProcess, tedy zároveň adresa vyskytující se někde uvnitř KERNEL32.DLL. Za využití různých algoritmů lze dojít až k Export Table či Import Table.

3.2 Metody infekce

3.2.1 Overwrite metoda

Podobně jako v případě DOS virů jde o nejprimitivnější formu infekce souboru. Jeho původní počáteční část je přepsána tělem viru a zároveň tak znehodnocena.

3.2.2 Parazitická metoda - append

Obecně lze skupinu virů, využívajících tuto metodu označit za parazitické, tedy takové, kdy při infekci spustitelných souborů nedochází k jejich trvalému poškození. Velice důležité je uložení některých hodnot PE hlavičky (ostatní lze dopočítat) během infekce do těla viru. Pokud je infikovaný PE spuštěn, dojde k aktivaci viru dle níže uvedených metod, ale následně i k rekonstrukci souboru do neinfikovaného tvaru a vykonání původního programového kódu[7]. Mluvíme o tzv. předání řízení hostitelskému programu. Uložené hodnoty mohou být podobným způsobem využity i k rekonstrukci souboru antivirovým programem při případné dezinfekci viru. S odstupem času můžeme parazitickou metodu rozdělit na několik dílčích.

[7] K tomuto procesu jsou nutné právě některé hodnoty PE hlavičky z doby, kdy byl soubor v neinfikované podobě. Ostatní hodnoty lze dopočítat, soubor v paměti obnovit do původní podoby a vykonat i původní kód programu, který byl náplní PE souboru před infekcí. Nedojde tak k podezření a kladení otázky „Proč se po spuštění nic neděje".

3.2.2.1 Přidání nové sekce

Virus může vhodnou úpravou Section Table vytvořit zcela novou sekci, do které uloží své tělo a dále zajistí i jeho aktivaci při dalším spuštění čerstvě infikovaného PE souboru. Výsledkem je modifikace atributu AddressOfEntryPoint a NumberOfSections.

3.2.2.2 Připojení ke stávající sekci

Virus v tomto případě upraví hlavičku v Section Table vhodné sekce tak, aby se do ní vešel i s původním programem. V některých případech je virus rozdělen i do více sekcí. Důležitou úlohu hrají především atributy VirtualSize, SizeOfRawData a SizeOfImage. V závislosti na tom, zda virus modifikuje atribut AddressOfEntryPoint lze tuto metodu dále rozdělit:

- **Infekce s modifikací AddressOfEntryPoint**

  Atribut AddressOfEntryPoint je během infekce upraven tak, aby ukazoval na začátek těla viru.

- **Infekce bez modifikace AddressOfEntryPoint**

  Specialitou některých virů je pak vyhýbaní se modifikaci výše uvedeného atributu. Virus v tomto případě vypočítá, kam ukazuje původní AddressOfEntryPoint a na toto místo umístí instrukci JMP, která se postará o skok (jump) na tělo viru. Uvedenou metodou může dojít k mírnému ztížení detekce antivirovým skenerům, založených na emulaci kódu. Ztížení může být umocněno, pokud je instrukce JMP vložena až pár desítek bajtů za místo, na které ukazuje AddressOfEntryPoint. Takové viry jsou daleko komplikovanější, jelikož si musí dát pozor na případné relokace směřující do přepsané části kódu.

3.2.2.3 Infekce hlavičky

V tomto případě je tělo viru umístěno mezi konec PE hlavičky a začátek první sekce, přičemž AddressOfEntryPoint ukazuje na začátek těla viru. Úspěšná infekce závisí na velikosti prostoru, který je odvozen od počtu záznamů v Section Table a hodnoty FileAlignment. Pokud totiž bude infikovaný PE soubor rozsáhlejšího charakteru, pak se může v Section Table nacházet velké množství definovaných sekcí, díky čemuž bude dostupný prostor nedostatečně velký.

Zajímavostí je, že Windows dovolí takový soubor bez problému spustit, i když atribut AddressOfEntryPoint nesměřuje do žádné sekce PE souboru.

3.2.2.4 Cavity „mezerová“ infekce

Uvedenou metodou se v historii snad nejvíce proslavil virus Win32/CIH, který byl novináři nešťastně pojmenován Černobylem[8].

Volné prostory, vznikající mezi sekcemi díky násobkům atributu FileAlignment, na kterých mohou další sekce začínat, jsou místy („jeskyněmi“ – caves), kam se může tato skupina virů schovávat. Vzhledem k délce typických virů je nutné tělo rozdělit do více takových oblastí (tj. za každou sekci). Při tomto způsobu infekce je hodnota atributu VirtualSize srovnána na SizeOfRawData pro každou postiženou sekci.

[8] Virus Win32/CIH totiž ve stejném datu, jako vybuchla jaderná elektrárna Černobyl, vykonával škodlivou akci. Formátoval část disku a pokud to bylo možné, přepsal na základní desce FlashBIOS. Kromě toho, že došlo k jeho masivnímu rozšíření, proslavil se i jako první virus, který dokáže poškodit hardware (i když většinou ne trvale).

Výsledkem využití této metody je, že infikovaný soubor má totožnou velikost jako původní originál.

3.2.2.5 Infekce DLL, speciálně KERNEL32.DLL

Infekcí DLL souboru, speciálně KERNEL32.DLL si může virus zajistit částečnou paměťovou rezidentnost. Z výše uvedených poznatků plyne, že nejdůležitější oblastí bude tzv. Export Table, tj. seznam, obsahující seznam všech API, které daný DLL exportuje (nabízí) ostatním aplikacím. Vhodnou úpravou frekventovaně využívaných API funkcí lze docílit toho, že kromě vykonání dotyčné funkce při jejím voláním jiným programem, dojde i k aktivaci viru. Během tohoto okamžiku nastává vhodná chvíle pro další replikaci viru, popřípadě k vykonání vedlejších činností.

Jistou překážkou v infekci je přítomnost kontrolního součtu v každém DLL. Zatímco Windows 95 tento kontrolní součet ignoroval, Windows NT již nic nenechává náhodě. Aby nedošlo k chybě při zavádění daného DLL souboru v případě, kdy je kontrolní součet rozdílný, je nutné tuto kontrolu obejít. Jednou z možností je tento kontrolní součet přepočítat ihned po infekci za využití API funkcí z IMAGEHLP.DLL.

Ve všech případech nemusí jít přímo o infekci DLL, ale pouze vnitřní úpravu takového souboru (modifikace, „opatchování“).

3.2.2.6 VMM & VxD

Plnohodnotnou paměťovou rezidentnost si mohou viry zajistit vhodným využitím VxD (virtuální ovladače zařízení). Jejich kódy pracují na nejvyšší úrovní oprávnění procesoru, zvaném ring 0 a mohou tak se systémem vykonávat cokoliv. Na stejné úrovní pracuje i ovladač IFS manager, přes který procházejí veškeré souborové operace Windows, ať už jsou volány z 16-bitové či 32-bitové aplikace Windows, nebo z virtuálního stroje MS-DOSu. Dokonalý plán snad kazí už jen fakt, že VxD se vyskytuje pouze pod Windows řady 9x/Me.

- **Vlastní VxD ovladač**

  Malá skupina virů s sebou vlekla (tj. virus - dropper) kompletní VxD ovladač, který byl při aktivaci infikovaného souboru vypuštěn do systému ve formě souboru. Po jeho zavedení do paměti získal virus oprávnění ring 0 a napojením se na Installable File System (IFS) získal i dokonalý přehled nad manipulací se soubory.

- **Úprava VMM (Virtual Machine Manager) & IDT (Interrupt Descriptor Table)**

  Ve všech případech jde o jedno: získat oprávnění na úrovni ring 0. Toho lze docílit například i modifikací funkce Schedule_VM_Event ve VMM, která je automaticky volána při spouštění programu. Virus Win32/CIH zase využívá IDT k modifikaci INT3 (debug interrupt). Výsledkem je, že rutina INT 3 je spuštěna na úrovni ring 0 přerušení. Jistě by se našlo několik dalších metod, jak získat oprávnění na úrovní ring 0.

3.2.2.7 Speciální případy

- **Modifikace hodnoty lfanew**

Atribut lfanew drží adresu začátku PE hlavičky a je uložen ještě před ní, v DOS hlavičce. Virus, využívající uvedenou metodu si s sebou nese vlastní „ideální“ PE hlavičku kterou při infekci použije místo originální. Původní hlavičku (včetně hodnoty lfanew) i program obsažený v daném PE souboru uschová. Po spuštění infikovaného souboru tak

dojde k vykonání činnosti viru a následně i vytvoření dočasného souboru s původním obsahem a jeho spuštění.

Díky tomu, že si virus s sebou nese kompletní PE hlavičku, odpadá mu řada starostí okolo importu funkcí.

- Atypické metody

V některých případech může jít o průniky výše uvedených metod. Příkladem může být metoda infekce, kdy virus daný soubor infikuje například metodou appending avšak takovým způsobem, že k úspěšnému odléčení souboru může dojít pouze na tomtéž PC (tj. existuje „klíč“, který vznikne z informací konkrétního PC). Jinou metodou může být zašifrování původního PE souboru a jeho „vlepení“ za tělo viru. K odléčení takového souboru je potřeba opět znát klíč, který může být jak v těle viru, tak i někde na disku, popřípadě vygenerován na základně specifických informací.

3.3 Techniky Win32 virů

3.3.1 EPO - Entrypoint Obscuring

Virus, využívající EPO (Entrypoint Obscuring) techniku nemění hodnotu AddressOfEntryPoint a vlastní aktivaci se snaží co nejvíce oddálit. Toho lze v případě EPO docílit „pověšením“ se na určitou API funkci, kterou původní program během své činnosti volá (tj. je v Import Table). Pokud je taková API funkce zavolána, kromě jejího zpracování dojde i k aktivaci viru. Pokud k volání (a zároveň aktivaci viru) dochází až v pozdějších pasážích původního programu, je velká pravděpodobnost, že emulace kódu antivirových skenerů nebude schopna virus detekovat (vzhledem k časové náročnosti kódu). Proto je velice oblíbenou API funkcí ExitProcess, která se aktivuje těsně před ukončením činnosti PE souboru. Volbou nevhodné API funkce, která je například volána při netypické události původního programu, nemusí dojít k aktivaci viru.

K úpravě volání vhodné API funkce nedochází přímo v Import Table, jelikož je automaticky aktualizována během inicializace před spuštěním PE souboru. Musí tak docházet k průzkumu kódu programu a hledání volání API funkcí (instrukce JMP DWORD PTR [XXXXXXXXh] nebo CALL DWORD PTR [XXXXXXXXh] v závislosti na kompilátoru). Cílová adresa je upravena, aby odskočila na tělo viru ve stejném PE souboru (viz. metoda appending) a následně na původně zamýšlenou API funkci (popř. opačně).

3.3.2 Multithreading

Tato technika využívá schopnosti operačního systému Windows vykonávat více než jeden kód jednoho programu v „jeden čas“. Dokonce by se dalo hovořit o jakémsi multitaskingu v rámci jednoho programu.

Příkladem multithreadingu může být aplikace MS Word. Zatímco v prvním threadu (toku) píšeme dopis, druhý thread právě tiskne. Takto se může v programu provádět více činností v relativně stejném čase.

Některé viry využívají tuto techniku k ochraně před emulátory kódu či heuristickou analýzou. Předpokládá se, že antivirový skener neumí simulovat vykonávání více než jednoho kódu v jeden čas. Podobná situace nastává při trasování kódu. Zatímco dochází k trasování bezvýznamného kódu, kód viru běží v jiném threadu.

Výše uvedený příběh s aplikací MS Word tak můžeme dokončit například následovně. Zatímco v prvním threadu (toku) píšeme dopis, druhý thread právě tiskne a třetí thread infikuje všechny PE soubory na pevném disku.

3.3.3 Multiprocessing & IPC

Využívá schopnosti Win32 platforem mít spuštěno více než jeden program (proces). InterProcess Communication (IPC) je komunikace mezi dvěma a více procesy.

Použití těchto technik se liší snad u každého viru. Virus Win95/BeGemot například umožňuje uživateli komunikovat s virem, pokud je v paměti, pomocí externího programu, textové konzole. Virus to provádí tak, že ve sdílené paměti vytvoří strukturu, která se bude ke komunikaci využívat. Externí program tuto strukturu najde a pomocí dohodnutých pravidel do ní zapisuje a čte z ní. Vstupem jsou povely, na které umí virus reagovat, výstupem jsou výstupní data zobrazená uživateli.

Virus Win32/Vulcano zase umí komunikovat mezi více instancemi stejného viru v paměti. V praxi to znamená, že pokud chce virus infikovat nějaký soubor, tak se pokusí najít "stejný virus spuštěný v jiném programu", a pokud ho najde, přikáže mu, ať infekci provede za něj. Vtip je v tom, že pokud ladíte nějaký program, ladíte pouze a jenom ten jeden program. Stejně jako v případě multithreadingu může tato technika způsobovat problémy při trasování či emulaci kódu.

3.3.4 Stream companion

Streams – datové proudy jsou záležitostí souborového systému NTFS, který je náhradou FAT32 pod operačními systémy řady Windows NT. Každý soubor pod NTFS se skládá z takzvaných streamů. Pokud si vezmeme jakýkoliv soubor, tak na souborových systémech FAT, FAT32 můžeme najít pouze hlavní (nepojmenovaný) stream, který je definován právě jménem souboru. Pod NTFS můžeme k tomuto hlavnímu streamu přiřadit další, pojmenované streamy, jejichž názvy se od jména souboru oddělují znakem „:“ (například: soubor.txt:mujstream) Důležitá je ovšem skutečnost, že běžné prostředky (například notepad, nebo klávesa F3 ve spojitosti „souborových manažerů“) prezentují uživateli pouze hlavní stream a všechny ostatní přičleněné mu zůstanou utajeny. Této skutečnosti využívá první virus této kategorie – Win32/Stream[9]. Proces infekce PE souboru probíhá tak, že k momentálně infikovanému PE je přiřazen stream :STR a do něho nakopírován celý původní obsah PE souboru. Tělo viru je pak uloženo na místo, kde figuroval původní program. Při volání infikovaného souboru dojde nejprve k aktivaci viru a následně i původního programu, který je uložen ve streamu :STR.

Idylka končí s přechodem infikovaného souboru k jinému souborovému systému než NTFS. Pokud dojde například ke zkopírování takového souboru na souborový systém FAT32 (nebo k jeho vypálení na CD), zkopíruje se pouze hlavní stream. Proto nelze větší rozšíření podobných virů očekávat.

3.3.5 SFP disabling

System File Protection (SFP) je novou záležitostí operačního systému počínaje verzí Windows 2000. Uvedená funkce zabraňuje nahrazení či modifikaci chráněných systémových souborů (SYS, DLL, EXE...) a značně tak omezuje i případné viry. K modifikaci či nahrazení může za normálních okolností dojít pouze v případě, že nová verze souboru je náležitě opatřena elektronickým podpisem společnosti Microsoft.

První viry se tak chráněné soubory nepokoušely vůbec infikovat a prostřednictvím údajů od API funkce SfcIsFileProtected se jim jednoduše vyhnuly. Novější se pokouší vypnout SFP prostřednictvím registrů, ale ne vždy to funguje.

[9] Virus Win32/Stream je dílem českých autorů - dvojice Benny/29A a Ratter/29A. V médiích vyvolal silný rozruch.

S nejpřijatelnějším řešením přišla opět česká dvojice Benny/29A a Ratter/29A. Úpravou winlogon.exe a využitím API funkce ExitThread dokážou SFP deaktivovat.

4 Viry šířící se elektronickou poštou

Viry, šířící se elektronickou poštou jsou nejrychlejší formou virové infiltrace. O schopnosti bleskového šíření mohou přesvědčit i grafy v části o antivirovém softwaru. Z praxe plyne, že prostřednictvím e-mailu může dorazit prakticky jakákoliv forma infiltrace, kromě „pravých“ síťových červů. Pokud ovšem hovoříme o infiltraci, která se přímo o rozesílání „infikovaných“ e-mailů stará, můžeme hovořit o:

- virech v binárních souborech (PE),
- skriptových virech,
- makrovirech.

Ve všech třech případech jde pochopitelně o viry, které dokážou manipulovat s elektronickou poštou a rozesílat se prostřednictvím e-mailů ve formě přílohy. Jak již bylo řečeno na počátku, některé osobnosti nazývají tento typ infiltrace jako červy, ale osobně zůstanu u označení virus s ohledem na skutečné červy jako SQL Slammer, Code Red apod.

4.1 Poštovní viry v binárních souborech (PE)

Nejčastější formou je prvně jmenovaná skupina, tedy viry šířící se poštou ve formě PE EXE souborů. Zde je nutno poznamenat, že formát souboru (tj. jeho vnitřní struktura), nemusí odpovídat použité příponě souboru. Díky této skutečnosti se mohou šířit nejen přímo infikované soubory s příponou EXE, ale i SCR, PIF atd. Častým jevem je využití tzv. dvojité přípony (např. soubor.jpg.exe), jelikož při určité konfiguraci systému je zobrazena pouze první z nich a případná oběť tak může propadnout mylné představě, že jde o neškodný soubor (v našem případě JPG, tedy grafický soubor). Mylnou představu může podpořit i interní ikona infikovaného PE souboru.

V řadě případů nejde o klasické parazitické viry, tj. souborové viry, které by infikovaly další souboru na pevném disku. Při spuštění infikovaného souboru v příloze e-mailu tak nejčastěji dochází pouze k vypuštění několika souborů do operačního systému, které se starají o budoucí replikaci prostřednictvím elektronické pošty. Obvykle je nutno zajistit jejich automatickou aktivaci po každém startu PC a jako nejvhodnější metodou se jeví klíč HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run v registrech Windows. Alternativou je i soubor WIN.INI a řádek RUN=, ale ten je využíván pro jiné, níže uvedené účely.

Další činnost závisí na „inteligenci“ daného viru. Jeho úspěch může být odvozen například od toho, zda bude testovat aktuální stav připojení k Internetu. Využít může například API funkci InternetGetConnectedState z WININET.DLL či využít vhodné API funkce z WINSOCK.DLL (jen Windows řady 9x).

Další průběh můžeme zhodnotit v následujících podkapitolách.

4.1.1 Získávání e-mailových adres budoucích obětí

Agresivita viru, s jakou se pustí do získávání dalších e-mailových adres, na které pak hromadně rozešle svoji kopii, může hodně napovědět o případném úspěchu daného viru. V dnešní době jsou známy následující tři základní metody získávání (harvesting) e-mailových adres.

4.1.1.1 Prohledávání vhodných souborů

Jednou z cest, jak získat dostatek e-mailových adres je průzkum HTML souborů na pevném disku. Viry mají v oblibě především adresář Internet Temporary Files, kam Internet Explorer průběžně ukládá stažená data. Jednoduchým algoritmem na vyhledávání řetězců označujících e-mailovou adresu (řetězec „mailto:“) jich lze najít velké množství.

4.1.1.2 Prohledávání WAB (Windows Address Book)

Soubor s příponou WAB obsahuje všechny e-mailové adresy, které si uživatel vede v adresáři aplikace MS Outlook Express. Vlastností novějších verzí Outlooku je skutečnost, že do knihy adres jsou automaticky přidávány i e-mailové adresy, na které jsme někdy v minulosti odpověděli. WAB může být tak velice rozsáhlý co do počtu e-mailových adres.

4.1.1.3 Prohledávání samotných složek pošty

Výsledkem poslední metody je v řadě případů zisk velice „blízkých“ e-mailových adres z pohledu uživatele, u něhož virus průzkum způsobil. Průzkumem složek jako „Doručená pošta“ (INBOX) lze najít velké procento adres, které patří blízkým přátelům. Viry zaslané od blízkých přátel a maskovány dokonalým textem mohou působit velice „psychologicky“[10].

4.1.2 Proces rozesílání / replikace

Obecně lze proces rozesílání rozdělit na

- **masové (mass-mailers) – často zkratka “@mm” na konci názvu viru**
- **pomalé (slow-mailers) – „@m”**

Prvně jmenovaná skupina je běžnější a jedná se o metodu, při které dochází k hromadnému rozeslání kopie infikovaného e-mailu na co možná největší počet adres v co možná nejkratším časovém úseku.

V druhém případě lze hovořit o důmyslném rozesílání v menším rozsahu. Příkladem mohou být některé viry, které odpovídají (reply) na čerstvě došlé e-maily.

Formy, jakou je virus připojen k e-mailu se nabízejí dvě. V případě virů v binární podobě připadá v úvahu první z nich - ve formě souboru jako příloha.

K aktivaci viru v elektronické poště by za normálních okolností docházelo pouze po manuálním rozhodnutí uživatele. Realita je ovšem zcela odlišná a to díky přítomnosti celé řady bezpečnostních děr (security holes) v e-mailových klientech. Pokud nejsou opatřeny příslušnými záplatami, které vydává výrobce daného produktu (v našem případě výrobce e-mailového klientu), pak je virus může zneužít například k svojí vlastní automatické aktivaci pouhým náhledem na infikovaný e-mail. Bližší informace jsou k tomuto hojně zneužívanému problému uvedeny v jedné z následujících kapitol.

K samotnému rozeslání kopií infikovaných e-mailů je pochopitelně potřebný SMTP server a tak se nabízí několik způsobů, jak se k němu dopracovat.

[10] Důkazem může být virus Win32/Bugbear.B, který se dokázal masově rozšířit během několik hodin po celém světě. Název přílohy i text byl poskládán z informací nalezených na pevném disku, takže často byl k vidění velice věrohodně vypadající e-mail.

4.1.2.1 Pevně definované SMTP servery (hard-coded)

Tuto metodu uplatňuje minimum dnešních virů šířících se elektronickou poštou. Důvodů je několik.

- **Pevně definované SMTP servery (nejčastěji prostřednictvím IP adresy) nemusí být dostupné během celé aktivní působnosti viru. Dostupnost SMTP serverů k odesílání e-mailů z vnější sítě (z Internetu, tzv. open relay) je ve většině případů důsledkem nevhodné konfigurace SMTP serveru. Při případném upozornění majitele takového SMTP serveru ze strany antivirových společností, může dojít k nápravě a tím i k zastavení šíření viru.**

- **Ve světě existuje veřejná databáze „Open Relay Database" (www.ordb.org), která ve svých záznamech udržuje právě open relay SMTP servery, tj. servery, které lze libovolně zneužít pro odesílání e-mailů odkudkoliv kamkoliv[11]. Pokud se případný virus bude šířit právě prostřednictvím takového serveru, může dojít k situaci, kdy cílový SMTP server e-mail odmítne s tím, že byl odeslán z open relay serveru.**

4.1.2.2 Uživatelem používaný SMTP server

Tato metoda vychází s využití MAPI rozhraní e-mailového klienta Microsoft Outlook, popřípadě s úpravy či vypuštění vlastního souboru WSOCK32.DLL, za pomocí něhož může virus monitorovat Internetové připojení. V případě zmiňované knihovny mohou být pro virus zajímavé například následující API funkce: send, recv, connect.

4.1.2.3 Vlastní SMTP server

Daleko populárnější je vlastní jednoduchý SMTP server na portu 25, který si virus po případné aktivaci vypustí do operačního systému.

4.1.3 Konání dalších činností

Výčet činností by mohl být velice široký a značně se prolíná s kapitolou „Techniky virů šířících se elektronickou poštou".

4.2 Poštovní skriptové a makroviry

Éra skriptových virů a makrovirů pravděpodobně skončila, ale i tak je vhodné se o nich zmínit. Zatímco makroviry šířící se elektronickou poštou tvoří vždycky soubor v příloze (s příponou DOC, XLS apod.)[12], u skriptových virů již není situace natolik jednoznačná. Malá skupina skriptových virů je totiž přímo součástí zprávy. K tomu by nemohlo nikdy dojít v případě, že by e-mail zůstal ke svému původnímu účelu, tj. k zasílání textových zpráv elektronickou cestou. Nové trendy ovšem působnost značně rozšířily, takže kromě textových zpráv může jít i o multimediální skvosty plné blikajících písmenek, obrázků a hudebních efektů. Z pohledu virů je v tomto směru největším přínosem implementace jazyka HTML přímo do poštovních klientů (především Microsoft Outlook / Express). HTML jazyk a tedy i e-mail může obsahovat vnořené skripty (například VBS, JS), čehož mohou viry využít a šířit se ve formě těchto skriptů (VBS/Bubbleboy, WScript/Kakworm). Alternativou mohou být skriptové viry, které hlavní část vlastního těla soustředí na určitém místě sítě Internet. Při případné aktivaci skriptu si lze HTML tagem IFRAME tuto část vyžádat (viz. www.w3.org).

[11] Často jsou zneužívány pro rozesílání SPAMu.
[12] Obvykle tak jde přímo o odcizený dokument.

Jelikož skriptové jazyky nenabízejí takové možnosti jako v případě assembleru a souvisejících „binárních“ virů, jsou skriptové viry a makroviry jednodušší a čitelnější formy. V řadě případů jsou tak odkázány na MAPI rozhraní poštovních klientů a zároveň tak schopny replikace pouze pod plnohodnotným MS Outlookem a nikoliv Expressem. Proto i následující kapitola se této skupiny virů týká okrajově.

4.3 Techniky virů šířících se elektronickou poštou

4.3.1 Dvojitá přípona

Jedná se často využívanou metodu. Infikovaný soubor v příloze e-mailu má dvojitou příponu, tj. název.př1.př2. Na některých konfiguracích Windows se výše uvedený příklad jeví pouze jako název.př1, přičemž skutečná přípona, ze které operační systém vychází (čte název souborů zprava) zůstává vizuálně utajena. Výsledkem může být mylná představa uživatele, že jde o zcela neškodný soubor a co víc, že jde o více než zajímavý soubor. Reálným příkladem může být úspěšný virus VBS/SST.A, který se šířil v souboru AnnaKournikova.jpg.vbs. Pokud se uživateli zobrazilo jen AnnaKournikova.jpg, jen těžko by někdo odolal pohledu na obrázek krásné ruské tenistky. Idylka skončila ve chvíli, kdy se místo obrázku tenistky spustil škodlivý VBS skript s cílem rozeslat tento „obrázek“ do dalších koutů světa.

4.3.2 „Bílé“ znaky

Jde o alternativu k dvojitým příponám. Pokud by šlo o konfiguraci systému, zobrazující obě přípony, je jistou šancí za konec první z nich vložit velké množství mezer a tak onu druhou posunout za vizuálně viditelnou oblast.

4.3.3 Využívání bezpečnostních chyb

Využívání bezpečnostních děr e-mailových klientů (především MS Outlook / Express) je velice časté. Pokud budeme hovořit konkrétně o společnosti Microsoft, příslušné záplaty (hotfixes) jsou zhotovovány velice brzy po objevení bezpečnostní díry. Avšak jedna věc je zhotovování bezpečnostních záplat a druhá jejich včasná aplikace[13].

Následuje přehled některých, které jsou z pohledu virů zajímavé.

4.3.3.1 Typelib.scriptlet/Eyedog

Této bezpečnostní chyby v Internet Explorer 4.0 a 5.0 využily viry VBS/Bubbleboy a později i úspěšnější WScript/Kakworm. Díky uvedené chybě bylo možno využít ActiveX pro změny v systému uživatele bez jeho svolení. Výše uvedeným virům to stačilo k tomu, aby se v takto nezáplatovaném systému usadily a provedly proces šíření.

Záplata byla dostupná během srpna 1999, VBS/Bubbleboy se objevil v polovině listopadu 1999 a WScript/Kakworm ke konci prosince 1999.

4.3.3.2 Incorrect MIME Header Can Cause IE to Execute E-mail Attachment

Běžně využívanou bezpečnostní dírou je „Incorrect MIME Header Can Cause IE to Execute E-mail Attachment“, která je popsána v Microsoft Security Bulletin (MS01-020).

[13] Příkladem mohou být viry, které se i dnes dokážou šířit jen na základě bezpečnostních děr, pro které již několik let existuje bezpečnostní záplata.

Internet Explorer (5.01, 5.5) díky ní automaticky aktivují soubor v příloze e-mailu, pokud na něj odkazuje nevhodný MIME typ. Příkladem může být následující ukázka:

```
Content-Type: audio/x-wav
Content-Disposition: attachment; filename="Sophos.exe"
Content-Transfer-Encoding: base64
Content-ID: <0EE08CFh43a5fe7504eC>
```

Výsledkem je ono zmiňované automatické spuštění přílohy – souboru Sophos.exe, jelikož Content-Type je „nevhodně“ nastaven na audio/x-wav.

4.3.4 Aktualizace viru prostřednictvím Internetu

Snaha o průběžnou aktualizaci viru z Internetu (podobně jako například u antivirových systémů) není ničím neobvyklým. Úspěšným příkladem může být virus Win32/Hybris, který se z Internetu snažil stahovat speciální plug-iny, které dokázaly rozšířit jeho působnost například o vykreslování černobílé spirály v určitém datu. S aktualizací je však spojeno několik problémů a teorií:

- **Dostupnost aktualizačního serveru. Nelze zaručit dlouhodobou dostupnost aktualizačního serveru, tj. například Internetové stránky, na které se vyskytují potřebné soubory na vykonání aktualizace vzdáleného viru. Stránky jsou obvykle zablokovány po upozornění webmastera daného serveru ze strany antivirových firem.**

- **Vypuštění antiviru ze strany aktualizačního serveru. Pokud je virus schopen aktualizací rozšířit svojí působnost, proč by ji nemohl i zcela ztratit ? Prakticky je „podstrčení“ antiviru na aktualizační server možné, ale z pohledu antivirových firem by šlo o neetické řešení. Dopravit další nevyžádaný kód (i když antivirus) na cizí počítač by si žádná antivirová společnost nedovolila.**

Jisté metody obrany virů proti těmto skutečnostem existují, ale ve většině případů jde o teorii, popřípadě byly provedeny jen první praktické pokusy.

- **Využití elektronického podpisu, kdy jsou všechny aktualizace pro virus elektronicky podepsány a pravost podpisu kontrolována ze strany viru. Tímto se zabrání podstrčení nepravé aktualizace například ze strany antivirové firmy (i když opět hovoříme pouze o teoretické možnosti). Pokud bude aktualizační server odstaven, elektronický podpis nepomůže.**

- **Využití aktualizačních sítí P2P (peer-to-peer). Zvláštností tohoto uskupení počítačů je, že neexistuje žádný centrální server a tím ani žádný, který by stačilo zneškodnit a zabránit tak budoucí aktualizaci viru. Prvním virem, který se snaží této skutečnosti využít je pravděpodobně Win32/Serotonin, opět českého autora – Bennyho/29A. Jmenovaný virus se snaží navazovat kontakty s jinými instancemi ve světě a s nimi pak vyměňovat jednotlivé komponenty[14].**

4.3.5 Vypouštění dalších programů

Nejčastějším případem je vypouštění backdoorů či PWS trojanů (viz. úvodní část o infiltraci) do operačního systému. Nemusí jít nutně o specialitu viru, šířícího se elektronickou poštou. V některých případech jsou tyto programy provázány s klíčem registrů, jakým může být následující:

[14] Benny/29A se nechal inspirovat biologií, přesněji Darwinovou teorií o prežití pouze toho nejsilnějšího či nejlepšího organismu a proto nazýval jednotlivé části GENY.

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\exefile\shell\open\command

Konkrétně tento klíč umožňuje definovat program, který bude vykonán automaticky při spuštění libovolného EXE souboru, přičemž původně žádaný EXE mu bude předán jako parametr. Standardně obsahuje klíč následující hodnotu: "%1" %* ,ale po případné infekci virem může být změněna do formátu: infikovany_soubor.exe "%1" %*

Problém nastává ve chvíli, kdy je infikovany_soubor.exe odmazán, ale hodnota ponechána v infikovaném tvaru. V takovém případě není možné spustit žádný EXE soubor, jelikož se operační systém dožaduje onoho smazaného infikovany_soubor.exe. Dočasným východiskem je přejmenování přípony EXE na COM a to především u souboru REGEDIT.EXE (-> REGEDIT.COM), který umožní odstranění tohoto problému.

4.3.6 Likvidace antivirových programů

Některé viry šířící se elektronickou poštou mohou využít svého rychlého masivního rozšíření a případné antivirové programy zlikvidovat ještě v době, kdy nejsou schopny takto útočící virus detekovat[15]. Antivirus pak není schopen provozu a ani do budoucna se jeho situace nezmění. Některé viry se pokoušely vymazat antivirový program kompletně z disku, jiné pak vhodně upravit virovou databázi tak, aby nebyl antivirový program do budoucna schopen žádný virus detekovat.

V době operačního systému Windows 9x/NT se uplatňuje zcela odlišná metoda. Virus si může využitím vhodných API funkcí (FindWindowA atd.) vytvořit seznam všech aktivních oken (i momentálně skrytých) a hledáním řetězců v titulcích (titles) oken se snažit najít eventuálního nepřítele v podobě rezidentního antivirového programu či osobního firewallu. Takové z jeho pohledu nevhodné okno může deaktivovat.

Jinou metodou je rozbor běžících procesů a jejich následné vypínaní. Velice úspěšný virus Win32/BugBear.B vyhledává hned několik desítek běžících procesů a následně je deaktivuje.

Skupinu těchto virů můžeme nazvat retro-viry, tj. odvetnými viry.

4.3.7 Falšování skutečného odesílatele (spoofing)

Většina dnešních virů šířících se elektronickou poštou využívá e-mail spoofingu, tedy falšování adresy skutečného odesílatele. Na místo adresy skutečného odesílatele bývá dosazena zcela odlišná adresa, ať už fixní (virus Win32/Sobig.B dosazoval adresu support@microsoft.com) či proměnlivá (prohlídkou knihy adres apod.). Nepříjemným efektem mohou být rozhořčené dopisy uživatelů, rozesílané na nesprávnou adresu odesílatele. Více viditelným problémem jsou ovšem automatické odpovědi některých antivirových systémů na poštovních serverech. Ty se snaží e-mailovou odpovědí upozornit dotyčnou osobu na skutečnost, že na hlídaný server zaslala virus. Bohužel tím, že je adresa skutečného odesílatele podvržena, dostane toto upozornění zcela nevinný uživatel. Pokud jde zároveň o nezkušeného uživatele, může celá situace dojít tak daleko, že přestane věřit vlastnímu antivirovému systému a celý svůj operační systém kompletně přeinstaluje jen na základně těchto informačních zpráv.

[15] V opačném případě by k odstranění antiviru nemohlo dojít. On-access skener by infekci odhalil ještě před spuštěním infikovaného souboru.

Skutečného odesílatele je možno vypátrat jen na základě údajů v hlavičce e-mailu, a to konkrétně u položky „Received:". Záleží zcela na poštovním serveru, jak podrobné informace do této části e-mailu zapíše, proto můžeme zjistit nic nebo všechno.

4.3.8 Šíření se po síťově sdílených discích

Šíření po síťově sdílených discích v rámci operačního systému Microsoft Windows bývá vedlejší činností některých virů (popřípadě hlavní v případě úspěšného viru Win32/Opaserv). Celá idea šíření po síťově sdílených discích je postavena na skutečnosti, že v řadě případů je pro čtení i zápis nasdílen celý pevný disk počínaje hlavním adresářem (root). Pro virus není problémem se ze vzdáleného počítače na takto nevhodně nasdílený disk překopírovat nebo rovnou některé soubory infikovat (jako souborový virus) a svoji automatickou aktivaci zajistit prostřednictvím upraveného souboru WIN.INI a sekce RUN=[16].

5 Makroviry

Makroviry jsou relativně mladou skupinou virů. První z nich se objevil v roce 1995 a jmenoval se WM/Concept.A. Jak naznačuje samotný název skupiny, jsou tvořeny makry. Již první kancelářské aplikace umožňovaly práci s takzvanými makry. Tehdy se ve většině případů jednalo o zapamatování a uložení posloupnosti kláves (klávesnicová makra). Tato makra mohla být opakovaně vyvolána a mohla zjednodušit a zautomatizovat práci aplikačního programu. Postupem se ale způsob práce s makry měnil, takže dnes jsou makra vytvářena ve speciálních vyšších programovacích jazycích, jejichž možnosti se vůbec neliší od klasických programovacích jazyků. Mohou tedy manipulovat s daty aplikace a také s makry, které jsou s danými daty spojeny (například je kopírovat). Bohužel mohou i modifikovat ostatní data na daném počítači.

Makro (nebo souhrn maker), které je schopno zkopírovat sebe sama z jednoho dokumentu do druhého (a to opakovaně), je nazýváno makrovirem. Je zřejmé, že úspěšné šíření viru vyžaduje několik podmínek. Používaná aplikace musí být široce používána a musí docházet k výměně dat včetně maker mezi jednotlivými uživateli a počítači. Všechny tyto podmínky dnes splňují hlavně programy Microsoft Word a Microsoft Excel, a proto jsou zdaleka nejrozšířenější právě makroviry pro tyto dva programy (součástí kancelářského balíku Microsoft Office).

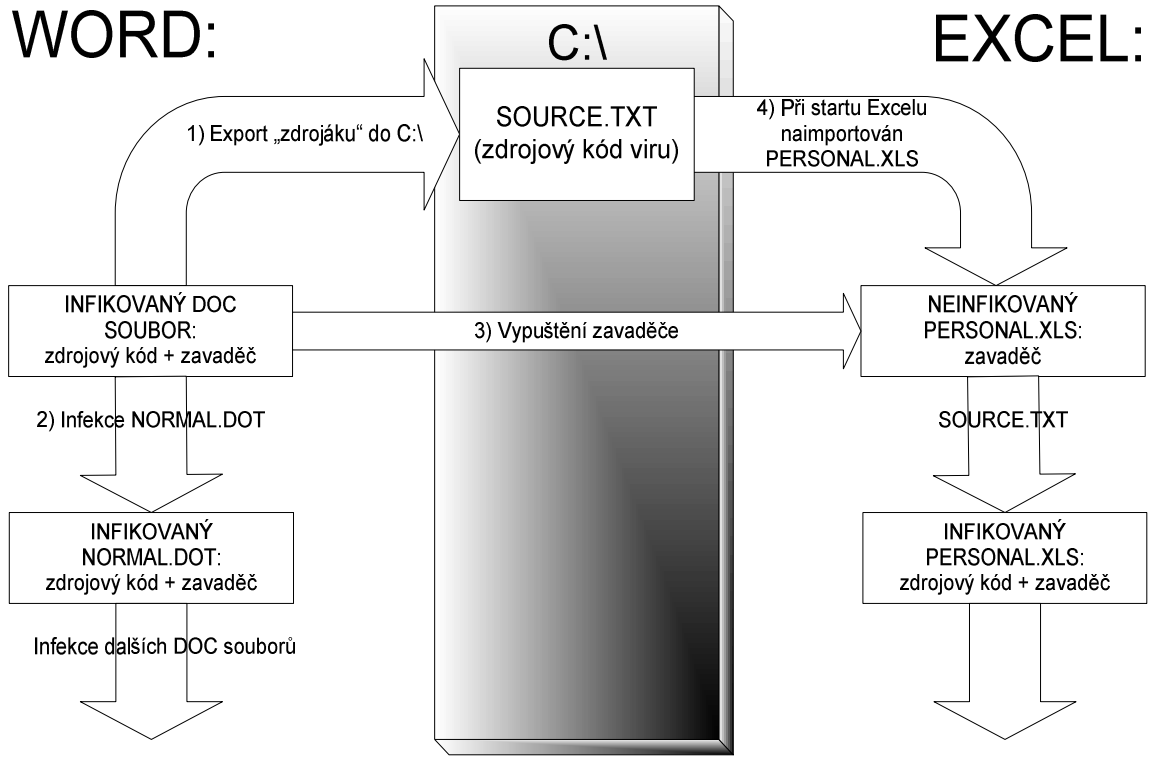
Programy z MS Office neukládají makra, spjatá s dokumentem, do zvláštního souboru, ale do stejného, ve kterém jsou uložena vlastní data. V takovém případě se tedy nejedná o čistě datové soubory, ale svým způsobem o programy. To zásadně mění přístup k takovým souborům z hlediska bezpečnosti.

Pro ukládání údajů do souboru používá Microsoft svůj vlastní formát, nazývaný Compound Storage File, který je součástí OLE2 (Object Linking and Embedding). Takový soubor obsahuje svůj vlastní souborový systém, včetně tabulek FAT, adresářů, podadresářů, souborů. Struktura souboru (a způsob uložení maker) nebyla firmou Microsoft nikdy zveřejněna, je definováno pouze API rozhraní, které však v řadě případů nechtějí či nemohou antivirové programy využívat, a to někdy z principiálních důvodů (například pro programy pracující pod MS-DOSem není API k dispozici). Způsob uložení maker se navíc mezi jednotlivými verzemi značně liší.

Zajímavou vlastností makrovirů je to, že mohou fungovat jak na různých počítačích (IBM PC, Macintosh) a tedy i operačních systémech (Windows 3.1, Windows

[16] Z principu operačního systému Windows plyne, že k aktivaci viru zavádějícího se prostřednictvím souboru WIN.INI dojde až po následujícím restartu Windows.

9x, Windows NT, MacOS)[17], tak i na různých aplikacích v rámci kancelářského balíku (nejčastěji MS Office). Příkladem může být makrovirus O97M/Tristate, který je schopen napadat dokumenty všech tří základních aplikací MS Office (Word, Excel, Powerpoint) – tzv. cross-infektor. Pokud hovoříme o fungování pod různými operačními systémy, pak lze do této kategorie zařadit teoreticky všechny makroviry využívajících kancelářského balíku MS Office. Důvod je prostý, MS Office neexistuje pouze pro MS Windows, ale i pro MacOS a makroviry lze považovat za multi-platformní.



Obrázek 6 Princip fungování cross-infektorů, kdy je exportován zdrojový kód makroviru do speciálního souboru (v uvedeném případě SOURCE.TXT) a následně pak importován do dalších aplikací balíku Microsoft Office.

Protože největší reálnou hrozbu představují makroviry aplikace MS Word, budou se následující části věnovat nejvíce právě ji[18].

Jaký je tedy princip činnosti makroviru? Pro Word funguje většina makrovirů následujícím způsobem: makrovirus je uložen v napadeném dokumentu. Pokud je takový dokument otevřen a načten danou aplikací, může být za určitých podmínek virus spuštěn (např. pomocí automaker, ale i jinými způsoby, o nichž se zmíníme za chvíli). Virus pak může zkopírovat sebe sama do nějakého globálního dokumentu, který mu zajistí aktivaci při každém následujícím spuštění aplikačního programu (u Word je to soubor NORMAL.DOT). Při dalších spuštěních aplikace tak zůstává aktivní a může napadat všechny vytvářené, modifikované či jen čtené dokumenty. Toto samozřejmě není jediná cesta, jak se mohou makroviry šířit, je však rozhodně nejpoužívanější.

[17] Jisté potíže mohou nastat při využití externích objektů mimo aplikaci Microsoft Office. Příkladem může být využití jistého externího programu, který je součástí platformy Windows, nikoliv MacOS.

[18] Kromě makrovirů pro Word a Excel jsou k vidění i pro PowerPoint, Access, AmiPro, Corel, Autocad, ale nepředstavují žádnou hrozbu v porovnání s prvními dvěma.

5.1 Dokumenty & Šablony

Word rozlišuje dva typy dokumentů, se kterými pracuje. Dokumenty a šablony. Počínaje Wordem 97 (8.0) došlo k radikálním změnám ve struktuře dokumentů a implementaci makrojazyka VBA 5 (Visual Basic for Applications). Do doby Wordu 95 (7.0) tak platilo následující.

Šablony mohly kromě klasických dat obsahovat různé další informace, jako definice tlačítek lišty, klávesové zkratky, či právě makra. Rozlišení mezi oběma typy nebylo dáno rozšířením ve jméně souborů (.DOC pro dokumenty a .DOT pro šablony), i když tato rozšíření Word standardně používá. Vlastní rozlišení bylo v jediném bitu ve stavovém slově uvnitř dokumentu. Typ dokument nemohl uživatel změnit pomocí prostředků Word, výjimkou byl příkaz makrojazyka, který byl makroviry používán. Soubory, označené jako dokumenty, mohly obsahovat makrovirus, který byl ale v tom případě v neaktivním stavu. Dokumenty, které makrovirus napadl, byly vždy změněny na šablony, aby virus mohl být při jejich načtení aktivován.

5.2 Automakra

Vlastností Wordu i dalších aplikací v rámci Office firmy Microsoft, která zcela jistě přispívá k šíření makrovirů, jsou automakra. Jedná se o makra se speciálními jmény, která jsou za určitých podmínek automaticky vykonána. Příkladem mohou být makra AutoExec (automaticky spuštěno při startu programu), AutoOpen (při otevření dokumentu), AutoClose (při uzavření dokumentu) a AutoExit (při skončení aplikace). Naneštěstí provedení těchto maker nejde jednoduše a spolehlivě zakázat, navíc mají přednost makra, uložená v právě otevíraném dokumentu před globálními makry (např. z NORMAL.DOT).

5.3 Změny v menu

To však zdaleka není jediný způsob, jakým může být makrovirus aktivován. Ve skutečnosti je celé prostředí dnešních aplikací velmi flexibilní - pomocí maker je možno předefinovat libovolnou položku menu či libovolné tlačítko na stavové liště. Makrovirus pak může být aktivován například tím, že uživatel uloží nebo vytiskne soubor. Tuto vlastnost řada virů využívá i pro skrývání své přítomnosti, když manipuluje s položkou Nástroje / Makro. Existují viry, které takovou položku prostě zakáží, změní ji na prázdné okno, simulují chybu či dokonce zobrazí všechna makra, samozřejmě kromě svých. Uživatel může jen těžko zjistit prostředky dané aplikace přítomnost viru.

5.4 Předdefinované klávesy

Další věcí, kterou jsou schopna makra modifikovat, jsou klávesové zkratky a vůbec stisk libovolné klávesy. Virus tak může být aktivován, kdykoli například stisknete klávesu „a", „e" nebo mezerník (virus WM/Gangsterz.A).

5.5 Šifrovaná makra

Word umožňuje makra, uložená v dokumentu, zašifrovat. Makra, zašifrovaná programem Word, jsou nazývána execute-only a jsou určena pouze pro spouštění, nelze je tedy žádným způsobem číst nebo modifikovat. Toho využívají distributoři nejrůznějších komerčních maker (chrání svoje intelektuální vlastnictví), správci velkých podniků (mají jistotu, že jim uživatelé makra nemění), ale bohužel také makroviry. Šifrovaná makra totiž nemohou být zobrazena či dokonce modifikována pomocí položky Nástroje / Makro. Samotné šifrování je v případě starších verzí Wordu (do Office 95) natolik triviální, že není nejmenší problém makra dešifrovat a případný virus nalézt.

5.6 Vlivy na šíření

5.6.1 Setkání více makrovirů

Na rozdíl od klasických virů nemají makroviry jednoznačný začátek, v řadě případů se skládají z více maker, a tak mají více vstupních bodů a cest, kterými mohou být aktivovány. Pokud se na jednom počítači setkají dva různé makroviry, jejichž sada maker se překrývá (jedná se o jména maker), může dojít ke vzniku nového druhu viru, který je kombinací dvou předchozích. V některých případech výsledný „produkt" není funkční, tj. není schopen se dále množit, ale někdy dojde k tomu, že nový makrovirus se bez problémů šíří dál. Dokonce ani nemusí jít o setkání virů, některé makroviry jsou totiž schopny „spolykat" i obyčejná neškodná makra uživatele či dokonce antivirová makra, která se někdy používají.

5.6.2 Vlastní degenerace

Některé makroviry jsou náchylné k tomu, že za určitých podmínek jsou schopny některá svoje makra „poztrácet" a přesto zůstávají funkční. Většinou se jedná o chybu v kódu makra, který někdy zapomene určité části viru zkopírovat . Typickým představitelem této skupiny maker je virus WM/Rapi.A, který má původně sedm maker, ale jejich počet se může snížit až na pět, přičemž virus je stále schopen se šířit.

5.6.3 Chyby v produktu

Některé starší verze programu Word obsahovaly chybu, jež vedla za určitých podmínek k tomu, že při zápisu souboru došlo k poškození makra. Mechanismus tohoto poškození nebyl příliš jasný, ale tu a tam došlo k tomu, že makro bylo zapsáno špatně. Změněno bylo většinou jen jedna či několik bajtů a samozřejmě záleželo na tom, kam se zrovna Word „trefil". V makrech byl totiž uložen zdrojový text, který mohl obsahovat řadu věcí, které s funkcí maker přímo nesouvisely (komentáře, řetězce znaků a podobně). Navíc i při poškození některé výkonné funkce mohl makrovirus celkově zůstat funkční. V praxi to vedlo k tomu, že se objevila celá řada různých variant téhož viru, které vznikly „samovolně", tj. bez přímého zásahu člověka. A je jasné, že výše uvedené platí zejména pro ty viry, které byly více rozšířeny mezi uživateli, a tedy častěji kopírovány.

5.6.4 Rozpad makroviru

Uvedli jsme si několik způsobů, kterým mohou být viry poškozeny. V některých případech virus může fungovat i nadále, jindy ale přestane být virem - ztrácí možnost šířit se do dalších dokumentů. Uvedený případ často nastává i po „odborném" ale pouze částečném odstranění makrovirů. Nepříjemný může být i případ, kdy manipulační činnost zůstane funkční, takže může dojít k jejímu vyvolání i u jinak nefunkčního viru.

5.6.5 Rozdílné verze aplikací

Jak je ve světě programového vybavení zvykem, neexistuje pouze jediná verze dané aplikace. Asi to pro tak rozšířené programy ani jinak být nemůže. Brzy po uvedení na trh se totiž objeví nejrůznější lokalizace, servisní aktualizace (service pack) a jen o málo později nová verze, přičemž se celý cyklus opakuje. Tento cyklus ovlivňuje i makroviry. Všimněme si nejdříve lokalizací. Největší rozdíly jsou patrné do verze Office 95. Například program Word se vyskytuje v nejrůznějších jazykových verzích. Zajímavé je, že způsob lokalizací se v jednotlivých zemích velmi odlišuje. Srovnáme-li německou a českou verzi Word verze 6, uvidíme, že v německé verze byly přejmenovány funkce, spjaté s položkami menu (DateiSpeichern místo FileSave), v české verzi se nic takového

nevyskytuje. Naopak u nás byla lokalizována i jména datových vnitřních struktur, k čemuž naopak nedošlo v Německu. Asijské mutace programu Word mají data uložena ve formátu Unicode, který používá pro kódování jednoho znaku šestnáct bitů, západní svět ukládá každý znak do osmi bitů, takže tyto programy nejsou vzájemně kompatibilní.

Z uvedeného plyne, že ne všechny starší makroviry (konkrétně napsány pro verzi Office 95 a starší) se mohou šířit ve všech verzích programu Word verze 6. Některé se šíří pouze v dané jazykové mutaci, jiné mohou pracovat v několika různých programech. Ve jménu makroviru se takové případy rozlišují uvedením kódu příslušné země (např. německý WM/Xenixos.A:De či tchajwanský WM/Twno.A:Tw). Díky české lokalizaci se České republice a částečně i Slovensku vyhnula první velká vlna šíření makrovirů, které se nebyly schopny šířit v českém Wordu a objevovaly se pouze sporadicky a zejména u firem, které byly v kontaktu se zahraničními partnery. Vše se změnilo až v létě 1997, kdy se objevil WM/CAP.A, který byl schopen šířit se snad ve všech západních verzích programu bez ohledu na způsob lokalizace.

Na jaře 1997 se objevila nová verze kancelářských programů firmy Microsoft pod názvem Office 97. Ta přinesla z hlediska virů řadu novinek. Zcela se změnil programovací jazyk, způsob ukládání maker do dokumentu, ochrana maker a podobně. Obecně lze říci, že vše je v nové verzi mnohem složitější. Podstatným rozdílem je i to, že prostředí maker nebylo lokalizováno vůbec, a tak se úplně stírají rozdíly mezi jednotlivými národními verzemi programu Word 97. Změny v následujících verzích nejsou již tak rapidní.

5.6.6 Konverze

Word 97 umí automaticky zkonvertovat makra, vytvořená předchozí verzí programu. To se samozřejmě týká i makrovirů, které mohou být převedeny do nového programovacího jazyka, a přitom zůstat plně funkční. Neplatí to pro všechny makroviry, hodně záleží na tom, jaké funkce používají - čím jednodušší virus, tím větší šance na jeho konverzi (je to podobné jako s klasickými viry v DOSovém okně Windows 95 či Windows NT). Makra, která jsou šifrována, zásadně konvertována nejsou. Některé viry sice zkonvertovány jsou, ale ztratí schopnost se množit. Příkladem může být známý virus WM/CAP.A. Z hlediska binárního vyjádření, vlastní činnosti, způsobu detekce i odstraňování jsou zkonvertované makroviry úplně jinými druhy virů, než jejich předchůdci. Přitom vznikají zcela automaticky. Microsoft sice do finální verze zařadil velice primitivní mechanismus na zjišťování tehdy nejrozšířenějších druhů virů, ale ten pouze nepatrně snížil nebezpečí konverze. Navíc se objevil až ve finální verzi, takže některé makroviry (např. WM/Wazzu.A) byly úspěšně konvertovány beta verzí programu (v případě makroviru W97M/Wazzu.A dokonce uvnitř Microsoftu!). Zveřejněný Service Pack pro Office 97 výrazně omezil nejrozšířenější metodu šíření tehdy známých makrovirů. Přesto se rychle objevila cesta, jak toto opatření obejít. Tento způsob se poprvé objevil ve virech WM97/Class a od té doby je používán i v řadě dalších makrovirů.

6 Skriptové viry

Pokud vynecháme primitivní pokusy škodlivých skriptů v podobě dávkových souborů (.BAT), které mohou překvapit pouze občasným primitivním kódováním, zůstávají reálnou hrozbou pouze, VBS a JS skripty.[19]

Možnosti dávkových souborů BAT nebyly natolik široké, aby uspokojili některé uživatele operačního systému Windows a tak příchod Internet Exploreru 5.0 znamenal i příchod modulu Windows Script Host.

[19] Hrozbou mohou být i soubory s příponou SHS. V tomto případě jde o jakousi OLE „obálku“, která VBS či JS skript může obsahovat.

Modul Windows Script Host (nachází se v souboru WSCRIPT.EXE) umožňuje vykonávat VBS skripty (Virus Basic Scripts) a Java skripty (JScript) a to podobně jednoduše jako v případě klasických spustitelných souborů, tj. pouhým poklepáním myši na příslušnou ikonu apod.

Bohužel s příchodem Windows Script Host nastal shodný problém jako v případě makrovirů. Skriptovací jazyk (Visual Basic) nabízí takové možnosti, že není problémem vytvořit úspěšně se šířící virus, nejčastěji prostřednictvím elektronické pošty.

Navíc můžou být výše uvedené skripty vloženy přímo do HTML kódu stránek a při nevhodném nastavení Internetového prohlížeče může dojít k jejich samovolné aktivaci a tím i aktivaci viru. Bohužel „nevhodné" nastavení je v řadě případů zároveň i standardním nastavením (především Internet Explorer). Příkladem může být následující skript, který ve spojení s Internet Explorerem dokáže neuvěřitelné věci[20].

```
<HTML>
<BODY>
<SCRIPT language=VBScript>
<!--
MsgBox "Click here to recieve a free cup holder",64,"Your free Cup holder"
Set oWMP = CreateObject("WMPlayer.OCX.7" )
Set colCDROMs = oWMP.cdromCollection
if colCDROMs.Count >= 1 then
For i = 0 to colCDROMs.Count - 1
colCDROMs.Item(i).Eject
Next ' cdrom
End If
-->
</SCRIPT>
</BODY>
</HTML>
```

Samostatnou kapitolou je ActiveX controls, kde opět rozhoduje nastavení Internet Exploreru a dále označení ActiveX controls „safe for scripting" („bezpečný pro provedení"). V minulosti totiž byla objevena bezpečnostní chyba v podobě nesprávného označení dvou ActiveX controls jako „safe of scripting": Scriptlet.typlib a Eyedog i když mohly být zneužity pro šíření virů, což se potvrdilo vznikem viru VBS/Bubbleboy.

```
CreateObject("Scriptlet.Typelib")
```

Výše uvedený řádek (a několik dalších) mu umožnilo vytváření souborů na PC oběti bez jeho svolení. Bezpečností záplata se objevila v bulletinu Microsoftu č. MS99-032.

7 Historie – viry pro DOS

Viry pro operační systém MS-DOS a jeho klony jsou nejstarší skupinou viru. Za naprosto první se považuje virus dvou bratrů z Pákistánu – boot virus Brain z roku 1986. Údajně ho úmyslně distribuovali na disketách s nelegálním softwarem, které si tam v jejich obchodě kupovali cizinci. I když šlo o zcela první virus, byl velice kvalitně napsán a úspěšně se šířil. V roce 1987 se objevily další, tentokrát souborové viry jako Lehigh, Cascade (padající písmenka na obrazovce) a první z obrovské rodiny virů Jarusalem.

Pojďme se na jednotlivé skupiny podívat podrobněji.

[20] Bez upozornění dojde k vyjetí dvířek všech CD/DVD mechanik instalovaných v PC.

7.1 Boot viry

Tato první nejstarší skupina virů infikuje části nacházející se v určitých systémových oblastech disku. Těmito oblastmi mohou být: boot sektory disket a MBR (Master Boot Record) pevného disku. Napadením nějaké z těchto oblastí si boot virus zajistí svoje spuštění hned po startu počítače.

Uvedené viry se chovají tak, že obvykle přepíší svým vlastním kódem boot sektor, a původní přepsanou část boot sektoru uschovají na jiné místo disku. Příkladem mohou být sektory:

- **v nevyužitých klastrech**
- **v použitých klastrech (hrozí poškození původního obsahu)**
- **v systémových oblastech**
- **které se nacházejí mimo aktivní oblast disku**

Taková virová infekce se potom šíří pomocí boot sektorů disket, které přišly do styku s nakaženým systémem. Operační systém DOS je pro ně totiž velmi výhodným hostitelem vzhledem k malé možnosti jeho kontroly a vzhledem k vysoké frekvenci používání těch nejjednodušších povelů jako je zápis a čtení z disku, kopírování disket, prohledávání obsahu adresáře atd.

Bootový virus infikuje systém (tj. instaluje se do paměti a zapíše svoje tělo do MBR pevného disku při zavádění systému z infikované diskety. Virus se obvykle nainstaluje do paměti jako paměťově rezidentní, a jakmile dojde k novému zavádění systému, začne infikovat boot sektor disket, které nejsou ochráněny proti zápisu, a při tom přijdou do styku se systémem (například při kopírování na disketu apod.).

V obou případech, jak u infekcí MBR nebo infekcí boot sektoru diskety, tento typ viru obvykle uloží původní boot sektor nebo tabulku rozdělení disku někam jinam na disketu či disk, ačkoliv to není vždy pravidlem. Jestliže virus zapíše původní boot sektor do kritické oblasti disku či diskety, jako je např. sektor obsahující část tabulky FAT, nebo hlavní diskový adresář, může dojít k tomu, že data na disku jsou navždy ztracena.

Charakteristickým symptomem přítomnosti infekce bootovým virem, který lze zaznamenat i u neznámých boot virů je skutečnost, že kapacita systémové paměti, oceněná programem chkdsk či mem, je většinou nejméně o 1024 bytů menší, než jaká je ve skutečnosti paměť instalovaná v systému (tj. není 640 KB, ale třeba jen 639 KB).

7.1.1 Typické schéma činnosti jednoduchého boot viru

- **Prvním krokem, kterým začíná kariéra takového viru v počítači je "nabootování" (tedy zavedení operačního systému) z diskety nakažené boot virem. V tomto okamžiku je vir zaveden do paměti a je mu předáno řízení. Není přitom nezbytné, aby disketa byla systémová; typickým zdrojem takovéto nákazy bývá disketa pro přenos dat, která zůstala omylem zamčena v mechanice při startu počítače.**

- **V další fázi si vir vyhlédne dostatečný velký kus paměti, který obsadí a překopíruje se do něj.**

- **Aby se kopie viru vytvořená v předchozím kroku dostala ještě ke slovu, musí vir následně změnit adresy některých systémových služeb (typicky diskové operace, někdy také časovač a jiné) tak, aby jejich vyvoláním došlo nejprve k aktivaci rezidentní části viru, která pak rozhodne, co se opravdu provede. Viry, které se takto usazují v paměti, nazýváme obecně rezidentní. Protože však každý boot vir je rezidentní už z principu, bývá zvykem tuto vlastnost již explicitně neuvádět.**

- **Následuje kontrola, zda je nakažen boot sektor pevného disku, ze kterého se běžně zavádí systém. Pokud je tento sektor "viru-prostý", virus jej okamžitě nakazí.**

- **Virus vyhledá na napadené disketě původní boot sektor, který načte do paměti, a spustí jej. Od této chvíle počítač pokračuje v běžném zavádění systému s tím, že virus je nadále aktivní v paměti připraven změnit běh věcí příštích.**

- **Pokud je po zavedení systému provedena jakákoli operace přistupující na disketu, je vir před provedením této operace aktivován (bod 3) a v případě, že tato disketa ještě není napadena, infikuje ji postupem analogickým bodu 4.**

- **Pokud jsou splněny zadané podmínky, provede vir připravenou akci (výpis textu, formátování disku apod.).**

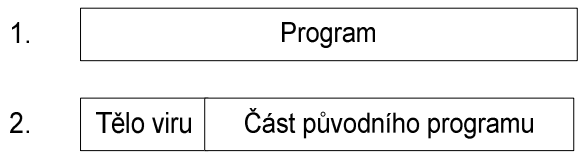
7.2 Souborové viry

Druhou a svého času nejrozšířenější skupinou jsou viry souborové. Jak již z názvu vyplývá, jejich hlavním hostitelem jsou soubory.

Tyto viry bychom mohli dále třídit podle toho, jaké soubory při infekci napadají - v zásadě jsou to vždy proveditelné soubory, neboť cílem viru je, aby provedením hostitelského kódu došlo k rozmnožení virového kódu. Nejčastěji se tedy jedná o soubory spustitelné binární (COM, EXE...). Může se též jednat o souborové viry infikující batové soubory (BAT), ovladače (SYS). Mechanismus činnosti souborových virů je však podobný ve všech případech. Podle metody infekce můžeme souborové viry rozdělit na několik skupin:

7.2.1 Přepisující viry (overwriting viruses)

Nejednoduší forma infekce, při níž dojde ke zničení (přepsání) části původního kódu programu tělem viru. Původní část je tak nenávratně ztracena a spuštěním souboru dojde pouze k aktivaci samotného viru snažící se o další replikaci a nikoliv původního programu. Ve všech případech lze tyto viry označit za viry přímé akce, jelikož nejsou paměťově rezidentní.



Obrázek 7 Přepisující virus (overwriting virus)

7.2.2 Parazitické viry (parasitic viruses)

Jako parazitické viry jsou označovány ty, které se regulérně připojí k proveditelnému hostiteli (souboru) bez toho, aby ho nějak trvale poškodily. Obecně můžeme tuto skupinu rozdělit podle toho, kam je tělo viru z pohledu původního souboru umístěno:

- **prepend (před)**
- **insert (v)**
- **append (za)**

Posledně jmenovaný způsob je nejčastější. Při infekci je původní soubor upraven tak, aby po jeho následné aktivaci došlo jak k aktivaci viru, tak posléze i původního programu.

7.2.3 Doprovodné viry (companion viruses)

Doprovodné viry jsou typem infekce, která měněny soubory, ani systémové oblasti disku. Doprovodné viry „infikují" soubory s příponou EXE tak, že vytvoří nový soubor obsahující samotné tělo viru a to se stejným názvem, ale s příponou COM (příklad: máme soubor SPUST.EXE, virus vytvoří SPUST.COM). Podle dosových priorit pak ale při volání daného souboru bez uvedení přípony dojde nejprve k aktivaci toho s příponou COM a tím i k aktivaci viru. Záleží pouze na viru, zda po vykonání vlastních činností uvolní prostor i původně žádanému programu (souboru s příponou EXE).

Alternativní metodou může být přejmenování původního EXE souboru změnou přípony a vložení se do nového souboru s původním názvem (například přejmenováním SPUST.EXE na SPUST.EXD a posléze vytvořením infikovaného souboru SPUST.EXE).

Alternativou může být tzv. i „Path companion", kdy se při infekci využívá priorit ve spojitosti s proměnnou DOS PATH (využíváno v autoexec.bat). Pokud se na disku nachází více souborů se stejným názvem (a klidně i příponou), je při volání určitého souboru spuštěn ten, který odpovídá požadavku a je nalezen v adresářích proměnné DOS PATH jako první. Proměnná DOS PATH je tedy jakýmsi seznamem adresářů, který je procházen od začátku do konce. Pokud je virus umístěn v určitém adresáři v horní části seznamu a stejně pojmenovaný soubor, který je žádán v dolní polovině seznamu, pochopitelně dojde k aktivaci toho špatného – infikovaného.

7.3 Metody infekce COM a EXE souborů

Pod operačním systémem MS-DOS můžeme najít dva základní formáty spustitelných souborů. Jeden z nich má obvykle příponu COM, druhý EXE[21].

7.3.1 Soubor formátu COM

Nejednodušší formát spustitelného souboru. Jeho maximální velikost je omezena délkou paměťového segmentu, tj. 64 KB. Nemá žádnou hlavičku.

7.3.2 Soubor formátu EXE

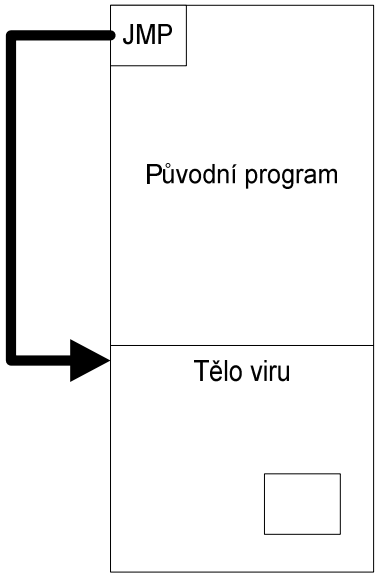
Vnitřní struktura formátu EXE je o něco komplikovanější. Skládá se ze dvou částí, hlavičky (header) a vlastního kódu programu. Struktura hlavičky je následující:

Offset:	Popis:
00h	Signatura souboru (znaky „MZ")
02h	Velikost poslední stránky v B
04h	Délka souboru v 512 B stránkách
06h	Počet položek v relokační tabulce
08h	Velikost hlavičky v 16 B paragrafech
0ah	Minimální potřebná paměť v paragrafech

[21] Opět je nutno rozlišovat rozdíl mezi uvedenou příponou a skutečnou vnitřní strukturou souboru – formátem.

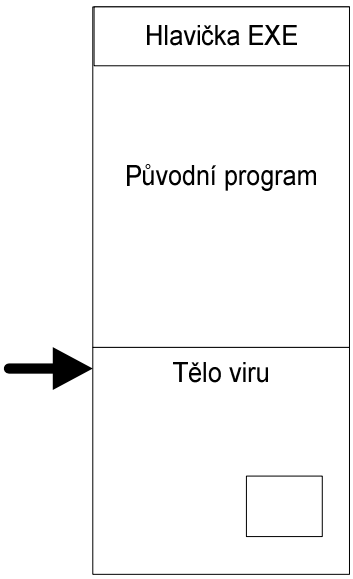
0ch	Maximální potřebná paměť v paragrafech
0eh	Počáteční hodnota SS
10h	Počáteční hodnota SP
12h	Kontrolní součet
14h	Počáteční hodnota IP
16h	Počáteční hodnota CS
18h	Offset první relokační položky
1ah	Úroveň překrývání

7.3.3 Parazitické metody infekce



Obrázek 8 Schéma převzetí kontroly COM infektoru. Instrukce skoku JMP na samotném začátku předá řízení viru

Nejzajímavější je bezpochyby infekce způsobená nedestruktivním virem z pohledu metody infekce spustitelných souborů. Nejednodušší je infekce souboru formátu COM, jelikož nemá žádnou hlavičku. Virus si tak pouze zapamatuje první tři bajty původního souboru (tj. uloží je na určité místo svého těla) a na jejich místo vloží instrukci skoku – JMP, směřující na počátek viru (hovoříme-li o skupině append). Pokud je infikovaný soubor spuštěn, dojde k aktivaci viru a následně i k rekonstrukci souboru do původního stavu právě díky uloženým třem bajtům. Zajistí se tak i spuštění původního programového kódu, který byl hlavní náplní souboru před infekcí. Mluvíme o tzv. předání řízení hostitelskému programu.

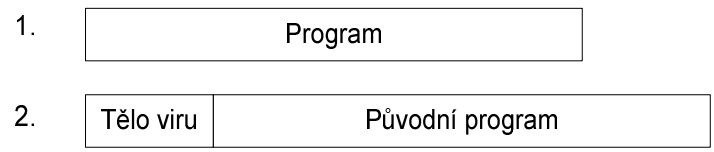


Obrázek 9 Schéma převzetí kontroly EXE infektoru. Hodnota entry pointu (CS:IP) v hlavičce ukazuje na kód viru

V případě EXE souboru je situace komplikovanější. Během infekce musí dojít ke změně některých hodnot hlavičky (původní jsou opět uloženy na určité místo těla viru – nejčastěji CS, IP, SS, SP nebo rovnou celá hlavička) a ostatní musí být dopočítány. Žádná instrukce skoku se nevkládá, na začátek kódu viru ukazuje přímo tzv. entry-point (vstupní bod), tedy hodnoty CS:IP hlavičky.

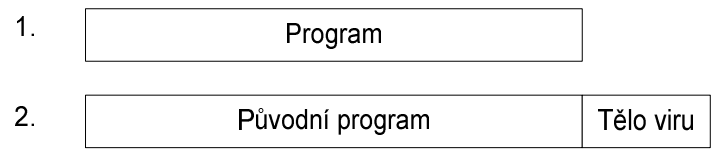
Hrádky s hlavičkou a s původním obsahem souboru lze různě kombinovat, proto mohou vzniknout následující metody infekce[22].

- **Prepend – Virus se nachází před původním programem. Původní program může být odsunut celý vzad, popřípadě přesunuta pouze úvodní část, místo které je umístěno tělo viru.**



Obrázek 10 Metoda infekce - prepend

- **Append – nejpoužívanější metoda. Obecně ji odpovídá i schéma při infekci COM a EXE souborů.**

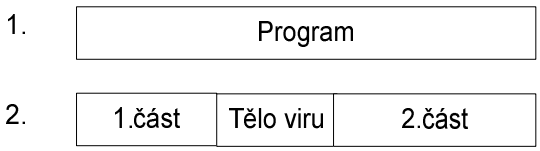


Obrázek 11 Metoda infekce - append

- **Insert – málo využívaná metoda. Při využití této metody se nabízejí více variant, jak vynaložit s hodnotou ukazující na začátek původního programu (entry-point). Virus ji nemusí měnit, pokud se vloží přesně na místo, kde původní program**

[22] Jde pouze o znázornění vzájemného umístění těla viru a původního programu.

začínal. V opačném případě ano. Jistou alternativou jsou tzv. „mezerové“ viry (cavity viruses), které jsou ale spíše záležitostí Win32 virů. Při této metodě dochází k přepsání „prázdných“ míst souboru a tak nedojde ke změně velikosti infikovaného souboru při porovnání s původní a ani k trvalému poškození.



Obrázek 12 Metoda infekce - insert

7.3.4 Schéma činnosti souborových virů

7.3.4.1 Viry přímé akce – přepisující

- **Virus je aktivován v okamžiku, kdy je spuštěn libovolný infikovaný program, a jeho první starostí je vyhledat jiný vhodný program, který je možno napadnout.**
- **Virus zkopíruje své tělo na začátek vyhlédnutého programu, čímž si zajistí, že bude aktivován okamžitě po jeho spuštění, ale zároveň je tímto krokem cílový program nenávratně zničen.**
- **Pokud jsou splněny zadané podmínky, virus vykoná své poslání.**
- **Virus nemůže pokračovat v činnosti původního programu, proto jej bezprostředně ukončí, často s nějakým hlášením naoko vysvětlujícím nemožnost běhu programu (například fiktivní nedostatek paměti).**

7.3.4.2 Viry přímé akce – parazitické

- **Tento krok je naprosto shodný jako v předchozím případě.**
- **Vir připojí vlastní kód k programu tak, aby zachoval všechny informace potřebné k jeho správné činnosti. Nejjednodušším způsobem jak to zařídit, je připojit tělo viru za konec napadeného programu. Aby však byl vir aktivován okamžitě po spuštění infikovaného programu, musí na začátek programu vložit instrukci skoku na tělo viru (případ COM souboru), přičemž původní kód, který je přepsán skokovou instrukcí, je uložen do těla viru.**

7.3.4.3 Viry paměťově rezidentní – parazitické

- **Při spuštění napadeného programu virus zjistí, není-li již rezidentní v paměti. Pokud nalezne svou kopii v paměti, pokračuje bodem 5.**
- **V opačném případě vyhledá a vyhradí pro své použití (tzv. alokuje) úsek paměti vhodné velikosti.**
- **Překopíruje sám sebe do tohoto úseku paměti a změní adresy některých diskových (a případně i jiných) služeb tak, aby směrovaly do těla.**
- **Obnoví začátek napadeného programu (pouze v paměti, ne na disku) a tento program spustí.**

- **Pokud je od této chvíle spouštěn jakýkoli program, je nejdříve vyvolán rezidentní virus, který tento program infikuje tím, že připojí svůj kód ke kódu programu a změní úvodní skok (stejně jako u parazitických virů přímé akce), popřípadě hlavičku (případ EXE souboru).**

- **Jsou-li splněny podmínky, provede akci.**

7.4 Paměťová ne/rezidentnost

7.4.1 Základní dělení

7.4.1.1 Nerezidentní viry - viry přímé akce

Tyto viry nevyužívají paměť pro své šíření. Stačí jim, když jsou aktivovány společně s hostitelským programem. Pak přebírají řízení jako první, provedou svoji činnost, nejčastěji replikaci a předají řízení zpět hostitelskému programu. Replikací zde většinou rozumíme například napadení všech vhodných souborů (postupně nebo naráz) v aktuálním adresáři, či napadení souborů uvedených v proměnné DOS PATH.

7.4.1.2 Paměťově rezidentní viry

Paměťově rezidentní virus setrvává ilegálně v paměti. Takový virus se většinou při prvním spuštění infikovaného souboru (pokud se jedná o souborový virus) nebo při prvním zavedení systému z infikovaného boot sektoru (pokud se jedná o boot virus) stane rezidentním v paměti, a odtud potom provádí svoji škodlivou činnost. Virus zůstává v paměti dokud není systém vypnut. Naprostá většina virů pro operační systém MS-DOS se umisťuje až na vrchol systémové paměti, ale pod mez 640 KB. Existují však i viry, které využívají nízkou systémovou paměť, paměť videokarty atd. pro ukrytí kusu svého kódu. Se zapnutím systému jsou okamžitě schopny infekce souborů či boot sektoru nebo tabulky rozdělení. Boot viry jsou už z principu paměťově rezidentní.

Paměťově rezidentním virům se věnují i následující kapitoly.

7.4.2 Přerušení (interrupt)

Přerušení je nástroj, pomocí kterého se dostane virus k prostředkům BIOSu a DOSu. Bez těchto prostředků by se virus neuměl množit ani škodit.

Přerušením (Interrupt) nazveme hardwarový signál, který vede k přerušení běhu programu vykonávaného procesorem a přesměrování běhu procesoru na program obsluhující přerušení. Po ukončení tohoto programu následuje obnovení původního stavu a pokračování přerušeného programu.

Přerušení můžeme rozdělit na přerušení vyvolané hardwarem (nemaskovatelné signálem NMI, maskovatelné signálem INTR) a softwarem (instrukce INT, chyba běhu programu). Softwarová přerušení jsou na úrovni BIOSu, DOSu a uživatelských programů.

Viry pracují s přerušeními vyvolanými instrukcí INT (je jich 256, tj. 0-255). Při tomto přerušení následuje skok na místo, kde je obslužný program. Adresa pro obslužný program se nachází v paměti RAM, kde je tzv. tabulka přerušovacích vektorů (adres obslužných programů).

Tabulka začíná od adresy 0000:0000 a má velikost 4*256 bytů. Na 1 vektor (adresu) potřebujeme 4 byty (2 na segment a 2 na offset).

7.4.2.1 Přesměrování přerušení (hook interrupt)

Z předchozího plyne, že přerušení jsou uložené v tabulce, která je v paměti RAM. Pokud změníme některý vektor (adresu) přerušení v tabulce stane se toto: Po vykonání příslušného přerušení bude následovat odskok do jiné oblasti paměti. Takto se můžeme postarat o to, aby se na místo původního obslužného programu vykonal náš program a po skončení našeho programu následuje skok na původní obslužný program (hook interrupt).

Tohoto hojně využívají viry (ale i jiné rezidentní programy) v MS-DOSu. Virus je uložený v paměti a některé z přerušení má přesměrované tak, že ukazuje do těla viru. Jestliže nějaká činnost vyvolá přerušení (např. prohlížení, čtení, zavádění programu), dochází k aktivaci viru a ten může infikovat program (prohlížený, čtený, zaváděný) nebo způsobit škodu.

7.4.3 Paměťově rezidentní viry (memory resident)

Umístění viru rezidentně v paměti probíhá ve dvou krocích. Prvním je vyhledání nebo vytvoření vhodného místa, kam by se vir umístil. Takové místo musí být jednak dostatečně velké, jednak dostatečně bezpečné.

V případě boot viru není výběr možností nijak převratný. Virus se totiž instaluje do paměti v okamžiku, kdy ještě není zaveden operační systém a nemá tedy k dispozici ani to málo funkcí pro manipulaci s pamětí, které DOS nabízí.

Naprosto nejčastějším způsobem, kterým se boot viry s tímto problémem vypořádávají, je snížení velikosti základní paměti a využití takto uvolněného místa. Datová oblast BIOSu obsahuje proměnnou, která říká, kolik základní paměti je k dispozici. Ve většině případů obsahuje tato proměnná hodnotu 640, což je maximální velikost základní paměti v kilobajtech. Pokud virus tuto hodnotu sníží, operační systém se domnívá, že má k dispozici méně paměti, a ani se nepokouší využívat tu část, která by tam podle něj být neměla, a virus se tedy může do této oblasti bezpečně nastěhovat.

Typická část kódu, která zajišťuje zmíněné uvolnění paměti vypadá následovně:

```
MOV AX, [413h]    ; Precti velikost volne pameti
DEC AX            ; Zmensit o velikost potrebnou pro virus
DEC AX            ; (zde o 2 KB)
MOV [413h], AX    ; Uloz zpet na sve misto
MOV CL, 06
SHL AX, CL
MOV ES, AX        ; Segment zacatku uvolnene pameti
```

K nevýhodám uvedené metody patří také poměrně snadná možnost zjistit netypickou velikost základní paměti (některá zařízení využívají tuto oblast obdobným způsobem jako viry, je tedy třeba zvážit, jestli se jedná o běžný stav nebo ke změně došlo bez nějakého zjevného důvodu).

Jinou metodou, kterou používají některé, zejména menší viry, je využití malých volných oblastí v datových oblastech v dobré víře, že do nich nebude nikdo jiný zapisovat. S využitím této techniky se lze občas setkat i v těle některých souborových virů. Dlužno podotknout, že podobný způsob zacházení s pamětí je operačním systémem chápán jako zdvořilá prosba o jeho havárii.

Důmyslnější boot viry využívají techniku jakéhosi "meziskladu", kterým řeší dočasnou nedostupnost služeb pro manipulaci s pamětí. Její princip spočívá v tom, že se vir umístí do oblasti paměti, o které předpokládá, že nebude po určitou, poměrně

krátkou, dobu změněna (typicky se k těmto účelům využívá horní část video-ram, která není v běžném textovém režimu využívána). Poté nechá zavést operační systém a teprve potom se překopíruje na definitivní pozici s využitím všech možností, které mají souborové viry.

Souborové viry mohou využívat buď standardní kolekci tří služeb pro práci s pamětí (alokuj blok, uvolni blok, změň velikost bloku), nebo mohou použít několik různých, většinou nedokumentovaných zásahů do řídících struktur DOSu. To má pro ně naproti využití systémových služeb tu výhodu, že takto „zrezidentněný“ virus nebude figurovat na případném seznamu rezidentních programů.

Modernější viry se také naučily manipulovat nejen se základní pamětí, ale dokáží se usadit i v paměti nad hranicí 640 KB, pokud je dostupná. Typickým příkladem může být virus Tremor.

Některé z pokročilejších virů mohou dokonce sledovat hospodaření jiných programů s pamětí a v případě, když usoudí, že se pro ně uvolnilo vhodnější místo, mohou se přesunout ze svého současného působiště do nového.

Aby mělo usídlení kopie viru v paměti nějaký smysl, musí být vir nějak spojen se systémem. Toto spojení bývá realizováno přesměrováním vhodných systémových služeb do těla viru. Důsledkem takovéto vazby na systém je, že v průběhu řádné činnosti systému jsou ve vybraných okamžicích aktivovány určité části viru.

Přesměrování služeb se provádí změnou tzv. vektorů přerušení, což jsou adresy, na které se předává řízení v případě generování přerušení (systémové služby jsou realizovány právě voláním přerušení). Tabulka adres těchto služeb je uložena na konstantním místě v operační paměti (na jejím úplném začátku) a v operačním systému DOS je bohužel nekontrolovatelně přístupná všem programům, které z ní mohou hodnoty nejen beztrestně číst, ale také do ní zapisovat.

Standardní postup převzetí systémové služby virem spočívá v naplnění vektoru přerušení adresou směřující do těla viru. Při vyvolání přerušení vir provede vlastní činnost a poté, pomocí zapamatované původní hodnoty vektoru přerušení zavolá původní funkci (možný je i opačný postup, kdy vir nejdříve zavolá původní systémovou službu a poté provede vlastní činnost, případně vir původní službu nevolá vůbec a její činnost zcela nahradí). Velmi zjednodušeně lze dění v systému znázornit následovně:

Před "převzetím služby" virem	Po "převzetí služby" virem
Běží uživatelský program	Běží uživatelský program
Vyvolá systémovou službu	Vyvolá systémovou službu
Operační systém provede požadovanou operaci	Vir prozkoumá požadavek a udělá, co uzná za vhodné; Případně se ke slovu dostane i OS
Uživatelský program pokračuje	Uživatelský program pokračuje

Vlastní čtení a změnu hodnot vektorů přerušení lze realizovat opět pomocí služeb DOSu. Používání těchto služeb lze ovšem snadno monitorovat, proto se mnoho virů uchyluje ke změně vektorů přerušením přímo do tabulky vektorů, které se nachází zcela na začátku operační paměti, což je jednak méně nápadné, a pokud je programátor šikovný, i kratší.

Složitější viry občas nevyužívají tuto přímou metodu převzetí vektoru, místo toho směrují službu do svého těla před jakýsi "můstek" umístěný odděleně od těla viru, jehož kód nedělá nic jiného, než že bezprostředně zavolá cílovou funkci viru.

Toto řešení není samoúčelné, ale má z pohledu virů jisté výhody. Jedná se o typický produkt souboje virů a antivirů; ty se totiž naučily sledovat, kam ukazují vektory některých důležitých služeb, a pátrat v jejich okolí po kódu viru. Uvedený postup se jim to pokouší alespoň zkomplikovat, také ruční analýza takového kódu je obtížnější.

8 Další části virů

Výše uvedené kapitoly detailně popsaly samotný proces infekce, ale k ostatním důležitým součástem se pouze přiblížily. Následuje tak výčet dalších částí běžného viru.

8.1 Vlastní identifikace & příznak napadení

Souborové viry se ve většině případů snaží infikované soubory nějakým předem definovaným způsobem označit, aby v budoucnu nedocházelo k jejích opětovné re-infekci. Hovoříme o příznaku napadení, přičemž pojmu vlastní identifikace rozumíme jako části viru, která se tento příznak snaží nalézt – identifikovat. Pokud by k této činnosti nedocházelo, virus by mohl tentýž soubor opakovaně infikovat a jediným limitem by v tu chvíli byla velikost pevného disku. Opakovaná infekce se často nazývá infekci sendvičovou.

Příznakem napadení může být například:

- **Identifikační řetězec**, kdy je na určitém místě souboru (obvykle konec) umístěn speciální řetězec. Jistým extrémem je potom záměna úvodní hodnoty „MZ“ u EXE souborů za jinou.
- **Datum & čas.** Pokud se vrátíme do historie, řada virů využívala datum a čas souborů jako příznak napadení. Buď byl nastaven na určitou hodnotu a nebo na hodnotu zcela nesmyslnou (některé varianty viru Vienna nastavovaly hodnotu času na 62). Jako komplikovanější způsob lze jmenovat nastavení data a času souboru dle určitého vzorce. Příkladem může být opět virus One_Half.3544.A, a vzorec (datum mod 1eh)=(čas and 1fh).
- **Délka souboru.** Zarovnání délky souboru na hodnotu dělitelnou určitým číslem patří mezi výjimečné případy.

8.2 Vyhledání obětí

Schopnost vyhledání vhodných obětí je velice důležitou vlastností každého viru, od kterého se očekává úspěšné rozšíření. Nejzajímavější skupinou jsou v tomto směru viry souborové – je nutné najít spustitelné soubory vhodné k infekci. Spustitelné soubory mohou být vyhledány:

- **na základě přípony**, kdy je při hledání použita maska *.EXE, *.COM apod.
- **na základě hlavičky**, kdy jsou jednotlivé soubory interně zkoumány (hledány znaky „MZ“ u EXE souborů) bez ohledu na jejich příponu. Ve výsledku tak mohou být nalezeny soubory, které ačkoli jsou vnitřní strukturou spustitelné, mají nesprávnou příponu. Díky této skutečnosti může být k vidění infikovaný soubor, který není běžně možno infikovat (například s příponou DAT, ale vnitřní strukturou EXE).

8.3 Výkonná sekce

Výkonná sekce zajišťuje vnější projevy viru. Ty jsou značně různorodé a jsou omezeny pouze nápaditostí a schopnostmi autorů. Mezi viry lze nalézt takové, které tuto sekci vůbec neobsahují a navenek se tedy nijak mimořádně neprojevují. Spektrum činností pokračuje nejrůznějšími zvukovými projevy, výpisy textů a zobrazováním grafických efektů. Mezi nejméně nápadité se řadí viry, které nějakým způsobem úmyslně narušují správný chod systému a modifikují soubory, případně postupně či naopak během zlomku sekundy zařídí, že dojde ke ztrátě dat.

Zajímavým příkladem jsou viry řady Win32/Sobig, které se po provedení aktivační podmínky přestanou dále do budoucna šířit.

8.4 Aktivační podmínky

Vykonání připravené činnosti je v drtivé většině případů vázáno na splnění tzv. aktivační podmínky (trigger conditions). Je to pochopitelné, neboť virus, který by se začal projevovat po prvním spuštění, by byl odhalen nepřijatelně brzy. Naprostým favoritem mezi aktivačními podmínkami je vazba na systémové datum nebo čas.

8.5 Ošetření chyb

Sekce ošetření chyb zahrnuje poměrně krátkou část kódu, která má význam pro důslednější utajení viru v systému. Virus by měl zajistit, aby v případě chybového stavu, který vznikl v důsledku jeho činnosti, neprozradil svou přítomnost.

9 Speciální skupiny virů

9.1 Multi-platformní (cross-platform)

Multi-platformní viry jsou zvláštní skupinou a pokud nepočítáme makroviry[23], tak i velice ojedinělou. Jde o viry, které jsou schopny šíření pod různými platformami, tj. operačními systémy různých výrobců. Nepočítáme-li makroviry, lze tuto schopnost uplatnit jen teoreticky. Otázka totiž zní „Jak přelézt na jiný operační systém ?“. Odpověď existuje, ale vyřčená situace nenastává příliš často. Odpovědí tedy je:

- **Prostřednictvím emulátorů. Například pod Linuxem existuje několik emulátorů operačního systému Microsoft Windows (win4lin, wine), které dokáží pod Linuxem spouštět Win32 aplikace a tím i případné viry.**

- **Prostřednictvím sdílených oblastí disku, kde uživatel například shromažďuje jak Win32 programy, tak i Linuxové aplikace (pro pozdější vypálení na CD apod.). Novější distribuce Linuxu obsahují nativní podporu souborových systémů FAT, FAT32, NTFS a tak není problémem k takovým oblastem disků přistupovat a společně s Win32 programy sdílet adresáře.**

Praxe je tedy taková, že multi-platformní viry jsou především lákadlem pro novináře a různé „press-release“ zprávy výrobců antivirového softwaru. Ukázkovým příkladem může být níže popsaný virus Winux českého autora – Bennyho/29A, který jako první dokázal napadnout jak PE Win32 soubory, tak i ELF soubory pod Linuxem. Ten ač

[23] V případě makrovirů lze mluvit o multi-platformních virech, jelikož kancelářský balík Microsoft Office existuje jak pro Microsoft Windows, tak i MacOS.

se reálně nešířil, vyvolal řadu diskuzí a článků, z nichž jeden byl k nalezení i na webových stránkách televizní společnosti CNN.

9.2 Multi-partitní (multipartite)

Označení „multi-partitní“ se nejčastěji využívalo ve spojitosti s viry, které kombinovaly několik metod replikace. Typicky šlo o DOS viry, které se dokázaly šířit soubory (souborové viry), ale dokázaly se usadit i v systémové oblasti pevného disku – MBR. Šlo tak zároveň částečně o boot viry.

I když by se i dnes našla celá řada multi-partitních virů (například kombinace souborového a viru šířícího se elektronickou poštou), pojem zůstal vyhrazen DOS virům dle výše uvedené specifikace.

Klasickým příkladem této skupiny může být virus One_Half.3544, tehdy velice moderně napsaný virus, způsobující některým antivirům velké problémy s detekcí (bližší informace v kapitole o polymorfismu).

10 Speciální techniky

10.1 Stealth

Pod tajemně znějícím pojmem „steath“ se skrývá celá skupina prostředků, určených k zamaskování přítomnosti viru. Princip spočívá v tom, že virus kontroluje nejrůznější aktivity počítače a v případě potřeby falšuje některé údaje nebo modifikuje některé činnosti.

Sledování aktivit je realizováno opět pomocí přesměrování služeb (vektorů přerušení, API funkcí) na virus. Podle toho, které služby je virus schopen zpracovávat a falšovat jejich výsledky, lze rozlišovat úroveň stealth techniky.

Za nejednodušší formu (semi-steath) je považováno uvádění falešné délky souborů. Přesměrováním vhodné služby operačního systému lze docílit, že soubory se budou ve výpisu (ať už příkazem DIR, v souborovém manažeru, v průzkumníku) jevit v původní délce před infekcí.

Vyspělejší viry jsou komplexnější, takže k maskování dochází i při prohlížení infikovaného souboru v nějakém editoru, kdy je zobrazen pouze původní nezávadný kód. K této efektivní činnosti může docházet při vhodném přesměrování služeb operačního systému, týkajících se otevření souboru (otevření souboru v editoru - rychlá dezinfekce) a uzavření souboru (ukončení práce se souborem - opětovná infekce).

I když byl v případě operačního systému MSDOS prvním stealth virem paradoxně hned ten nejstarší – Brain z roku 1986, v případě Win32 virů jsme se takové události hned tak nedočkali. S příchodem Win32 jako by stealth viry zcela vymřely a najít některý rozšířený Win32 virus s výše uvedenou technikou je skoro nemožné.

Kupříkladu rutina[24] starající se o zamaskování DOS viru v MBR pevného disku nebo boot sektoru diskety by mohla vypadat následovně (virus na pozici 0/0/1, původní kód na 0/0/7 – track/head/sector):

```
int_13h_entry:
      pushf
```

[24] Zdrojem **-zine #1*, článek *Stealth a handy overview* autorů *MGL/SVL*.

```
        cmp dl,80h
        js flopak           ; floppy or hard drive ?
                            ; this should hide the presence of
                            ; virus in the MBR
        push cx
        or dl,dl
        jnz OK              ; head 0 ? If so, then if

        cmp cx,1
        jnz OK              ; track 0 sector 1, check critical
                            ; functions

        cmp al,1
        ja OK               ; stealth only when 1 sector read
        cmp ah,02h          ; read
        jz zvedavec
        cmp ah,0ah          ; long read ( is not necessary )
        jz zvedavec
        cmp ah,03h          ; write
        jz write
        cmp ah,0bh          ; long write (is not necessary )
        jnz OK
write:
zvedavec: mov cl,7          ; redirect R/W to stored MBR
OK:    call emulINT13h
        pop cx              ; we call original INT 13h with "good
                            ; parameters and we return callers CX
                            ; which covers our tracks
        jmp short VRATsa

flopak:  ....               ; here 'd be handled floppy access
     ....                   ; similar to hard drive access
     ....
VRATsa:
        popf
        retf 2
emulINT13h:
        pushf
        call dword ptr cs:[original_INT13h]
        ret
```

10.2 Kódování & Polymorfismus

10.2.1 Souborové viry

Prvotním účelem kódování bylo znepřehlednit vlastní kód viru a ztížit tak jeho analýzu. Navíc je tímto postupem zkomplikováno uzdravení napadeného programu, protože původní obsah začátku souboru (hlavičky) je také zakódován. Takové kódování realizovaly některé souborové viry tak, že vlastní kód viru byl v programu uložen – zakódován nějakým jednoduchým algoritmem (oblíbené bylo provedení operace XOR s každým bajtem kódu viru) a před vlastním virem se nachází krátká dekryptovací smyčka, která zajistí jeho transformaci do původní podoby (některé viry tuto smyčku rozdělují na několik navzájem propojených částí po souboru).

Takto zakódovaný virus má téměř konstantní podobu. Téměř proto, že stejně jako běžné viry si potřebuje do svého těla zapsat několik proměnných hodnot, které mohou být v každé generaci odlišné; převážná část viru zůstává neměnná.

Novější viry začaly využívat kódování s proměnlivou hodnotou k dosažení toho, aby každý exemplář viru byl z velké části odlišný; shodná zůstává pouze dekódovací smyčka na začátku programu. To sice značně zkomplikovalo vyhledávání některým antivirovým programům používajícím charakteristické sekvence, ale neustále bylo relativně snadné určit kódovací hodnoty a program dekódovat. Navíc šlo stále použít vždy stejně vypadající dekódovací smyčku k identifikaci viru.

Nástupce této technologie představují polymorfní viry, které používají poměrně komplikovaných metod k tomu, aby i dekódovací smyčka mohla mít proměnlivou podobu. U takového viru se liší každý jednotlivý exemplář a proto je nemožné použít k jeho vyhledávání charakteristické sekvence.

Jednodušší viry používají k dosažení různorodosti metodu prokládání nevýznamnými instrukcemi a záměny pořadí, v případě DOS souborového viru například jednoduchá smyčka typu:

```
MOV BX, zacatek_viru          ; adresa prvni zakodovane instrukce
znovu:
XOR [BX], AL                  ; kodovaci funkce
INC AL                        ; zmen kodovaci konstantu
INC BX                        ; prejdi na Halsi bajt
CMP BX, konec_viru            ; test na konec viru
JNE znovu                     ; pokud neni cele telo rozkodovano,
                              opakuj
```

může být v příští generaci viru beze změny funkčnosti realizováno takto:

```
MOV BX, zacatek_viru
NOP                           ; nevyznamna instrukce
znovu:
XOR [BX], 13h
CLC                           ; zmena priznaku CY nic neovlivni
INC BX                        ; prohozene poradi techto dvou instrukci
INC AL                        ; nema zadny vliv na cinnost
CMP BX, konec_viru
NOP                           ; nevyznamna instrukce
NOP                           ; nevyznamna instrukce
JNE znovu
```

Takové změny později dokázaly všechny antivirové programy podchytit, proto autoři virů tyto metody dále zdokonalovali.

Jednou ze skutečností, která je využívána k těmto účelům je fakt, že díky ne zcela dokonalému instrukčnímu souboru použitých procesorů Intel lze stejnou instrukci zapsat různým způsobem:

```
; operacni kod          instrukce
   93                   HG AX,BX          ; instrukce vymeni obsah
                                          ; registru AX a BX

   87 C3                XCHG AX,BX        ; coz ma naprosto stejny
   87 D8                XCHG BX,AX        ; efekt jako predchozi
```

Postupem času se začaly objevovat viry, které každou svoji kopii kódují odlišným algoritmem a používají stále komplikovanější programové konstrukce uvnitř dekódovacích smyček.

Příchod operačního systému Windows 9x/NT s sebou přinesl další prostor pro rozvoj těchto virů. Některé obsahují dekryptovací smyčku hned v několika vrstvách, jiné pak aplikují „brute force attack“ náhodně generovanými klíči k dekryptování (Win32/Crypto českého autora Prizzy). K tomuto musí dojít, jelikož virus úmyslně „ztratí“ klíč potřebný k dekryptování těla viru. Při spuštění infikovaného souboru je tento klíč hledán metodou „pokus/omyl“ (tedy „brute force attack“), takže než dojde k vykonání samotného těla viru, může uplynout i polovina sekundy (záleží na rychlosti CPU). Úmyslem je stížení prostupnosti kódu případným emulátorem.

Nástupci jsou metamorfní viry.

10.2.2 Makroviry

Podobný vývoj potkal i makroviry. K jisté úrovni polymorfismu může makrovirus dojít několika způsoby. Příkladem může být náhodné prohazování řádků kódu, které jsou ve správném pořadí volány skoky (goto).

```
GoTo 10
70 NI = NormalTemplate.VBProject.VBComponents(1).CodeModule.Lines(2, 1):
GoTo 80
110 If NormInstalled = True And ActInstalled = True Then Exit Sub Else:
GoTo 120
140 With Carrier: VirCode = .Lines(1, .CountOfLines): End With: GoTo 150
10 Application.EnableCancelKey = wdCancelDisabled: GoTo 20
```

Jiným příkladem může být prokládání kódu poznámkami (rem).

```
Sub AutoOpen()
'126072574489154071950412607257448915407195041260725744891540719504126072574
44891540719504
Randomize
'936133573694442222593613357369444222259361335736944422225936133573694442222
25
x = 0: o = 0
'105831801124206508396161058318011242065083961610583180112420650839616105831
801124206508396161058318011242065083961610583180112420650839616
On Error GoTo 93
```

V neposlední řadě pak proměnlivost názvů proměnných, procedur a funkcí.

```
If j68657670 = vbReadOnly And System.OperatingSystem = "Windows" And
System.LanguageDesignation = "English(United States)" Then Call
XRoach(f9517$)
If j68657670 = vbReadOnly + vbArchive And System.OperatingSystem =
"Windows" And System.LanguageDesignation = "English(United States)" Then
Call XRoach(f9517$)
If j68657670 = vbReadOnly Then GoTo dRoach
If j68657670 = vbReadOnly + vbArchive Then GoTo dRoach
For I = 1 To ActiveDocument.VBProject.VBComponents.Count
If ActiveDocument.VBProject.VBComponents(I).Name = "Roach" Then
tcd7670485512 = True
```

10.3 Metamorfismus

Narozdíl od polymorfních dochází u metamorfních virů ke změnám v samotném zdrojovém kódu viru (v případě polymorfních virů bylo tělo konstantní, i když navenek pokaždé jinak zakryptováno) i když ke změnám jeho působnosti nedochází.

Pravděpodobně prvním a zároveň úspěšným virem tohoto druhu byl slovenský TMC:Level_6x9.A (TMC – Tiny Mutation Engine) autora Ender. V infikovaném souboru se nenachází tělo viru v klasické binární podobě, ale jen kompilátor, společně se zdrojovým pseudo-kódem viru. Při spuštění infikovaného souboru kompilátor sestaví do paměti novou, vzhledově zcela odlišnou kopii viru. Kompilátor neobsahuje žádné podezřivé instrukce a není ho proto možno běžně detekovat heuristicky. Během kompilace se mezi jednotlivé instrukce mohou, ale nemusí vkládat instrukce skoku. Délka skoku není předem známa, proto se musí vyhradit pro každý skok více místa, než je potřeba. Důsledkem je poměrně častý výskyt třech instrukcí NOP následujících po sobě.

Trochu odlišně přistupoval k situaci již 32-bitový virus Win32/Apparition. Ten kompilátor nenesl v infikovaném souboru s sebou, nýbrž si ho drze dovolil hledat na uživatelově disku a rovněž ho využít.

Virus Win32/Regswap modifikoval vlastní kód výměnou používaných registrů. V následující ukázce je to viditelné.

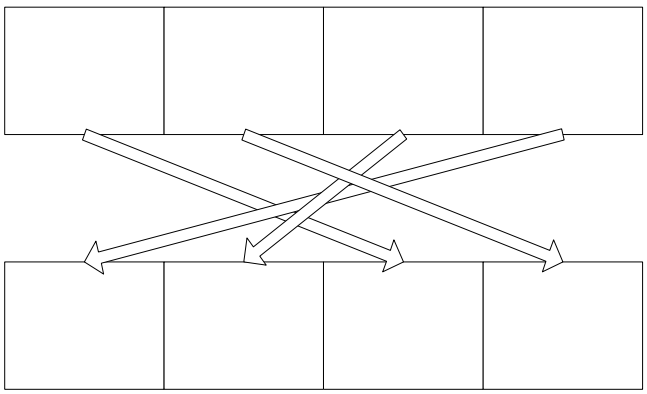
Ukázky z článku „Hunting for metamorphic“ od Péter Ször a Peter Ferrie – Symantec Corp.

```
5A              pop edx
BF04000000      mov edi,0004h
8BF5            mov esi,ebp
B80C000000      mov eax,000Ch
81C288000000    add edx,0088h
8B1A            mov ebx,[edx]
899C8618110000  mov [esi+eax*4+00001118],ebx

58              pop eax
BB04000000      mov ebx,0004h
8BD5            mov edx,ebp
BF0C000000      mov edi,000Ch
81C088000000    add eax,0088h
8B30            mov esi,[eax]
89B4BA18110000  mov [edx+edi*4+00001118],esi
```

Jak je z ukázky patrné, je stále možno využít detekci na bázi sekvencí.

Některé další Win32 viry se pokoušely o přehazování vlastních podprogramů (procedur a funkcí), čímž mohlo vzniknout teoreticky až n! kombinací, jestliže n značí množství takových podprogramů.



Obrázek 13 Ilustrační schéma přehození podprogramů

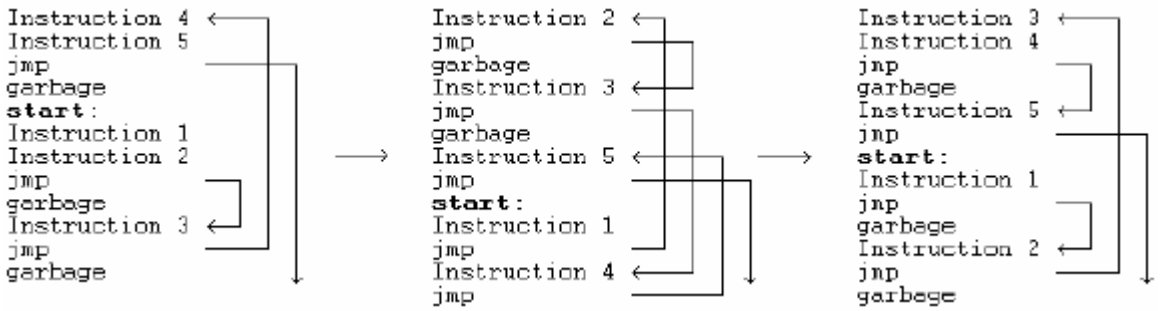
V následující ukázce (virus Win32/Evol) již sekvence nepomůžou. Jednotlivé generace vyjadřující jedno a to samé jsou odděleny prázdným řádkem.

```
C7060F000055      mov dword ptr [esi],5500000Fh
C746048BEC5151    mov dword ptr [esi+0004],5151EC8Bh

BF0F000055        mov edi,5500000Fh
893E              mov [esi],edi
5F                pop edi
52                push edx
B640              mov dh,40
BA8BEC5151        mov edx,5151EC8Bh
53                push ebx
8BDA              mov ebx,edx
895E04            mov [esi+0004],ebx

BB0F000055        mov ebx,5500000Fh
891E              mov [esi],ebx
5B                pop ebx
51                push ecx
B9CB00C05F        mov ecx,5FC000CBh
81C1C0EB91F1      add ecx,F191EBC0h ; ecx=5151EC8Bh
894E04            mov [esi+0004],ecx
```

Později se objevil virus Win95/Zperm, který navíc podobný kód prokládal „smetím“ (garbage), tedy nepotřebnými instrukcemi a dále skoky JMP, takže ani posloupnost nezůstala v každé generaci jednotná.



Obrázek 14 Tři různé generace viru Win95/Zperm

Navíc společně s tímto virem byl do oběhu vypuštěn Real Permutating Engine (RPME), na základně něhož se v budoucnu objevilo několik podobných virů.

10.3.1 Virus Win95/Zmist

Samotnou kapitolu si určitě zaslouží virus Win95/Zmist, který byl na konferenci Virus Bulletin 2000 prezentován jako nejvíce „undetectable virus“ a celkově je odborníky považován za to „nejlepší“, co se v celé historii počítačových virů objevilo.

Z0mbieho (autor viru ze skupiny 29A) Mistfall engine obsažený ve viru Win95/Zmist totiž dokáže dekompilovat jakýkoliv PE soubor a vlastní kód viru proložit do původního programu ! Jednotlivé instrukce viru jsou provázány skoky – instrukcemi JMP, takže nemusí být vizuálně za sebou (podobně jako Win32/Zperm). Z výše uvedeného je zřejmé, že dochází k obrovským změnám v původním kódu (i když významově představuje stále totéž) a následně musí dojít i k úpravě souvisejících záležitostí (data references…), aby byl nově vytvořený (zkompilovaný) PE soubor korektní.

Nově vytvořený PE vypadá jistě příšerně, ale společně s využitím techniky EPO a polymorfní dekryptovací smyčky rozdrobené na několik částí navzájem propojených instrukcemi skoku se stal pro některé antiviry opravdu nedetekovatelným[25].

10.4 Obrana proti antivirům

Častým jevem, se kterým se lze v těle viru setkat, je používaní technik, které ztěžují trasování a analýzu kódu.

10.4.1 Obrana proti krokování kódu

Těchto postupů je celá řada a není možné (ani účelné) pokoušet se o jejich úplný výčet.

Proti trasování kódu debuggerem se viry brání například vkládáním časově závislého kódu. To znamená, že program předpokládá, že se nějaká část kódu provede v určitém, velmi krátkém časovém intervalu. Pokud je program trasován ručně, instrukci po instrukci, je tato doba pochopitelně nesrovnatelně delší a vykonávání programu se pak může ubírat zcela odlišnou větví.

Jiným příkladem je kód, v průběhu jehož provádění nelze zapisovat do zásobníku (což je činnost, bez které se debugger za běžných okolností neobejde).

Použití instrukcí koprocesoru byla jednou z dočasných metod, jelikož před několika lety je emulátory antivirových skenerů nedokázaly zpracovat.

Použití MMX (multimedia extension) instrukcí byla další z dočasných metod, podobně jako v předchozím případě.

Koprocesor a instrukční sada MMX nemusí být nutnou součástí každého PC (i když v dnešní době jde o téměř 100% jisté záležitosti) a odtud plyne, že viry využívající těchto instrukcí nebudou schopny provozu na procesorech nižších než Intel 80486DX (v případě instrukcí koprocesoru) nebo nižších než Intel Pentium MMX (v případě instrukcí MMX). Jelikož jsou výše uvedené instrukce využity v dekódovacích smyčkách polymorfních virů, mohou být k vidění hned dvě takové smyčky. Jedna s využitím těchto instrukcí, druhá pak bez nich. Na základě instrukce CPUID je pak rozhodnuto, jaká z nich bude aplikována.

10.4.2 Tunelování

Technika tunelování spočívá ve schopnosti vyhledávat původní adresy systémových, nejčastěji diskových služeb a ty pak používat při práci s disky ve snaze obejít případný paměťově rezidentní antivirový software. Jde především o záležitost DOS virů.

10.4.3 Retroviry

Takto jsou označovány viry, které bojují s antiviry napřímo. Tedy jejich cílenou likvidací. Bližší informace o této problematice jsou v kapitole o virech šířících se elektronickou poštou.

[25] V některých případech muselo dojít k hlubším zásahům do skenovacího jádra antivirového programu, v jiných se prostě detekce tohoto viru „opomenula“.

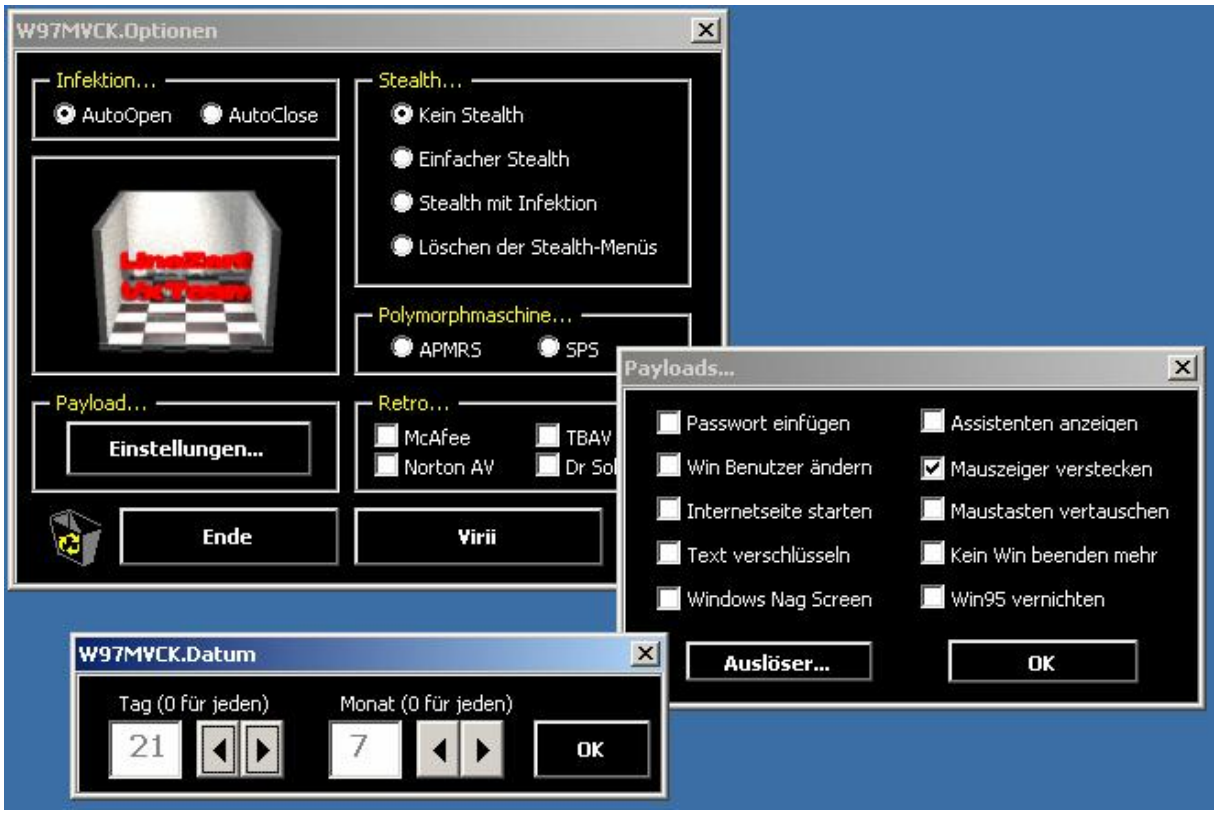
10.5 Operační systém závislý na viru

Nejznámějším příkladem se bezpochyby stal virus One_Half.3544.A, který během své přítomnosti na hostitelském počítači postupně šifroval uživatelská data na pevném disku. Pokud operační systém potřeboval zapsat nějaká data na disk, virus převzal kontrolu a tato data nejdříve svým algoritmem zašifroval a až poté je nechal zapsat na disk. Při požadavku ke čtení je naopak dešifroval a předal dál operačnímu systému. Při neodborném odstranění takového viru z počítače mohlo dojít i ke ztrátě klíče, podle něhož byla data šifrována, popřípadě dešifrována do čitelné podoby. Uživatel tak přišel o data a ve většině případů i o možnost úspěšného startu PC. Operační systém a tedy i PC se tak stalo závislým na samotném viru.

Podobným příkladem pod operačními systémy MS Windows může být virus Win32/Crypto (opět od českého autora, tentokrát Prizzyho), který použitím API funkcí LoadLibrary a FreeLibrary sleduje právě používané DLL knihovny. Při volání LoadLibrary danou DLL knihovnu dešifruje, naopak při konci jejího užití (FreeLibrary) ji opět zašifruje dle určitého klíče. Neodborným léčením může opět dojít ke ztrátě takových DLL knihoven a díky tomu i k omezení funkčnosti aplikací, které danou knihovnu využívaly.

11 Generátory virů

Na Internetu lze najít řadu generátorů virů (nebo konstruktorů), tedy programů, které na základě požadavků uživatele dokážou vygenerovat hotový virus.

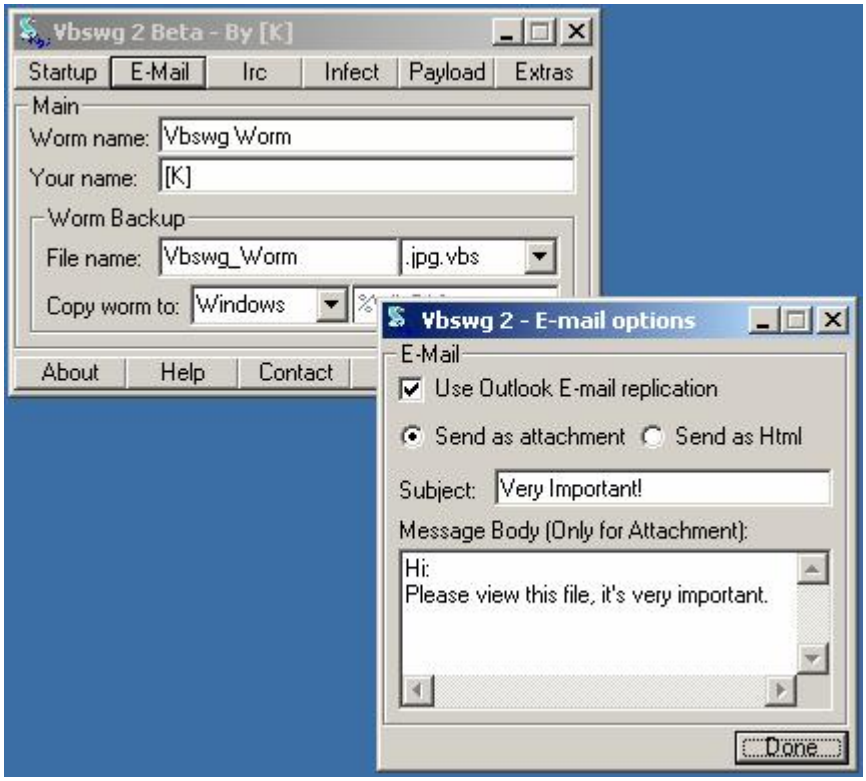


Obrázek 15 Generátor W97MVCK pro tvorbu makrovirů

Vytvořit virus tak může být hračkou i pro běžného uživatele. Stačí jen zatrhnout několik položek, vyplnit pár řádků a stisknout tlačítko, které virus dle požadavků vygeneruje.

Jedním z nejstarších byl generátor VCL (z roku 1990) pro generování jednoduchých COM infektorů. Největší raritou pak byl pravděpodobně generátor V.G.O.L, jelikož byl jako jediný v podobě webových stránek ! Autorem byla osoba „MadDeamon“ a tuto službu nabízel na serveru histeria.sk. V dnešní době se většina generátorů zabývá skriptovými viry a makroviry.

Samotný výstupní kód produkovaný generátorem musí být průběžně modifikován. Pro antivirovou společnost totiž není problémem zajistit 100% detekci všech virů, které různou kombinací přepínačů můžeme vygenerovat (často bývá název takového viru doplněn dovětkem .based).



Obrázek 16 V generátoru VBSWG vznikl i veleúspěšný virus VBS/SST.A - Anna Kurnikova !

12 Služby

12.1 PC Viruses In-the-Wild (www.wildlist.org)

PC Viruses In-the-Wild je seznamem nejvíce hlášených virů z celého světa. Každou lokalitu (stát, republiku...) zastupuje nejčastěji osobnost antivirové společnosti (pro ČR Pavel Baudiš ze společnosti Alwil software, na Slovensku Miroslav Trnka - Eset), která pravidelně dodává obraz o lokální virové situaci. Takto získané údaje jsou centrálně zpracovány a každý měsíc je vydáván onen zmiňovaný seznam.

Část takového seznamu může vypadat například následovně:

```
W32/Acebot.............[Newbiero.......] 6/02 AoMoSgSoTm
W32/Aliz.A-mm..........[...............]11/01 AlAmAoAsSmSoTa
W32/Aplore.A-mm........[Aphex..........] 6/02 AoAsZbZz
W32/Apost.A-mm.........[...............]10/01 AoAsTaZz
W32/BadTrans.A-mm......[13312.........] 5/01 AlAoAsFpGrJdKdLsOzPhSfSgSr
                                               Zz
W32/BadTrans.B-mm......[29020.........]11/01 AlAmAoAsDpEiEwFpJdJmKdMoMs
```

```
                                                    OzPbPhSaSfSjSmSrStZvZy
W32/Benjamin.A-mm.......[..............] 6/02 AoMsSoStTaTm
W32/Bibrog.C-mm.........[..............] 4/03 AcShStTm
W32/BleBla.B-mm.........[Verona.B.......]12/00 AoAsAyFpJmSjSoZz
W32/Braid.A-mm..........[..............]11/02 AlAmAsAyFpGrJdJkJmJpKdMhMs
                                                    MtOzPbPhSgSjSkSmSoStTcTmZb
                                                    ZvZyZz
W32/BugBear.A-mm........[..............]10/02 AlAmAoAsAyDpEiEwFpGrJdJkJm
                                                    JpJwKdMhMoMsMtOzPbPhRvRzSa
                                                    SfSgShSjSkSmSoSrSsStTaTcTm
                                                    XcZvZyZz
W32/Cervivec.A..........[..............] 4/02 AoAsMtPb
W32/Chir.A-mm...........[..............]10/02 SjStTa
W32/Choke.A.............[..............] 7/01 AoEiOz
```

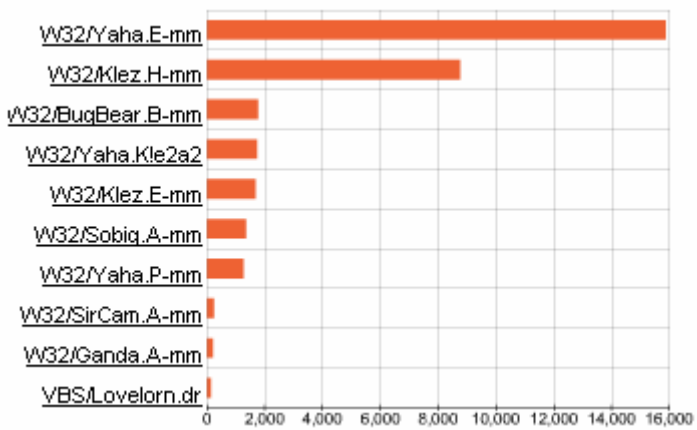
V levé části je uveden název viru, uprostřed pak jeho alternativní název. Následuje datum prvního přidání viru do hlavního seznamu PC Viruses In-the-Wild. Zkratky vpravo představují iniciály osobností (nejčastěji), které daný virus ohlásili jako šířící se ve vlastní lokalitě. Přítomnost dvojice písmen „Pb“ (Pavel Baudiš) tak signalizuje aktivní přítomnost viru na území ČR, podobně jako „Mt“ (Miroslav Trnka) na území SR.

V hlavní části seznamu se nacházejí viry, které jsou hlášeny ze dvou nebo více lokalit. Bohužel je tento údaj zkreslen, jelikož některé lokality (například USA) jsou zastoupeny větším počtem osobností. Dalším problémem je nepravidelnost jeho vydávání. Seznam PC Viruses In-the-Wild je využíván jako jeden z podkladů pro kvalitní srovnávací testy antivirových skenerů.

12.2 MessageLabs (www.messagelabs.com)

Kompletní URL nejzajímavější oblasti je www.messagelabs.com/viruseye/threats.

„MessageLabs Intelligence“ provádí kontrolu miliónů e-mailů denně po celém světě (s největším podílem v UK a USA). K tomuto účelu jsou po celém světě rozestaveny speciální servery, přes které prochází obrovské množství elektronické pošty. Elektronická pošta je podrobena důkladné analýze. Je kontrolována několika antiviry, ale i jinými technologiemi (například na detekci spamu – nevyžádané pošty). Získané informace jsou průběžně centrálně vyhodnocovány a vznikají velice zajímavé statistiky. Jednou z nich může být následující graf.

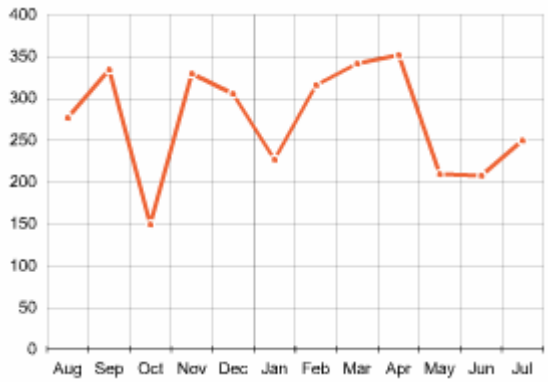


Obrázek 17 Ukázka deseti nejvíce se šířích virů elektronickou poštou za posledních 24 hodin (snímek z 20.7.2003 - 21:45)

Obrázek 18 Virus Win32/Yaha.E a informace o něm

Jelikož jsou veškeré události reportovány, lze i později zpětně zjistit, odkud například virus prvně přišel (v tomto případě Indie). Zajímavou hodnotou je výskyt v jednotlivých státech, ale i „Peak infection ratio“, což je četnost výskytu daného viru v e-mailech (ve výše uvedeném případě je virem Win32/Yaha.E infikován průměrně každý 186. e-mail !).

Podobný poměr může vypadat v celkovém měřítku pro všechny viry následovně.



Obrázek 19 Poměr infikovaných a čistých e-mailů v globálním měřítku

Opravdu lahůdkou je promítnutí číselných hodnot z jednotlivých serverů do mapy. Krásně tak lze sledovat šíření vybraného viru v prvních hodinách od objevení. Dokonce nechybí ani živý přenos, kdy jsou do mapy označovány výskyty virů, které byly nalezeny před malým momentem !

12.3 EICAR

EICAR je institutem (European Institute for Computer Antivirus Research) pořádající různé konference a ostatní akce. Pro nás je ovšem nejdůležitější existence tzv. „The Anti-Virus test file“ právě od institutu EICAR. Onen „The Anti-Virus test file“ je zcela neškodný soubor (nejčastěji EICAR.COM) o velikosti 68 bajtů, který obsahuje následující řetězec[26]:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

[26] Za řetězcem mohou následovat mezery a znaky cr, lf a CtrlZ, takže nemusí jít nutně o 68 bajtů veliký soubor.

Několik antivirových společností se dříve dohodlo, že pokud bude antivirovým skenerem nalezen právě takový, 68 bajtů veliký soubor s výše uvedeným řetězcem, antivirus ho identifikuje jako virus. Jedinou podmínkou je, že ze strany antivirového skeneru je tento soubor podporován (dnes je tomu v naprosté většině případů).

Soubor dle popisu se stal významným pomocníkem při zjišťování, zda funkčně vyhlížející antivirus je opravdu funkční a dokáže detekovat reálné viry (k tomuto nemusí docházet například při vzájemné kolizi s jiným antivirovým systémem atd.).

Co se týče řetězce, je složen tak, že ho lze utvořit i na klávesnici (snad jen třetí znak by mohl zmátnout, jedná se o „ó“, nikoliv nulu) a navíc je spustitelný (vypíše „EICAR-STANDARD-ANTIVIRUS-TEST-FILE“).

Antivirový software

Stále stoupající množství počítačových virů nás nutí brát antivirovou ochranu jako naprostou samozřejmost. V dnešní době je již poměrně vzácné najít zodpovědného uživatele, který by si hrozby neuvědomoval a ponechával počítač bez ochrany. Například ekonomické ztráty, které mohou vyplynout ze ztráty firemních dat, jsou vážnou hrozbou pro stabilitu firmy na to, aby ji bylo možné beztrestně ignorovat. Daleko jednodušší to mají domácí uživatelé, těm obvykle stačí nainstalovat některý z dostupných antivirových programů a cítit se tak v relativním bezpečí. Řada antivirových programů je navíc šířena také bezplatně a lze si je za jistých podmínek stáhnout z Internetu. Zajistit ovšem komplexní antivirovou ochranu firmy dá mnohem více práce a vyžaduje i nepoměrně více finančních prostředků. Ve firemní počítačové síti je totiž ukryto obrovské množství míst, která mohou její fungování ohrozit.

Dalším faktorem ovlivňujícím náročnost je problematičnost nasazení antivirové ochrany. Nutností je odborná dovednost a zkušenost pro skutečnou fungující ochranu. Stoupající rafinovanost škodlivých kódů vede logicky i k růstu složitosti nasazované antivirové ochrany. Zatímco pro domácího uživatele tato složitost zůstává skryta uvnitř programů, pro administrátory sítí to často představuje těžce řešitelné problémy spojené s instalací a nastavením antivirové ochrany firemní sítě. Se stále větší rozšířeností Internetu a počítačových sítí obecně, stoupá i pole působnosti škodlivých kódů. Příkladem může být e-mailový červ o kterém by i ten největší odborník před několika lety řekl, že nic takového nezná. Tyto relativně nové hrozby kladou na antivirovou ochranu další náročné požadavky.

1 Dělení antivirových programů

Antivirové programy můžeme rozdělit do několika skupin například následovně od nejednodušších až po nejsložitější:

1.1 Jednoúčelové antiviry

Jde o antivirové programy, které jsou zaměřeny na detekci, popřípadě i dezinfekci jednoho konkrétního viru, popřípadě menší skupiny virů. Jednoúčelové antiviry nelze rozhodně použít jako plnohodnotnou antivirovou ochranu, jde pouze o jakousi „krabičku poslední záchrany". Pokud uživatel zjistí (popřípadě ho někdo upozorní), že je jeho počítač infikován určitým virem, není nic jednoduššího, než využít právě schopností jednoúčelového antiviru. Jednoúčelové antiviry jsou obvykle k dispozici zdarma a slouží k likvidaci pouze rozšířeného viru v dané době.

1.2 On-demand skenery

I když jde zároveň o jednu ze součástí antivirového systému, je on-demand skener nabízen některými AV společnostmi zdarma, popřípadě jako shareware. Obvykle jde o jednoduché verze pro OS DOS ovládané přes příkazový řádek (prompt). Tato kategorie antivirových programů se uplatní především při dezinfekci počítačů, kdy např. operační systém MS Windows není schopen provozu.

Zajímavou alternativou jsou i Internetové on-line skenery, které někteří výrobci antivirových programů nabízejí na svých stránkách. Obvykle jde o skript, který ve spojení s internetovým browserem (Internet Explorer, Netscape Navigator atd.) dokážou plnohodnotně prohledat na výskyt virů pevný disk uživatele, bez toho, aby tento antivirus získal fyzicky natrvalo.

1.3 Antivirové systémy

V dnešní době jde o nejčastější formu antivirových programů. Antivirový systém se skládá s částí, které sledují všechny nejpodstatnější vstupní/výstupní místa, kterými by případná infiltrace mohla do počítačového systému proniknout. Mezi tyto vstupní/výstupní místa může patřit například elektronická pošta (červi šířící se poštou), www stránky (škodlivé skripty, download infikovaných souborů), média (cédéčka, diskety apod.). Nedílnou součástí dnešních antivirových systémů je aktualizace prostřednictvím Internetu. Jde o komplexní antivirové řešení v některých případech doplněno i o osobní firewall. Do této kategorie patří takové produkty, jakými jsou: avast!, AVG, Norton Antivirus, Kaspersky Antivirus, NOD32, McAfee Viruscan atd.

2 Antivirová ochrana sítí

Doba, kdy byla počítačová síť postradatelným luxusem některých firem je dávno pryč. Bez počítačové sítě by byla činnost některých dnešních společností silně ohrožena. S rozvojem a rozšířeností sítí vzrůstá i potřeba účinné antivirové ochrany. Antivirovou ochranou v tomto případě není jen antivirus samotný, ale například i správně formulovanou firemní bezpečnostní politiku, prováděcí směrnice a zodpovědného a zkušeného administrátora s náležitě (a prokazatelně) poučenými uživateli. Počítačová síť umožňuje rozšířit jak možností virů, tak i antivirových systémů.

Z pohledu ochrany sítí můžeme AV software rozdělit na:

- **antivirovou ochranu stanic,**
- **antivirovou ochranu groupware a souborových serverů,**
- **antivirovou ochranu na vstupních branách do Internetu.**

Některé AV společnosti nabízejí všechna výše uvedená řešení v jednom „balíku“, například pod názvem „corporate“ či „business“. Síťové verze AV systémů jsou obvykle licencovány dle množství stanic v síti. V některých případech pak podle počtu serverů, domén či množství poštovních schránek. Je zřejmé, že cena takových produktů není nejnižší a proto jsou v některých případech nakupovány postupně. V tomto případě je vhodné, aby byl nakupován v pořadí od nejnižších vrstev (stanice) až po specifické servery (poštovní server, souborový server apod.).

2.1 Antivirová ochrana stanic

Bývá často zajišťována antivirovým systémem. Jak už bylo řečeno, antivirový systém se skládá s částí, které sledují všechny nejpodstatnější vstupní/výstupní místa, kterými by případná infiltrace mohla do počítačového systému proniknout.

Antivirový systém se tak obvykle skládá z části:

- **vykonávající nepřetržitý dohled – antivirovou kontrolu nad daty, se kterými uživatel pracuje (tzv. on-access skener). Zajímavým doplňkem některých AV je i tzv. osobní firewal.**
- **umožňující provést antivirový test na vybrané oblasti. Test je vyvolán na základě požadavku uživatele (on-demand) a díky čemuž se tato část označuje jako on-demand skener.**
- **udržující antivirový systém v aktuální podobě. Zajišťuje stahování aktualizací antivirového systému z Internetu.**
- **vykonávající automatickou antivirovou kontrolu příchozí a odchozí elektronické pošty.**

Mezi další části, které již nejsou tak běžné může patřit například:

- **plánovač událostí (scheduler), který umožňuje ve zvoleném termínu automaticky vyvolat naplánovanou úlohu (např. antivirovou kontrolu důležitých dokumentů).**
- **kontrola integrity.**
- **karanténa (quarantine).**
- **monitorovací programy.**
- **antivirový plug-in pro aplikaci Microsoft Office.**
- **antivirový spořič obrazovky (screensaver).**
- **další.**

Absence řady uvedených částí v antivirovém systému nemusí nijak ovlivnit celkovou kvalitu antivirového systému. Nutným minimem je ovšem on-access skener (bod 1) a část udržující antivirový systém v aktuální podobě (stahování aktualizací z Internetu – bod 3). Pokud antivirový systém tyto části neobsahuje, bude bezpečnější se mu velkým obloukem vyhnout.

2.1.1 Aktualizace (update) antivirového systému

V předchozím textu byla zmíněna pouze aktualizace antivirového systému prostřednictvím Internetu. Ještě před pár lety se běžně aplikovala aktualizace prostřednictvím disket či cédéček. Ty rozesílaly antivirové firmy například jednou za měsíc. V dnešní době Internetu je nutností aktualizovat antivirový systém právě skrze tuto celosvětovou počítačovou síť. Důvodem jsou viry, popřípadě červi, které se po celém světě dokážou rozšířit sítí Internet za několik hodin. Modul, starající se o stahování aktualizací antivirového systému z Internetu je tak důležitou strategicky významný. Důležitou úlohu v tomto směru hrají i antivirové společnosti, které tyto aktualizace vydávají a umisťují je na své servery. Důležitým parametrem je především jejich rychlost, s jakou dokáží zareagovat na nově objevený virus. Tedy, jak dlouho jim trvá, než vydají aktualizaci a zajistí tak ochranu uživatelů, používající jejich antivirový systém. K úplné dokonalosti je nutné ještě zajistit, aby uživatel stáhl tuto aktualizaci co možná nejdříve od jejího vydání. Proto by měl být antivirový systém vybaven automatickou aktualizací, která v co možná nejkratších intervalech stahuje ze serveru výrobce nejaktuálnější verze jejího antivirového produktu.

Souhrnně lze prohlásit, že pro efektivní činnost aktualizace je nutné zajistit:

- **rychlou reakci ze strany AV společnosti,**
- **správné nastavení části, stahující aktualizace na straně uživatele.**

Některé antivirové společnosti se „chlubí" ve svých materiálech s vydáváním aktualizací každý den (tzv. daily updates), často obohaceny o pasáže „jen antivirus s každodenní aktualizací dokáže zajistit maximální ochranu před viry". Nezkušený uživatel se tímto nechá velice snadno ovlivnit při rozhodování o koupi budoucího antivirového systému. Praxe již několikrát dokázala, že jde jen o reklamní trik. To že antivirová společnost vydává aktualizace například jednou týdně, vůbec nemusí znamenat, že „spí na vavřínech". Pokud se totiž objeví něco významného, zareagují velice rychle všechny schopné antivirové společnosti. Zajímavostí je v tomto případě aktualizace, podmíněna speciálním e-mailem, hromadně odeslaným antivirovou společností svým klientům. Pokud takový e-mail projde skrze poštovní server chráněný předpokládaným antivirovým systémem, stane se signálem pro vykonání okamžité aktualizace.

Majitelé s pomalejším připojením do sítě Internet (prostřednictvím modemu – dial-up apod.) jistě bude zajímat rychlost, s jakou se aktualizace ze serveru stáhne do počítače uživatele. AV společnosti snaží zajistit co možná nejrychlejší proces stahování

aktualizace. Rychlost ovlivňuje nepochybně velikost aktualizace. Metodou, jak snížit velikost aktualizace je její rozdělení na dvě nezávislé části:

- **aktualizace programové části antivirového systému. Tato aktualizace odstraňuje nedostatky v programové části antiviru, popřípadě tuto část rozšiřuje o nové funkce.**
- **aktualizace virové databáze. Tato aktualizace zajišťuje detekci nových virů, popřípadě upravuje detekci těch stávajících.**

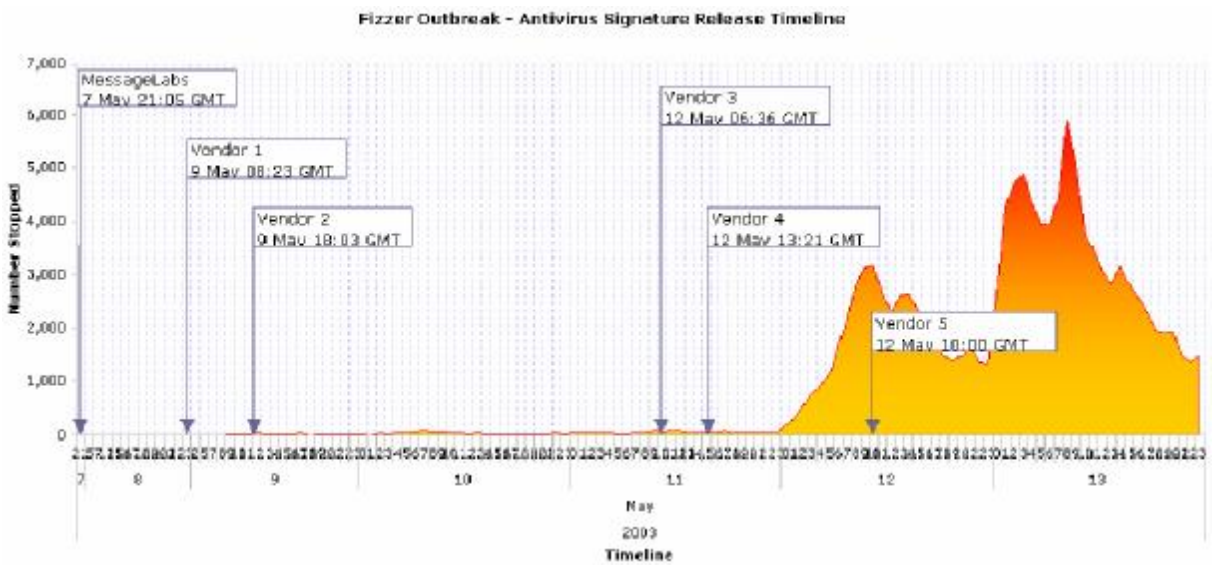
To jakým způsobem se aktualizují virové databáze je opět závislé na konkrétním antiviru. Obecně existují dva způsoby:

- **„plná" aktualizace, kdy se pokaždé stahuje celá virová databáze znovu. Logicky je tento typ aktualizace časově více náročný s přibývajícím množstvím známých virů. Velikost každé takové aktualizace lze obvykle měřit na MB. AV společnosti se tak snaží tento způsob aktualizace ze svých produktů odstranit a přejít na níže uvedený.**

- **inkrementální aktualizace, kdy se stahují pouze ty části virové databáze, které na serveru výrobce přibyly od poslední aktualizace provedené uživatelem. Výsledkem je, že jsou stahovány pouze ty informace, které se na cílové stanici dosud nevyskytují (nestahuje se vše opakovaně jako v předchozím případě). Pozitivem je zároveň i rychlost a velikost aktualizací (obvykle maximálně několik desítek KB).**

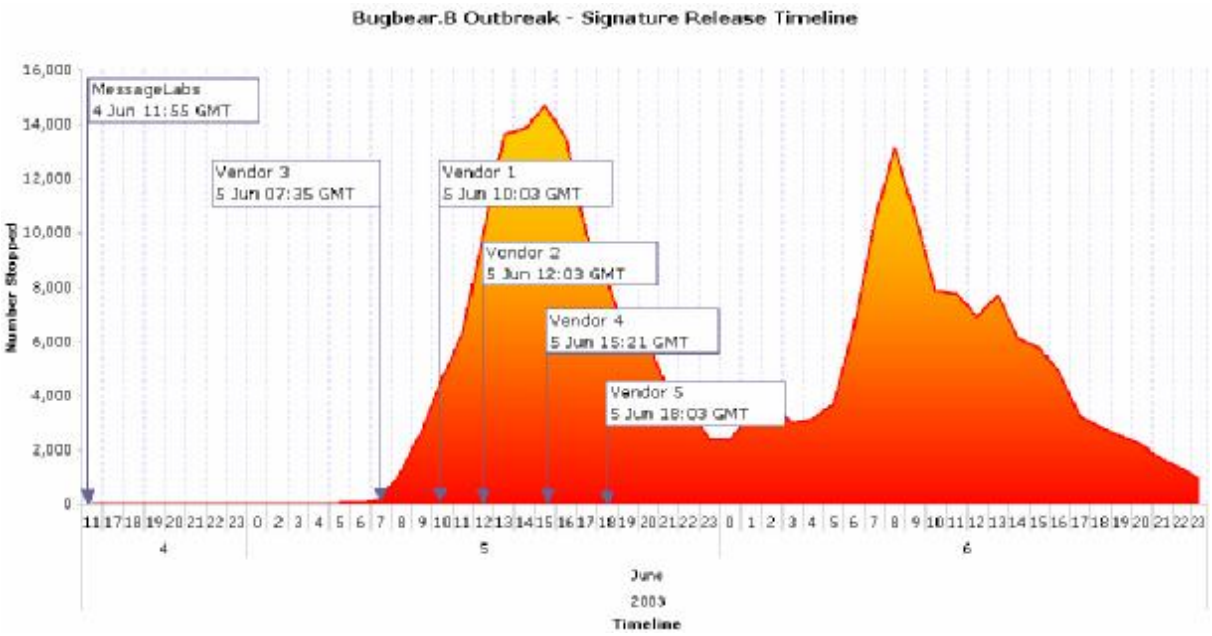
Poznámka: Část, zabezpečující se o stahování aktualizací z Internetu je v případě všech známých AV natolik „inteligentní", že při každém pokusu stahuje pouze nové aktualizace, nikoliv znovu ty, které se vyskytují na stanici uživatele.

Nutnost časté aktualizace dokládají i následující grafy z produkce společnosti MessageLabs[27]. Grafy znázorňují první zachycení uvedené infekce systémy MessageLabs a následně i rychlost reakcí antivirových společností (Vendors). Vodorovná osa vyjadřuje čas a svislá množství zachycených infikovaných emailů uvedeným virem.

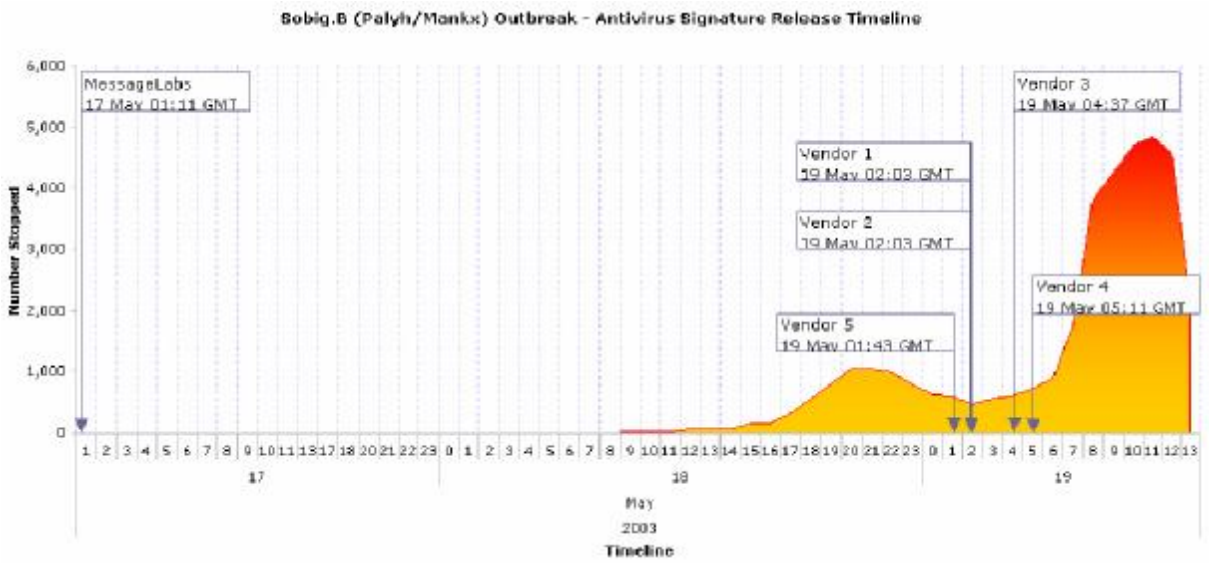


Obrázek 20 Virus Win32/Fizzer

[27] Společnost MessageLabs (www.messagelabs.com) zachytává milióny e-mailů po celém světě a vhodným rozborem vzniká celá řada zajímavých statistik a grafů.



Obrázek 21 Virus Win32/BugBear.B, nejrychlejší nástup viru v celé historii



Obrázek 22 Virus Win32/Sobig.B

2.1.2 Virová databáze

V předchozích odstavcích byl nakousnut problém "virová databáze". Nyní přesněji, o co se jedná.

Virová databáze obsahuje informace, na základě kterých dokáže antivirový skener vyhledávat známé viry. Virová databáze je obvykle označena datem vydání. Antivirový skener dokáže na základě informací z virové databáze detekovat většinu známých virů, které vznikly před datem vydání virové databáze. Pravidelnou aktualizací lze zajistit, že rozdíl mezi současným datem a datem vydání bude co nejmenší a budou tak detekovány i nejnovější přírůstky mezi viry.

Virová databáze obsahuje obvykle následující minimum:

- **název viru,**

- **informace, na základě kterých lze virus detekovat. Například: Signatury, tj. sekvence znaků stabilně se vyskytující v těle jednotlivých virů. Kontrolní součty (CRC) statických částí viru, díky kterým lze snížit riziko falešných poplachů způsobených nešťastným výskytem virové sekvence ve zdravém programu (viz. kapitola o falešných poplaších).**

Informace o virové databázi se mohou obtížně interpretovat, obzvláště ty, které se týkají počtu detekovaných virů. Od tohoto počtu rozhodně nelze odvozovat kvalitu AV programu. Obrovské odlišnosti v počtu mohou být způsobeny například způsobem kalkulace signatur pro generickou detekci. Při užití této metody lze několika signaturami detekovat celé rodiny virů. Záleží pouze na výrobci AV, jak bude informace prezentovat.

2.1.3 Antivirové skenery

Antivirové skenery (od slova „scanner“) jsou nejstarší součástí každého antiviru. Umožňují vykonávat proces skenování (scanning), během kterého jsou vyhledávány počítačové viry. V předchozí části bylo řečeno, že skener vyhledává viry na základě informací z virové databáze. Pokud virová databáze informace o daném viru neobsahuje, „obyčejný“ skener ho nedokáže detekovat. Proto postupem času vznikly metody detekce, které dokážou odhalit i doposud neznámé viry a tohoto nedostatku se tak částečně zbavit. O těchto metodách bude řeč později. Skenery lze rozdělit na hlavní dvě skupiny:

- **on-demand**
- **on-access**

On-demand skener je takový, který vyhledává viry (skenuje) až po vydání požadavku uživatelem (proto on-demand). Požadavek musí být vydán manuálně, obvykle vybráním požadované oblasti pro test (adresáře, pevný disk, disketa atd.) a stiskem tlačítka “start” v antivirovém programu. On-demand skenery se hojně využívaly v době operačního systému MSDOS, dnes přijdou vhod v momentě, kdy počítač není schopen rozumného provozu. Jelikož rychlost není natolik důležitá jako u níže specifikovaného on-access skeneru, řada on-demand skenerů dokáže prohlížet na výskyt virů i interně pakované binární soubory (UPX, Pklite atd.) a archivy vytvořené řadou známých archivačních produktů (WinRAR, WinACE, WinZIP atd.). Velice úzká skupina je schopná i manipulace s případnými infikovanými soubory uvnitř takových archivů.

Provoz on-demand skeneru je tak zjevně pro běžného uživatele až příliš komplikovaný, proto jsou dnes všechny antivirové systémy vybaveny on-access skenerem. On-access skener zcela automaticky a neustále vyhledává viry v datech (nejčastěji v souborech), se kterými přichází uživatel do styku. On-access skener tak může testovat:

- **spouštěné soubory / programy**
- **otevírané soubory**
- **ukládané soubory**
- **systémové oblasti**

Hledat viry ve spouštěných souborech je nutným minimem pro on-access skener. Z principu je zřejmé, že on-access skener provede antivirovou kontrolu souboru ještě před okamžikem, než dojde k jeho spuštění. Pokud by kontrolovaný soubor čistě náhodou obsahoval virus, on-access skener k danému souboru zablokuje přístup, zobrazí varování a čeká do doby, než se uživatel rozhodne, jak s ním naloží. „Otevírání“ souborů je velice širokým pojmem, dochází k němu například i při přesouvání či kopírování. Běžný on-access skener tak dokáže ohlídat před spuštěním / otevřením i infikované přílohy elektronické pošty. Antivirový systém bez kontroly elektronické pošty tak není nutné ihned zatracovat, on-access skener ji na poslední chvíli nahradí (tj. těsně před spuštěním přílohy).

On-access skenery se začaly běžně vyskytovat až s nástupem operačního systému Microsoft Windows 95. Důvodů bylo několik:

- **Operační systém MSDOS byl narozdíl od on-access skenerů až příliš nenáročný na množství operační paměti a výkon počítače. Přítomnost on-access skenerů pod MSDOS tak ostatní činnost počítače citelně zpomalovala a často tak vznikaly ochuzené verze, které dokázaly detekovat pouze nejrozšířenější viry.**

- **Operační sytém Microsoft Windows 95 byl na tehdejší dobu natolik náročný na množství operační paměti a výkon počítače, že přítomnost on-access skeneru nehrála významnou roli. Navíc, celá řada OS Windows je narozdíl od MSDOSu známa tím, že „co chvíle, to jiný čas na dosažení cíle". On-access skenery tak do tohoto prostředí zapadnou dokonale a v dnešní době nelze přesně určit, za jakou prodlevu může on-access skener a za kterou jiná aplikace běžící pod Windows.**

- **Rychlost dnešních počítačů je natolik vysoká, že přítomnost on-access skenerů nelze v řadě případů vůbec pozorovat.**

Aby docílil skener úspěchu, je potřeba zvolit nějakou strategii, podle níž je realizován výběr souborů pro skenování:

- **Prohlížejí všechny soubory (*.*). Virus se sice nikam "neschová", ale celý test trvá citelně déle a navíc může přinést i řadu falešných poplachů.**

- **Prohlížejí soubory podle dlouhého seznamu masek (*.exe, *.doc, *.xl?, *.scr atd.). Virus se může vyhnout detekci v případě, kdy infikuje kupříkladu soubor typu EXE, který má nestandardní příponu (tj. nemá příponu .EXE). Proces skenování je v tomto případě nejrychlejší.**

- **Nahlížejí do hlaviček všech souborů a na základě rychlého úsudku rozhodnou, zda bude soubor podroben detailnímu průzkumu v podaní skeneru (tj. zda by mohlo jít o soubor vhodný pro infekci virem). Obvykle je tato metoda kombinována s bodem č.2. Výsledkem je nejlepší poměr mezi rychlostí a spolehlivostí.**

U skenerů jsou nejvíce ceněny dvě věci: detekční schopnosti (vysoká úspěšnost, minimum falešných poplachů) a rychlost. Detekční schopnosti jsou závislé jak na samotném provedení skenovacím „motoru" (tj. emulátor kódu, proces hledání signatur), ale i na kvalitě virové databáze. Rychlost je ovlivněna způsobem manipulace se soubory a s informacemi ve virové databázi. Rychlost může být ovlivněna například využitím dynamické cache paměti či způsobem prohledávání virové databáze (lineární, binární atd.).

V žádném případě není doporučeno kombinovat on-access skenery více antivirových systémů na jednom PC ! Může docházet k vzájemným kolizím, „padáním" operačního systému, ale i k neschopnosti detekovat jakýkoliv virus !

2.1.3.1 Skenery uvnitř[28]

Na počátku éry skenerů byla využívána metoda, vyhledávající viry na základě skupiny (sekvence, řetězec) instrukcí, které byly pro daný virus typické. Jednoduše řečeno, virová databáze byla naplněna sekvencemi známých virů a skener tyto sekvence vyhledával v jednotlivých souborech, popřípadě systémových oblastech disku. Pokud byla sekvence z virové databáze totožná se sekvencí v souboru, skener ho považoval za infikovaný, což oznámil i uživateli. Pro spolehlivější detekci s minimem falešných

[28] Antivirové společnosti tají využívané technologie a tak není možné detailněji popisovat jejich činnost.

poplachů používaly některé antivirové programy více sekvencí pro detekci jednoho viru. Vedlejším efektem je v tomto případě i vyšší schopnost rozpoznat novou, dosud neznámou variantu existujícího viru. Rychlost stoupla od chvíle, kdy antiviry vyhledávaly tyto sekvence pouze v místech souboru, kde se daly očekávat (na konci, na začátku). Na druhé straně toho využívali i autoři virů, kteří se snažili umístit tělo viru někam, kde by ho antivirus nenašel (obvykle do středu souboru). Významné snížení množství falešných poplachů přinesla tzv. exaktní identifikace, kdy po nalezení sekvence ještě skener spočítá kontrolní součty konstantních oblastí v těle viru, porovná je s informacemi ve virové databázi a pak teprve upozorní uživatele na téměř jistou přítomnost viru. Exaktní identifikace umožňuje i jemné rozlišování variant jednotlivých virů, o tomto v části věnované pojmenování virů.

Kromě exaktní identifikace známe i generickou detekci. Z velké části je popsána v kapitole o falešných poplaších.

Výběr spolehlivé sekvence býval relativně snadnou záležitostí. Autoři virů se proto pokoušeli znesnadnit detekci svých dílek tím, že začali psát zakódované viry. V takovém případě je možné sekvenci vybrat pouze z velmi malé části kódu - dekryptovací smyčky. Zbytek těla viru je v každém exempláři jiný. Opravdový problém ovšem začíná až s příchodem polymorfních virů, které umí generovat různé tvary dekryptovacích smyček. Pro některé z nich je sice možné stvořit sekvence (nebo několik sekvencí), která virus zachytí, ale ta už obsahuje tolik variabilních částí, že se často najdou i zdravé programy, ve kterých nějaký fragment kódu nebo dat takové sekvenci vyhovuje. Většina polymorfních virů ale generuje takové dekryptory, že nelze hledání podle sekvencí použít. Skenery se nějaký čas snažily o rozpoznávání polymorfních virů pomocí jednoúčelových funkcí, ale to byl vlastně krok zpět. Moderní skenery proto obsahují emulátor strojového kódu, kterým se pokouší emulovat provedení smyčky, a pak mohou hledat sekvence až v dekryptovaném těle viru. Dnešní emulátor kódu bývá natolik propracovaným systémem, že za jeho asistence není větším problémem detekovat jakýkoliv, například i ten nejsložitější polymorfní virus. Emulátor kódu znamenal i příchod heuristické analýzy.

2.1.3.2 Heuristická analýza

Heuristická analýza je další z mnoha kouzelných termínů, které věrně doprovázejí moderní antivirové programy. V podstatě jde o rozbor kódu hledající postupy pro činnost virů typické nebo nějak podezřelé. Tímto způsobem lze odhalit i dosud neznámé viry.

Heuristická analýza měla odjakživa své zastánce i odpůrce. Zatímco zastánce těšila možnost detekce neznámých virů, odpůrce strašila zvýšená hladina falešných poplachů. Dnešní heuristická analýza bývá často natolik propracovaná, že výskyt falešného poplachu je spíše náhodou. To však nic nezměnilo na tom, že odpůrci existují i nadále.

Starší heuristické analýzy by bylo možné označit za „pasivní“. První z nich byla použita v antivirech F-PROT a TBAV. Pasivní heuristika prohledávala soubory a hledala v nich typické příznaky (sekvence znaků) pro virus. Pokud bylo takových příznaků (často označovány jako flags) nalezeno dostatečné množství, byl takový soubor považován za napadený. Příznakem mohlo byt např. volání nějaké služby INT 21h, zápis do souboru apod. Nevýhodou bylo, že pasivní heuristika nedokázala proniknout pod "povrch" kódovaných či polymorfních virů a tak složitější nedokázala detekovat. Neaktivní heuristiku mohly viry snadno oklamat. Pokud například heuristická analýza využívala sekvenci v hexadecimálním tvaru: B440CD21 (vyjadřuje souhrn instrukcí mov ah,40; int 21h), mohl virus stejnou činnost vyvolat jinou sekvencí: B43FFEC4CD21 (mov ah,3f; inc ah; int 21h).

V dnešní době je ve všech známých případech využita „aktivní“ heuristická analýza. Základem je emulátor kódu a s ním spojená existence virtuálního prostředí

počítače. Emulátor kódu dokáže spustit soubor a jeho část od-emulovat podobně, jako by ho spustil sám uživatel. Emulátor kódu ovšem veškerou činnost provádí ve virtuálním prostředí a skutečný počítač uživatele tak v případě „spuštění" infikovaného souboru nemůže ohrozit. Pokud by byl zpracovávaný soubor infikován, emulátor v podstatě vykoná i činnost viru, od-emuluje dekódovací smyčku (dekryptor) a dostane se tak přímo na vnitra viru. V tomto stavu může již skener vyhledávat podle sekvencí. Pokud jsou během emulace sbírány informace o aktivitách programu (např. proces přesměrování vektorů přerušení...), může být do akce zapojena i aktivní heuristická analýza, která na základě získaných informací vyhodnotí, zda se ne/jedná o virus. Jelikož emulace programu probíhá pomaleji než při skutečném spuštění programu, má emulátor nastaven tzv. timeout - tj. čas (či počet instrukcí), po kterém se chod emulátoru na aktuálním souboru zastaví. Tohoto časového limitu některé viry dokážou využít pro svůj prospěch, viz. techniky virů – EPO.

Smutnou zprávou je, že celá řada dnešních heuristik zaspala dobu. I u jinak velice vyspělých AV systémů lze spatřit heuristickou analýzu, jejíž vývoj skončil v lepším případě během příchodu makrovirů a nedrží se tak s dnešními trendy. Aby byla heuristická analýza účinná i v dnešní době, je nutné ji postupně rozšiřovat o detekci dalších, nově se objevujících skupin infiltrací. První heuristické analýzy tak detekovaly vesměs pouze souborové viry pro DOS, popřípadě boot viry. S příchodem Wordu 6.0 a Excelu 5.0 (oba Microsoft) byla působnost některých heuristik rozšířena i o detekci neznámých makrovirů (pochopitelně až po jejich obecné detekci – skenerem). V dnešní době lze najít opravdu malé množství těch, které dokáží detekovat neznámé viry pro Windows, a ještě menší těch, které je dokáží detekovat s vysokou úspěšností. Významnou roli v tomto hrají viry, jenž jsou napsány ve vyšších programovacích jazycích (Delphi, Visual Basic apod.) a jejichž alternativy se pro operační systém DOS neobjevovaly v takovém množství (resp. objevovaly, ale často šlo o nerozšířené viry). Problém spočívá v daleko komplikovanější struktuře spustitelných souborů, které takový vyšší programovací jazyk vyprodukuje, ale i v množství reálně se šířících virů tohoto ražení.

2.1.3.3 Falešné poplachy

Již od začátku doprovázejí všechny skenery tzv. falešné poplachy (false positives). Za falešný poplach označujeme situaci, kdy antivirus detekuje virus i když ve skutečnosti o žádný nejde. Následuje výpis některých okolností, které mohou vést k falešným poplachům a zároveň k znehodnocení celého antiviru:

- **Použití krátkých sekvencí pro detekci virů.**

  Při použití krátkých sekvencí (obvykle délky několika bajtů) se zvyšuje pravděpodobnost, že stejná sekvence bude nalezena i ve zcela nezávadných oblastech (soubory, systémové oblasti).

- **Použití nesprávných sekvencí pro detekci virů.**

  Příkladem může být sekvence, reprezentující kus textu. Text se sice vyskytuje v těle viru, ale může být například i součástí zcela nezávadného dokumentu.

- **Zvýšení citlivosti antiviru za účelem zvýšení úspěšnosti detekce.**

  Přílišné zvýšení citlivosti antiviru může nejen zvýšit úspěšnost detekce, ale i množství falešných poplachů. Typickým příkladem je „přecitlivělá" heuristická analýza.

Tomuto mohou AV společnosti předcházet částečně tím, že před vypuštěním nové virové báze provedou důkladný test na rozsáhlé sbírce souborů.

Naopak výběrem vhodných sekvencí lze docílit i tzv. generické detekce. Obvykle jde o „univerzální" sekvencí znaků, která se vyskytuje v podobné, nebo v nezměněné formě ve více virech současně. Může jít například o některé typické replikační mechanismy virů, které se již z jejich povahy musí nutně objevit. Generická detekce se často uplatňuje při detekci mnohých variant makrovirů, které vznikají z původní verze jen drobnými úpravami (odlišná destrukční akce, jiné vypisované zprávy atd.). Také bývá často využívána při detekci virů, které vznikly odlišným nastavením jednoho z mnoha generátorů virů. V případě jednoho starého generátoru virů pro MSDOS bylo možno několika sekvencemi odchytit několik tisíc virů, které vznikly různou kombinací nastavení onoho generátoru !

V nástupem virů šířících se elektronickou poštou se generická detekce využívá k detekci „exploitů", tj. pokusu o zneužití bezpečnostních děr aplikace Internet Explorer (a Outlooku). Z kapitoly o virech / bezpečnostních dírách je zřejmé, že techniky pro vyvolání určitého „nebezpečného stavu" bývají velice podobné a právě tuto část kódu lze využít pro vytvoření signatury, tj. ke generické detekci exploitu a zároveň tak i případného viru, který ho využívá[29].

2.1.4 Kontrola integrity

Kontrola integrity je založena na porovnávání aktuálního stavu souborů a oblastí na disku s informacemi, které si kontrolní program (integrity checker) uschoval při posledním spuštění, popřípadě při jeho instalaci. Jestliže se do takto chráněného počítače dostane virus, zdvořile na sebe upozorní změnou některého z kontrolovaných objektů a může být zachycen kontrolou integrity. Spolehlivě tak lze zachytit i nové, doposud neznámé viry pro skener či dokonce heuristickou analýzu. Nasbírané informace lze často využít i k velice účinnému léčení, ale o tom až později. Aby se tato idylka proměnila v realitu je nutné kontrolu integrity správně nainstalovat a hlavně se o ni správně a pravidelně starat. To je jeden z důvodů, proč kontrola integrity pomalu z antivirových systémů mizí, popřípadě je využita v takové formě, že uživatel nemá o její přítomnosti potuchy. Běžné kontrole integrity je totiž potřeba věnovat více času než skenerům.

Kontrola integrity pracuje v režimu on-demand, takže uživatel ji musí aplikovat ručně. Po každém průchodu kontroly integrity diskem se musí uživatel správně rozhodnout, zda pozměněné objekty jsou jeho dílem, popřípadě dílem nějakého viru. Pokud se rozhodne špatně (tj. změny způsobené virem označí za změny, které způsobil systém, popřípadě uživatel), všechny další kontroly postrádají smysl. Ve skutečnosti infikované objekty (soubory, systémové oblasti) budou považovány za nezávadné, informace které by pomohly virus odstranit budou přepsány informacemi, které již vyjadřují infikovaný stav, atd. To jestli kontrola integrity dokáže objevit virus je zcela v rukou uživatele. Dalším předpokladem pro úspěšné nasazení je provedení instalace kontroly integrity v době, kdy počítač není infikován. Kontrola integrity by v opačném případě vycházela z uložených informací, které by byly již ve stavu infekce. Dalším negativem je, že kontrola integrity dokáže zachytit virus až ve chvíli, kdy je aktivní (již "řádí" na pevném disku uživatele).

Mezi informace, které si kontrola integrity zapisuje během průchodu disku patří například:

- **délka souboru**
- **datum souboru**

[29] Vzhledem k tomu, že exploity využívají převážně pouze viry, lze takto odhalit celou řadu nových virů šířících se poštou a využívající konkrétní exploit.

- **atributy souboru**
- **kontrolní součet souboru (popřípadě jeho části)**
- **kontrolní součet systémové oblasti**
- **informace pro budoucí léčení (část hlavičky souboru, část systémové oblasti apod.)**

Jak už bylo řečeno, klasická forma kontroly integrity postupně mizí, místo ní lze najít u některých antivirů kombinaci: kontrola integrity & on-demand skener. Díky tomu, že kontrola integrity provádí jen několik základních operací, je podstatně rychlejší než skener. Některé antiviry tohoto využívají a tak on-demand skener ve skutečnosti provádí kontrolu integrity. Pokud se skutečný stav liší od stavu, který si kontrola integrity uložila při předchozím testu, dojde k vyšetření změny skenerem (i s využitím heuristické analýzy, pokud ji antivirus nabízí). Stručně řečeno, skener je použit pouze v případě modifikovaného či nového souboru na disku. Výsledkem tohoto spojení je vyšší rychlost při zachování stejné spolehlivosti detekce.

Důležitým poznatkem je, že kontrolu integrity lze obvykle aplikovat pouze u pevných disků. V případě disket a ostatních médií by se musely přenášet i posbírané informace kontroly integrity. Uživatelé, mezi kterými by tato média putovala by navíc museli používat kontrolu integrity stejného výrobce - stejného antivirového systému. Kontrola integrity nepotřebuje pro svůj chod virovou databázi. Viry dokáže (za spolupráce uživatele) detekovat pouze na základě změn v systému.

2.1.5 Monitorovací programy

Monitorovací programy (behavior blocker) obecně hlídají změny v nastavení systému a chrání systém před replikací viru a to na základě neustále kontroly a posléze aktivního zastavení takové ilegální akce.

Monitorovací programy jsou aktivními nástroji pro detekci virů na základě změn v chování systému, a to v reálném čase. Tyto programy zabraňují nelegálním akcím a signalizují, kdykoliv se cokoliv v systému pokouší o nějakou podezřelou akci, která má charakteristiky chování viru, popř. jinak škodlivého, ilegálního chování, např. pokus o zápis do chráněných souborů, pokus o formátování disku atd. Protože však virus není ničím jiným než sekvencí příkazů, je zde značná pravděpodobnost, že i legitimní programy mohou provádět stejné akce, a povedou ve svém důsledku k signalizaci stejně jako virus (např. sebe-modifikující programy).

Monitorovací program předpokládá, že viry provádějí akce, které jsou svou povahou podezřelé, a proto mohou být detekovány. To však nemusí být vždy platné tvrzení. Nové viry mohou využívat nové metody, které mohou být mimo působnost monitorovacího programu. Takový virus nebude monitorovacím programem detekován.

Techniky, které se využívají u monitorovacích prostředků pro detekci chování podobného chování viru, rovněž nejsou neselhávající. Navíc jsou monitorovací programy rovněž napadnutelné. Existují totiž viry, které obejdou nebo zcela vypnou celý monitorovací systém, viz. kapitola tunelující viry.

Na rozdíl od skenerů není tak snadné používat monitorovací programy. Dost totiž záleží na jejich nastavení. Obecně řečeno, je-li nastavení příliš jemné, bude program neustále hlásit poplach (a uživatele zákonitě otráví a sníží jeho důvěru v software). Je-li naopak nastavení příliš hrubé, program nebude detekovat téměř nic (tedy ani některé viry).

V případě, že monitorovací program ohlásí pokus o nějakou z jeho pohledu podezřelou akci, je na uživateli, aby byl schopen posoudit, zda se jedná o falešný poplach

(zda daný program provádí legitimní činnost), nebo o pokus viru. Tím je kladen na uživatele značný nárok. Na druhé straně jsou tu však i výhody monitorovacích programů:

- **Tyto programy mohou být při optimálním nastavení velmi citlivými detekčními prostředky a mohou zachytit i některé dosud neznámé viry.**

- **Monitorovací software nemusí být tak často obnovován. Není závislý na konkrétních virech, a proto v podstatě nevyžaduje obnovu, pokud nedojde ke vzniku nějaké nové virové techniky.**

2.1.6 Další součásti

Mezi méně viděné součásti některých AV systémů můžeme zmínit například:

- **karanténu (quarantine), kam je možné „zakonzervovat" infikované soubory. V praxi jde o speciální adresář, který je pod kontrolou antiviru a často je jeho obsah zajištěn před vlivem externích faktorů. Karanténa si najde uplatnění v případě, kdy jsou napadeny životně důležité soubory uživatele, které nemohou být současnou verzí AV systému úspěšně vyléčeny. Uživatel tak může tyto soubory bezpečně přesunout do karantény, vyčkat na verzi, která již bude tyto soubory schopna od viru osvobodit a následně je z karantény vrátit zpět na původní místo disku. Těsně před léčením je některými AV systémy využita karanténa jako místo pro tvorbu záložních kopií.**

- **antivirové plug-iny. Nejčastěji je k vidění plug-in ("vsuvka") pro kancelářský balík Microsoft Office. Novější verze Office jsou ze strany Microsoftu uzpůsobeny tak, že obsahují rozhraní i prostor uvnitř produktu pro instalaci právě těchto antivirových plug-inů. Komunikační rozhraní umožňuje plug-inu testovat otevírané dokumenty, sledovat činnosti MS Office apod. Z pohledu AV systému jako celku, jde o velice nezajímavou součást. Plně ji totiž dokáže nahradit on-access skener.**

- **antivirový spořič obrazovky. Jde o speciální verzi on-demand skeneru, který se spouští automaticky místo klasického spořiče obrazovky (screensaver). Během skenování je v některých případech možné vyvolat původní spořič.**

2.2 Antivirová ochrana bran, groupware a serverů

Předchozí kapitola pojednala o ochraně cílových stanic (tedy i běžných počítačů v domácnosti), nyní následuje výklad ochrany všech ostatních lokalit sítě.

2.2.1 Zabezpečení vstupní brány (gateway)

Narozdíl od klasické ochrany souborových serverů jde o v tomto případě o relativně mladou skupinu antivirového softwaru. Zatímco dříve spolehlivě k šíření virů sloužila jako prostředek disketa, dnes to jsou tři významné aplikační protokoly: HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol).

Přes všechny zmíněné protokoly mohou de fakto pronikat všechny typy infiltrací počínaje souborovými viry, konče nebezpečnými ActiveX aplety. Nejvýznamnější podíl na celkovém množství infekcí nese protokol SMTP, tvořící základ poštovních serverů. Úspěšnost infiltrací šířících se prostřednictvím elektronické pošty a zároveň tedy využívající služby SMTP je viditelná již na první pohled. Narozdíl od ostatních je totiž případný virus / červ doručen nešťastníkovi až „pod nos".

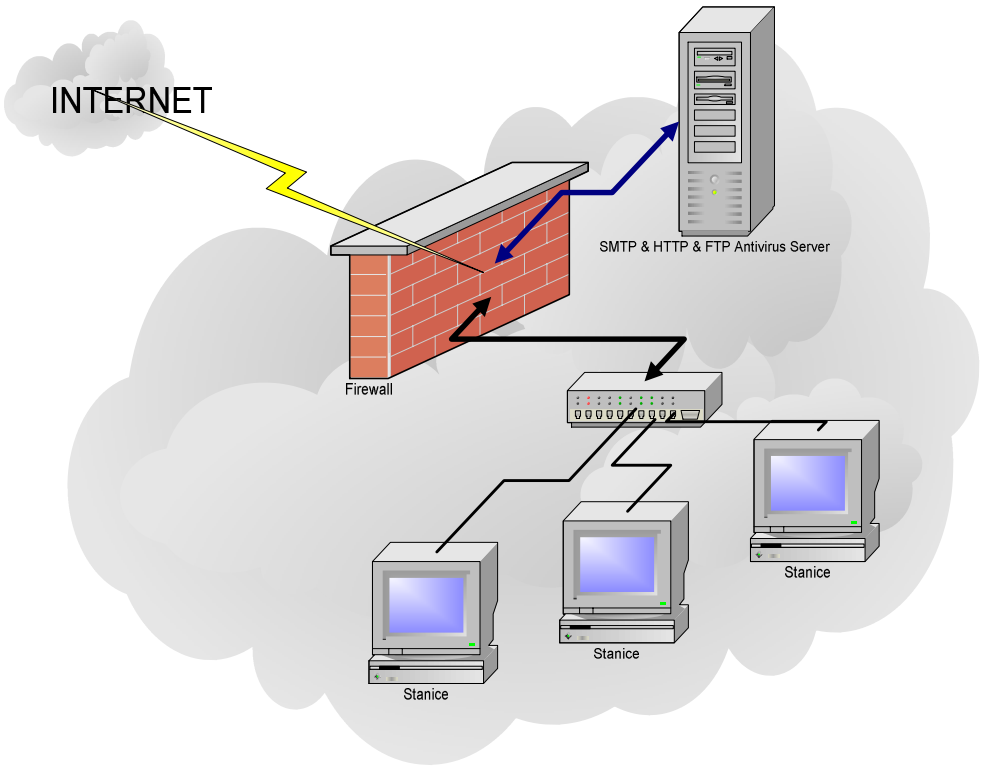
Důležitým poznatkem při ochraně vstupních bran je, že tato řešení chrání počítačovou síť proti útokům z venku, ale už nechrání firemní data proti útoku zevnitř.

Do této kategorie tak můžeme zahrnout antivirové systémy pro ochranu firewallů a pro ochranu e-mailových serverů.

2.2.1.1 Ochrana Firewallů

Všeobecně antivirové programy na vstupních branách – firewallech, mohou fungovat dvojím způsobem.

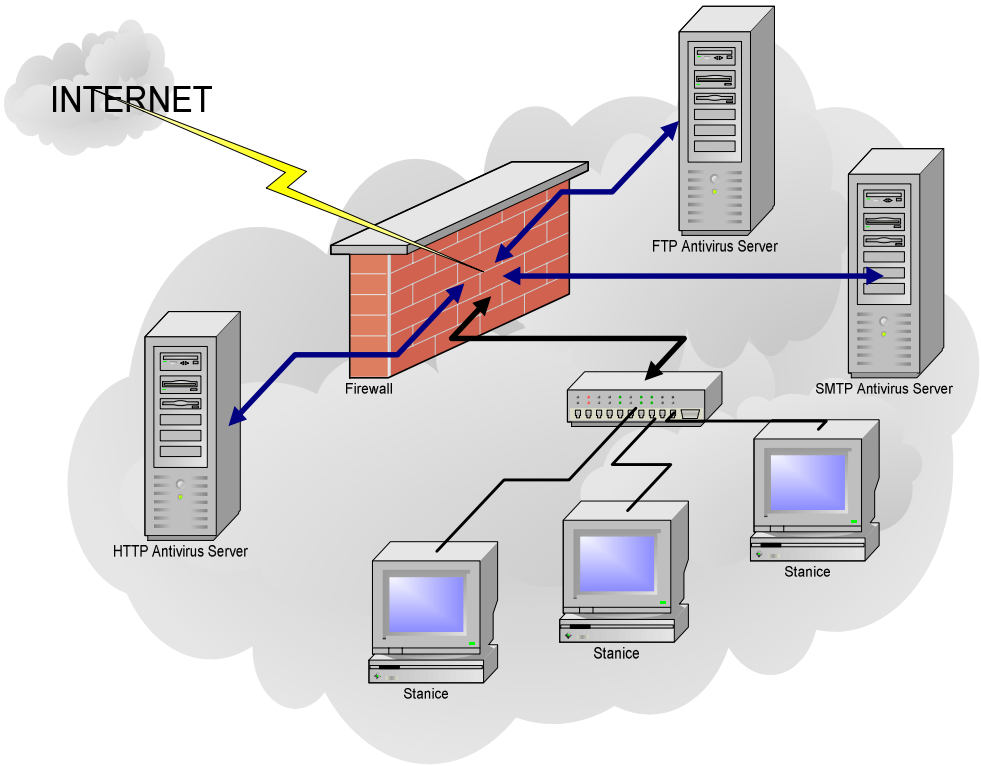
Protokol CVP



Obrázek 23 CVP protokol, možná varianta

Jde o speciální protokol (Content Vectoring Protocol), který slouží k přesměrování paketů na protokolech HTTP, FTP a SMTP na jiný server, kde dojde k jejich zpracování, v našem případě k otestování antivirovým programem a k jejich následnému vrácení na původní firewall. Firewall na základně výsledků AV kontroly tyto pakety pustí či nepustí do vnitřní sítě (popřípadě do vnější). Je zřejmé, že pro úspěšné nasazení tohoto řešení je potřeba jak odpovídající firewall s podporou CVP, tak i příslušný antivirus. Vzhledem k tomu, že antivirový program není přímo instalován na firewall a že proudí velké množství paketů, je nutno toto řešení vybavit rychlým hardwarem a zároveň zajistit mezi firewallem a AV systémem rychlou topologií (tj. bez případných prvků).

V případě velkých organizací je vhodné zátěž rozložit tím, že každý server s AV systémem se bude věnovat pouze jednomu ze tří aplikačních protokolů (HTTP, FTP, SMTP). Zajistíme tím vyšší průchodnost linky.



Obrázek 24 Příklad řešení, kdy je každému monitorovanému protokolu přiřazen server

Alternativou k výše uvedenému řešení může být software společnosti Checkpoint – Firewall-1, kde je možno každý aplikační protokol rozložit mezi dva antivirové skenery. Celkově jich lze připojit až 6. V neposlední řadě existují i řešení s možností clusteringu. Pak není jejich počet teoreticky omezen a hodí se tak především ISP. Průtok dat se pochopitelně odvíjí od počtu připojených AV serverů. Ke clusteringu dochází za podpory softwaru třetí strany, například StoneBeat Security Server od firmy StoneSoft.

Přehled hlavních firewallu, podporujících protokol CVP:

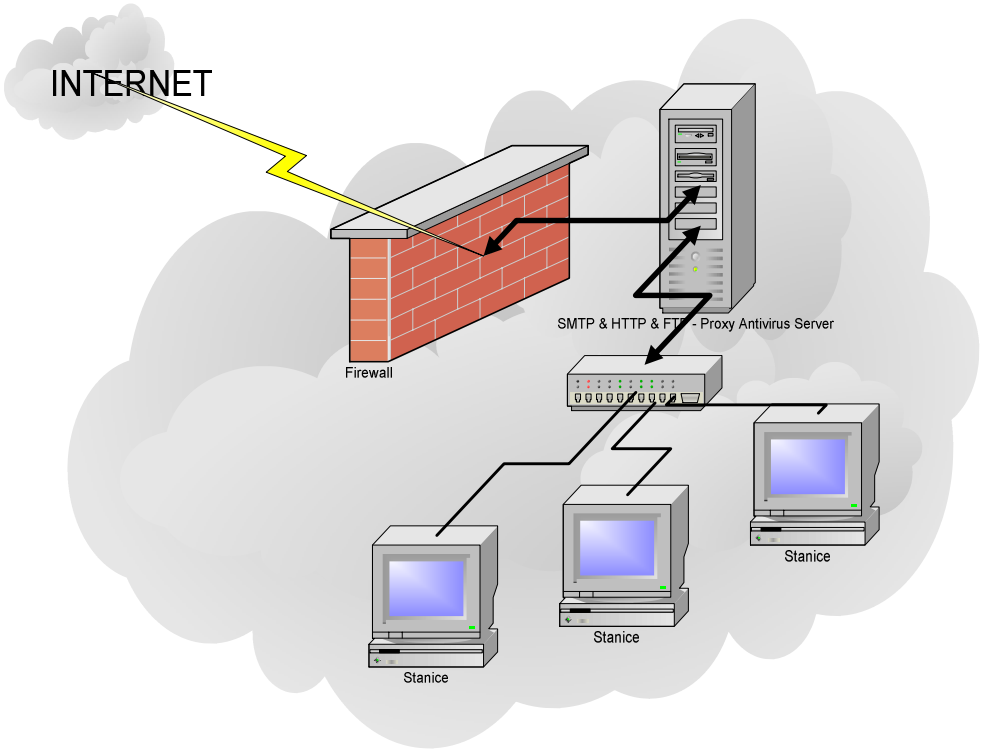
Altavista Firewall
Check Point FireWall-1
Cyberguard Firewall for NT
Gauntlet
Milkyway/SLM SecurIT Firewall for Solaris
Secure Computing Firewall for NT
Secure Computing SecureZone
Sun Solstice Firewall

Firma Checkpoint na základě tohoto rozhraní provádí certifikace antivirových produktů vzhledem k jejich firewallu (www.checkpoint.com).

Proxy architektura

Výhodou těchto řešení je především jejich rychlost i když jen v určitých případech. Pakety procházejí skrze antivirovou kontrolu rovnou ke stanicím uvnitř sítě a nedochází k jejich návratu jako v předchozích případech. Odtud plyne i jednodušší způsob konfigurace tohoto řešení. Naopak za nevýhody lze považovat nemožnost rozložení zátěže na více antivirových skenerů zároveň a paradoxně v některých případech i vyšší cenu.

Cenově dostupnou alternativou může být řešení společnosti Kerio Technologies, jejíž produkty vznikají na území České Republiky. Od počátku roku 2003 nabízí novou verzi softwaru Kerio WinRoute Firewall 5, který je vybaven speciálním rozhraním, komunikujícím s řadou antivirových systémů (pro zajímavost například: AVG, avast!, NOD32, Norton, Sophos, McAfee). Antivirová kontrola probíhá v tomto případě na úrovni HTTP a FTP a k realizaci dochází na stejném stroji jako je umístěn samotný firewall. Řešení je tak vhodné pro malé a větší společnosti.

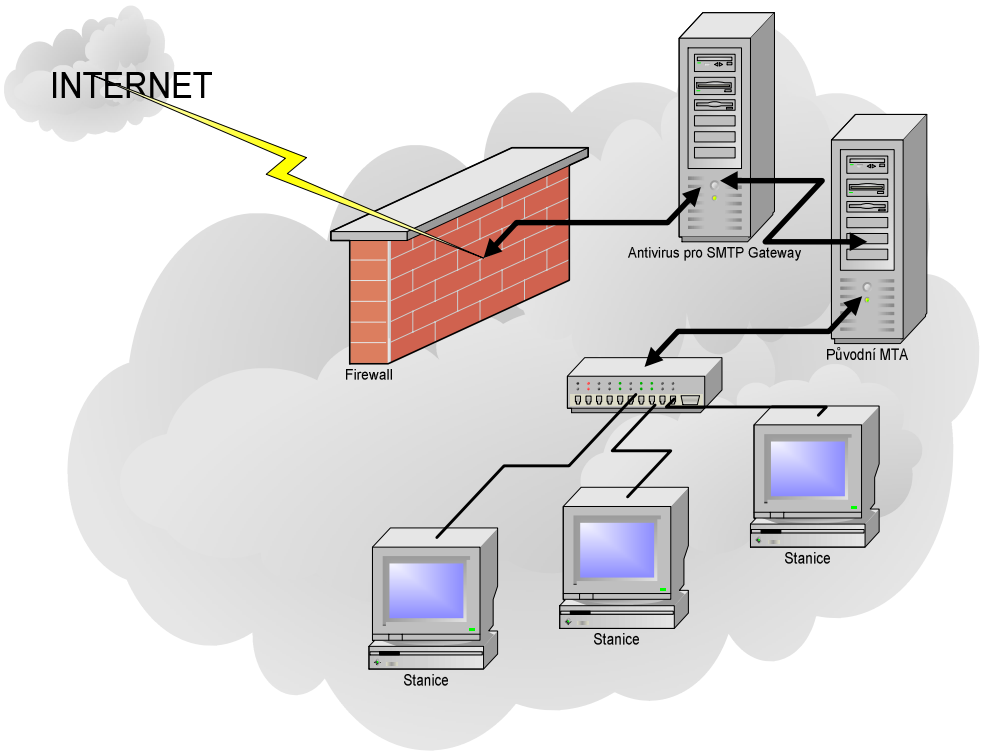


Obrázek 25 Proxy řešení antivirové ochrany

Z výše uvedeného schéma je zřejmé, že antivirový proxy server je vložen mezi firewall a koncové stanice. Přicházející pakety z Internetu jsou složeny do souborového tvaru, zkontrolovány antivirovým systémem, poté opět rozloženy a zaslány na příslušné cílové stanice v lokální síti.

2.2.1.2 Ochrana poštovních brán

Antivirové programy pro e-mailové brány jsou v podstatě „odlehčenou“ verzí toho, co bylo uvedeno výše. Narozdíl od firewallů s podporou protokolu CVP chrání pouze nejdůležitější protokol související s elektronickou poštou – SMTP. I když padlo pouze slovo „jen“, rozšířenost je vyšší než v případě firewallů s podporou CVP. Princip fungování antivirů pro e-mailové brány spočívá v samostatném antivirovém serveru, který je umístěn buď v chráněné nebo venkovní síti a má přiřazeno vlastní jméno v DNS (Domain Name Server). Správnou úpravou příslušných MX záznamů v takových DNS lze docílit toho, že pošta nejdříve putuje na tento antivirový skener (tam jsou pakety složeny do formy e-mailů, dochází k jejich antivirové kontrole, následně jsou opět rozloženy do paketů) a pak teprve na původní „pravý“ podnikový poštovní server.



Obrázek 26 Schéma zapojení antiviru pro SMTP Gateway

Toto řešení a jim podobná řešení jsou velmi rychlá, proto jsou vhodná například pro poskytovatele internetových připojení. Na druhou stranu jsou i velice drahá (i když jen do jistého momentu), jelikož bývají licencována na počet poštovních serverů a nikoliv na počet poštovních schránek či domén (typické pro groupwarová řešení).

2.2.2 Ochrana Groupware serverů

Pokud hovoříme o groupware, pak z pohledu antivirových systémů opět hovoříme jen o poštovních serverech, jelikož elektronická pošta je to hlavní, co zajímá viry a pochopitelně i antivirové systémy.

Mezi nejznámější groupware servery patří:

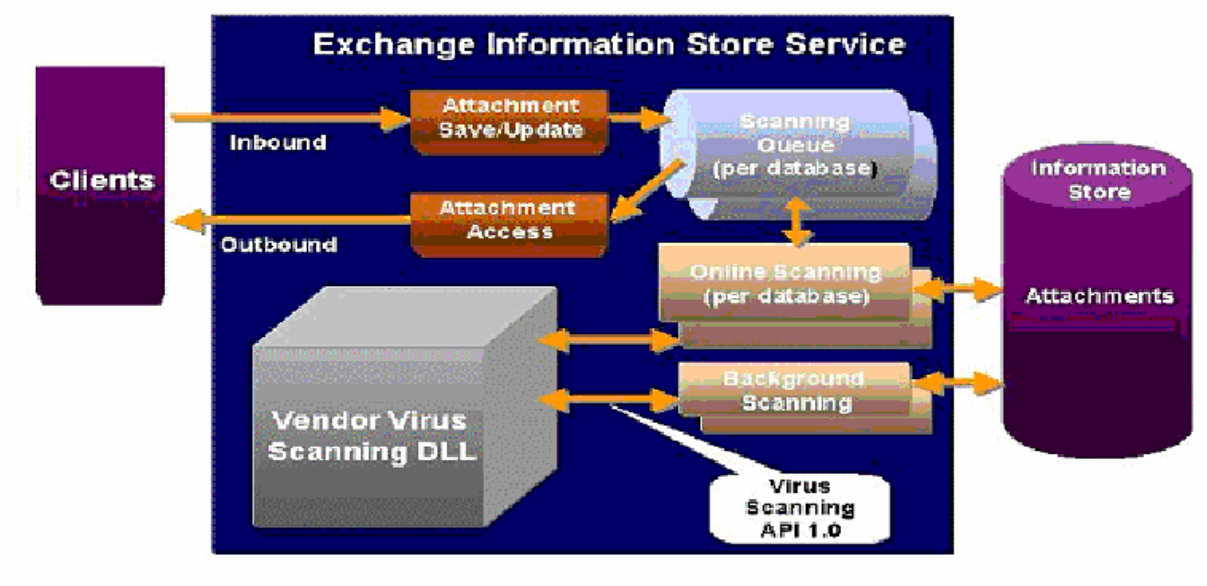
Microsoft Exchange Server
Lotus Notes & Domino Server
Novell Groupwise

Vzhledem k rozšířenosti Microsoft Exchange Serveru bývá toto prostředí ze strany antivirových společností podporován jako první, pokud se do tohoto odvětví zapojí. Opačným případem je GroupWise, kde není nabídka tak rozsáhlá, ale co je zajímavější, že je o to víc zajímavá. Překvapivě je toto prostředí podporováno spíše méně známými AV společnostmi.

2.2.2.1 Microsoft Exchange

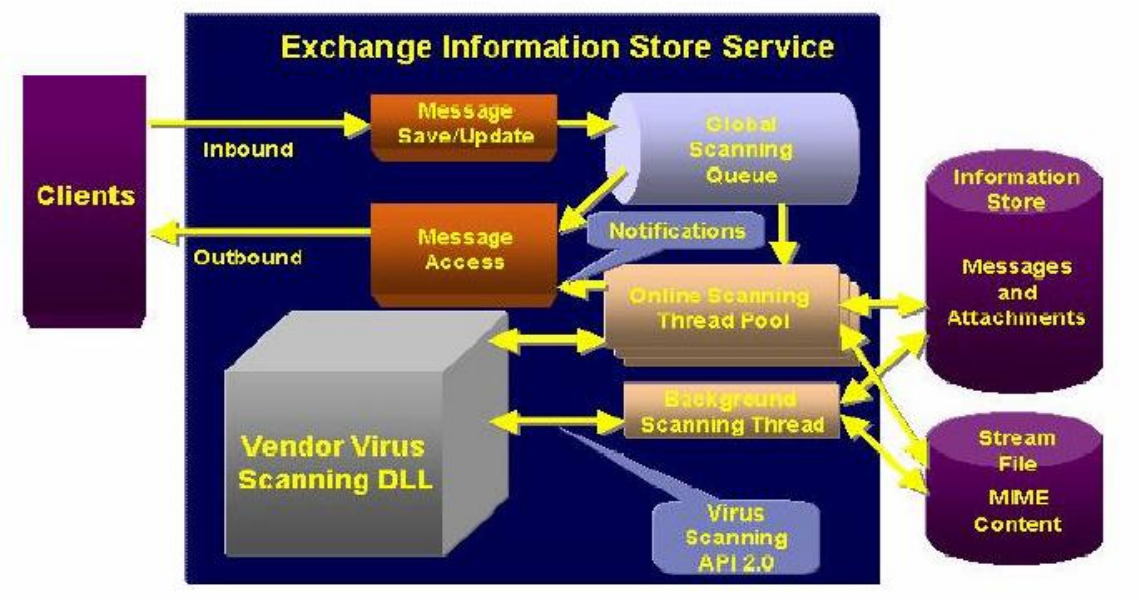
Microsoft Exchange nebyl pro antivirové programy původně příliš navržen. V počátcích tak musely antivirové systémy využívat pro jejich účely zcela nevhodné rozhraní MAPI (nízká rychlost, nemožnost kontrolovat odchozí poštu apod.).

Až při uvedení jednoho za řady servisních balíčků (Service Pack) bylo přestaveno nové rozhraní speciálně navrženo pro antivirové systémy – VSAPI 1.0 (Virus Scanning API). I když bylo uvedené rozhraní ze strany antivirových společností s napjetím dlouhou očekáváno, nakonec přineslo zklamání. Jeden ze spousty nedostatků byl například neschopnost zjistit, odkud kam byl virus zaslán.

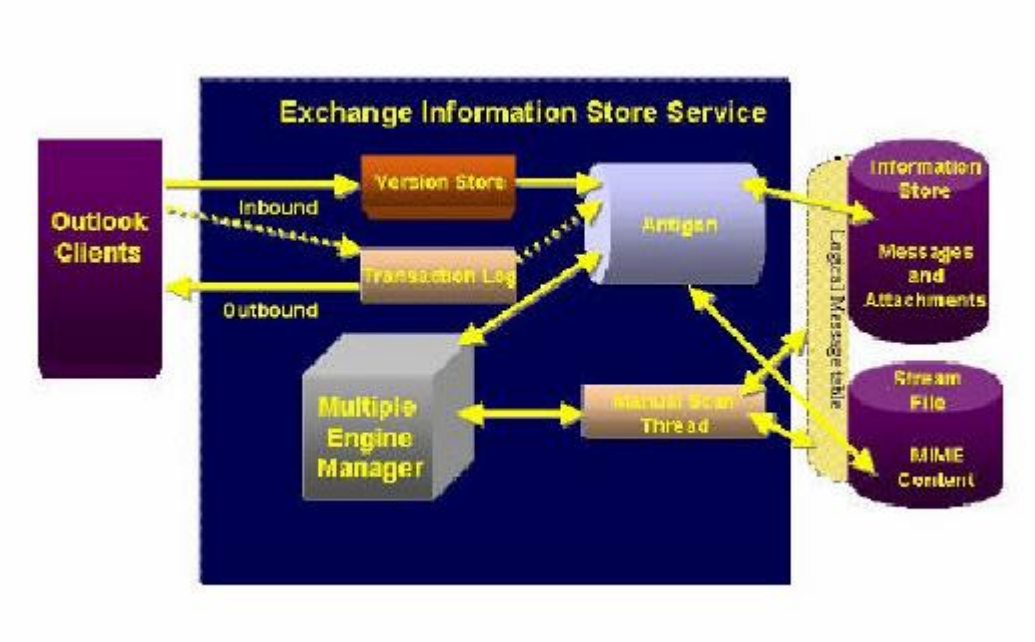


Obrázek 27 Rozhraní VSAPI 1.0

Významnou novinku přinesl až první servisní balík k produktu Microsoft Exchange 2000. Objevilo se totiž rozhraní VSAPI 2.0, které již fungovalo tak, jak si antivirové společnosti představovaly. Kromě toho bylo k dispozici i rozhraní ESEAPI, které Microsoft převzala od společnosti Sybari.



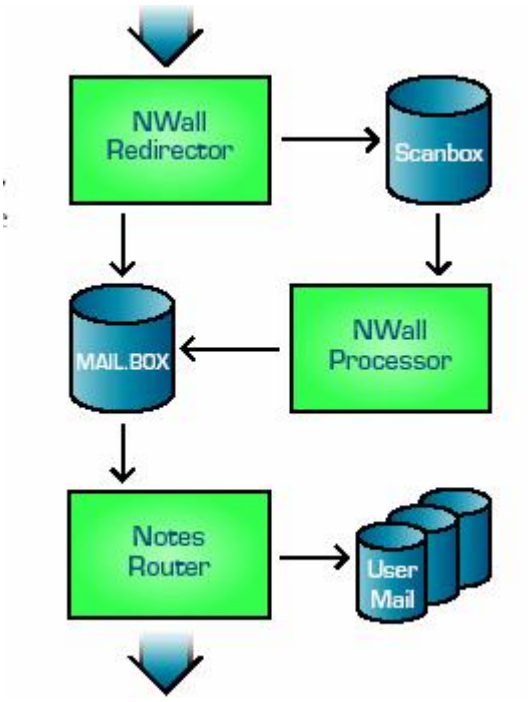
Obrázek 28 Rozhraní VSAPI 2.0



Obrázek 29 Rozhraní ESEAPI

2.2.2.2 Lotus Notes & Domino

Základem antivirového skenování je samostatná databáze, kde jsou přesměrovány všechny transakce a kde dochází ke kontrole. Zkontrolovaná data jsou přesměrována zpět na původní místo určení.



Obrázek 30 Schéma činnosti Lotus Notes & Domino

Z výše uvedených řešení je zřejmé, že není potřeby samostatně vyhrazeného počítače, kde k antivirové kontrole dochází. Antivirový systém se instaluje přímo na groupware server.

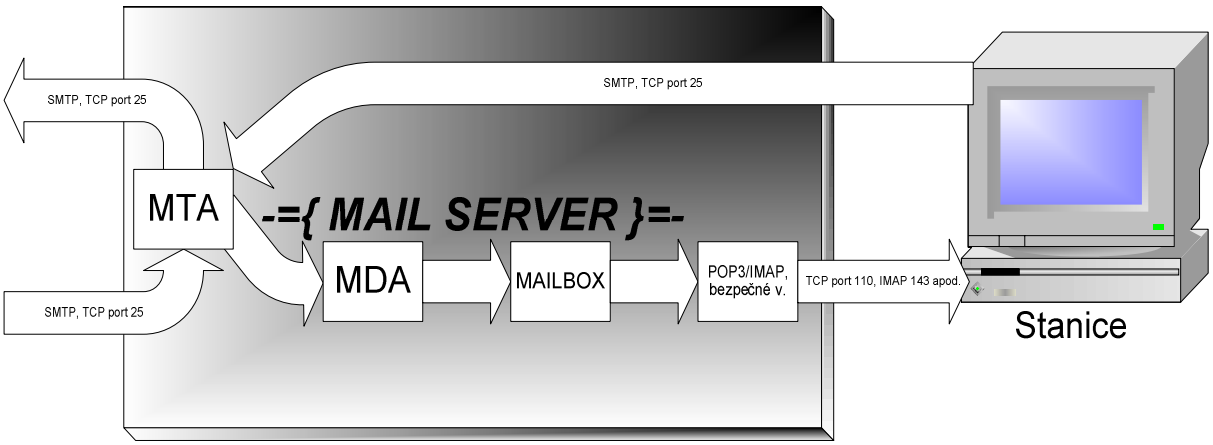
2.2.2.3 Pošta pod Linuxem

V linuxových systémech lze najít základní dva způsoby kontroly pošty:

- **Využití softwaru třetích stran**
- **Konkrétní antivirové řešení**

Prvně zmiňovaná varianta je velice oblíbená hlavně z důvodů nízké ceny (v některých případech i zcela zadarmo). Pro kontrolu pošty je využíván antivirový software, který není primárně určen pro kontrolu pošty, ale jen ke skenování souborů metodou on-demand. Uváděné skenery jsou ovládány prostřednictvím příkazového řádku, čehož využívají produkty třetích stran. Mezi nejznámější patří například Amavis (www.amavis.org). Amavis se stará rozložení přicházejících e-mailů na soubory a jejich „předhození“ on-demand skeneru. Následně je tento skener Amavisem vyvolán a provedena kontrola těchto souborů. Na základě výsledků je pochopitelně rozhodováno o dalších činnostech. Nevýhodou řešení může být nižší rychlost skenování, jelikož on-demand skener musí být při každém procesu skenování znovu a znovu inicializován (tj. včetně načtení virové báze). Naštěstí někteří výrobci antivirových systémů přicházejí s různými démony, které tyto problémy z velké části odstraňují.

Pochopitelně nejlepším řešením jsou antivirové systémy přímo pro konkrétní poštovní servery, či dokonce nezávislá řešení na MTA. Pod Linuxem jich je populární celá řada, jmenujme například: SendMail, Postfix, Qmail.



Obrázek 31 Schéma činnosti poštovního serveru pod Linuxem. MDA (Mail Delivery Agent) uloží e-mail do konkrétní schránky

2.2.3 Ochrana souborových serverů

Ochrana souborových serverů spočívá v nasazení speciální varianty on-access skeneru, který běží přímo na souborovém serveru a který sleduje síťové aktivity související s manipulací se soubory (zápis, čtení, kopírování atd.). Pokud je manipulováno s infikovanými soubory, popřípadě je infikovaný soubor ukládán na server, pak antivirový systém tuto činnost zablokuje a provede definovanou událost. Tato řešení bývají obvykle licencována podle počtu stanic, které se k serveru připojují.

Mezi nejznámější systémy, umožňující vytvoření souborového serveru patří:

- **Microsoft Windows Server**
- **Novell NetWare**
- **Samba server (Linux)**

2.2.3.1 Microsoft Windows Server

Nejširší nabídka antivirových systému je pochopitelně spjata s Microsoft Windows Server. Souborový server nemusí vzniknout nutně na serverové edici operačního systému Windows, jehož přítomnost je pro řadu antivirových systémů pro ochranu souborových serverů podmínkou. Při přivření očí lze za jistý souborový server považovat i síťově nasdílené adresáře na edicích pro stanice -Microsoft Windows 98 / ME či Windows 2000 /XP Professional. V takových případech se je nutné spolehnout na schopnosti antivirových systémů pro stanice, z nichž některé dokážou monitorovat manipulaci ze sítě a to s lokálními soubory.

2.2.3.2 Novell NetWare

Řešení pro Novell NetWare jsou realizovány s využitím NLM modulů, které se příkazovým řádkem Novellu za užití příkazu LOAD instalují rezidentně do paměti. Narozdíl od řešení pro Microsoft Windows Server je konfigurace často realizována prostřednictvím konfiguračního souboru (nikoliv v grafickém prostředí), avšak existují i produkty, jenž konfiguraci zaimplementovali do Novell Directory Services – NDS, kde se již všechno odvíjí na pohodlnější úrovni.

2.2.3.3 Samba server

I když není Samba příliš známým produktem pro vytvoření souborového serveru, jde o velice kvalitní alternativu a co hlavní, je zdarma. Vzhledem k tomu, že je kompatibilní s protokoly využívanými Microsoft Windows Serverem, v řadě případů může zcela nahradit placený produkt od Microsoftu bez toho, aby přistupující stanice zjistily změnu. Microsoft o této alternativě ví a již několikrát dal nahlas najevo, že si ji nepřeje.

Vzhledem k výše uvedeným schopnostem je nutno i pro Samba server najít účinné antivirové řešení. Řadu problémů vyřešil některým antivirovým společnostem produkt Dazuko (www.dazuko.org), který obsahuje rozhraní pro kontrolu nad soubory pod Linuxem. Vhodným využitím tohoto kernel modulu ze strany antivirového systému lze docílit vzniku on-access skeneru pod Linuxem (což dosud není nic běžného). On-access skener pochopitelně dohlíží jak na lokální manipulaci se soubory (například uživatele přihlášeného přímo ke stroji s Linuxem), tak i na manipulaci ze sítě, tedy i stanic s MS Windows přistupujících prostřednictvím Samba serveru k centrálně uloženým souborům.

2.3 Síťové schopnosti antivirových systémů

2.3.1 Centrální správa

Centrální spravování antivirových systémů rozsetých po síti společnosti značně usnadní správci práci. Díky centrální správě má dokonalý přehled nad antivirovými systémy na jednotlivých stanicích i serverech. Kromě toho, že může monitorovat jejich stav, může je i na dálku obsluhovat a vykonávat tak požadované testy či manuální aktualizaci virové báze.

Pro centrální správu může obvykle posloužit libovolný počítač v síti, který vybaven speciálním softwarem (konzole) daného výrobce antiviru, umožňuje přístup. I když to není bezpodmínečně nutné (v závislosti na řešení), někde v síti se nachází antivirový server umožňující komunikaci mezi konzolí správce s jednotlivými kopiemi antivirového systému v síti.

Často jsou pro komunikaci využity standardní protokoly jako TCP/IP či HTTP (HTTPS) či řešení vhodné pro menší společnosti v podobě komunikačního adresáře, který

je síťově nasdílen a přes něhož si konzole správce a stanice s antivirovým systémem vyměňují informace.

K vidění jsou i všemožné odlehčené verze, umožňující správu kupříkladu přes WAP mobilního telefonu.

2.3.2 Zrcadlení aktualizací

Typickou možností síťových antivirových řešení je zrcadlení aktualizací. Programové aktualizace včetně aktualizace virové báze jsou staženy od výrobce AV systému z Internetu pouze dedikovaným počítačem a následně jsou nabídnuty ostatním stanicím a serverům v lokální síti. Přínosem tohoto řešení je významné snížení zátěže Internetové linky, jelikož ostatní stanice a servery provádějí aktualizaci v rámci lokální sítě.

Řešení bývá realizováno nejčastěji prostřednictvím protokolu HTTP či HTTPS, k vidění jsou řešení na bázi síťově nasdíleného adresáře, odkud si jednotlivé stanice požadované aktualizace stahují.

2.3.3 Notifikace

Pochopitelnou součástí jsou moduly, zajišťující informovanost vybraných osob například prostřednictvím e-mailů, broadcastových zpráv či SMS. Aby nedošlo k zahlcení sítě, musí být antivirus rozumně dimenzován tak, aby neinformoval o naprosto každém infikovaném souboru, který byl nalezen během velice krátkého časového období.

2.3.4 Hromadné a centrální instalace

Během nasazování antivirového řešení do velkých společností nemusí být v silách správce sítě „oběhnout“ všechny stanice a příslušný antivirus nainstalovat. Proto je řada antivirových řešení vybavena produktem, umožňujícím vzdálenou instalaci antiviru. V praxi může taková vzdálená instalace proběhnout několika způsoby:

- vzdáleným povelem (platí jen pro NT systémy) za využití protokolu ???
- zasláním speciálního e-mailu na cílovou stanici s žádostí o spuštění přiloženého programu, který zprostředkuje případnou budoucí instalaci antivirového systému (agent)
- modifikací login skriptu, díky němuž je při logování do sítě automaticky spuštěn krátký program (agent), umožňující budoucí instalaci antivirového systému, popřípadě je zahájena přímo instalace antivirového systému. Zmiňované řešení je pochopitelně možno aplikovat pouze v případě, kdy jsou použity login skripty (síť s doménou v případě MS Windows Server či síť pod Novellem).

3 Identifikace infiltrace a následné činnosti

Dosud nebylo příliš zmíněno chování antivirových systémů v případě nalezení infiltrace a způsob, jakým je tato infiltrace signalizována.

3.1 Pojmenování

Každá infiltrace identifikovaná antivirovým systémem má přiděleno vlastní jméno. Již v začátcích bylo zřejmé, že bude potřeba zavést nějaká systém pravidel, podle nichž

budou jednotlivé infiltrace pojmenovány. V roce 1991 tak vznikla organizace CARO (Computer Anti-Virus Researchers Organisation), kterou založili pánové Fridrik Skulason (Virus Bulletin), Alan Solomon (S&S International) a Vesselin Bontchev (Univerzita Hanburg). Systém pojmenování virů, červů a dalších škodlivých kódů je velice rozsáhlý, proto si dovolím uvést jen základní fakta. Jméno škodlivého viru (především virů) by mělo vypadat dle definice následovně:

Family_Name.Group_Name.Major_Variant.Minor_Variant[:Modifier]

Viry by tak měly být řazeny do „rodin" a dále se v rámci nich větvit v závislosti na jejich podobnosti. Kromě řady dalších pravidel bylo usneseno, že mezery v názvu budou tvořeny podtržítkem „_", budou obsahovat pouze alfanumerické znaky [A-Za-z0-9_$%&!'`#-] a podobně. Systém byl společnými silami neustále vyvíjen tak, jak přibývaly nové virové trendy. Řada dnešních antivirů tak využívá relativně podobného způsobu pojmenování. V praxi, co se týče pojmenování, to vypadá následovně:

{typ infiltrace/} jméno infiltrace {.délka v bajtech} {.varianta} {další atributy}

Části mezi {} jsou nepovinné, popřípadě je nelze u daného typu definovat.

Typ infiltrace

Řetězec	Vyjadřuje
Win32 (nebo W32)	Souborový virus pro MS Windows (obecně pro všechny počínaje Windows 95, konče Windows XP).
Win95 (nebo W95)	Souborový virus pro MS Windows 95, 98, ME. Obvykle nejsou schopny provozu pod MS Windows NT.
WM	Makrovirus pro MS Word 6.0 a 7.0.
W97M	Makrovirus pro MS Word 8.0 a výše (počínaje MS Office 97).
XM	Makrovirus pro MS Excel 5.0 a 6.0.
X97M	Makrovirus pro MS Excel 7.0 a výše (počínaje MS Office 97)

Méně využívané

O97M	Malá skupina makrovirů, které se dokáže šířit s využitím více produktů kancelářského balíku MS Office 97 a výše.
PP97M	Malá skupina makrovirů, šířících se v prezentacích aplikace MS PowerPoint 97.
A97M	Malá skupina makrovirů, šířících se ve výstupech produktu MS Access 97.

Některé základní pravidla pro pojmenování byly uvedeny výše, stejně jako naznačení, co představuje část délka v bajtech. Jde pochopitelně o délku viru v bajtech. Jednoduše ji lze získat odečtením délky infikovaného souboru od délky totožného souboru před infekcí.

Viry, které si jsou nějakým způsobem podobné nebo jde dokonce o z velké části totožné viry bývají odlišeny v části varianta přičemž vše ostatní zůstává zachováno. Především v oblasti makrovirů vznikalo obrovské množství variant a tak byly k vidění kousky, jenž začaly u písmene A (například WM/Concept.A) a jelikož označení probíhá vzestupně dle abecedy a variant vznikaly celé desítky, k vidění byly i kombinace AA, AB, AC, ... , BA, BB, BC atd. (WM/Concept.BC).

Další atributy[30]

Řetězec	Vyjadřuje
@mm	zkratka pro „mass-mailing“ viry – tedy ty, které se hromadně rozesílají elektronickou poštou (př.: W97M/Melissa.A@mm).
.intended	Takto bývají označeny nefunkční viry z důvodů chyb v jejich zdrojovém kódu.
.based	Virus vytvořený (based) na základech jmenovaného viru.
.germ	Tímto dodatkem bývají označeny viry, které jsou ve své nulté generaci - tj. po kompilaci zdrojových kódů do EXE souboru.
.dropper	Takto bývají označeny programu s cílem vypustit (drop) jmenovaný virus do systému.
.generic	Virus identifikován generickou detekcí.
.dam	Odvozeno od slova „damaged“. Takto se označují poškozené exempláře virů vlivem vnějších faktorů (chyba na disku, špatně odléčený soubor apod.).

Zajímavé je to taktéž se vznikem jmen. V době rychle se šířících virů elektronickou poštou je nutno rychle vydat příslušné aktualizace antivirových programů. V tomto krátkém okamžiku nemusí dojít k úplně shodnému navržení jména dle ostatních AV společností a tak je až s podivem, pod jakými jmény je daný virus identifikován rozličnými antiviry. Pokud je virus objeven o víkendu, jsou rozdíly ještě viditelnější. Posléze se snaží AV společnosti rozdíly v pojmenování daného viru minimalizovat, ale i tak jsou v globálním měřítku natolik veliké, že vnikl například projekt VGrep (www.virusbtn.com/resources/vgrep).

Pokud je virus pojmenován v zahraničí, zároveň se jedná o původem český virus a „nešťastnou“ náhodou je jméno odvozeno ze slov, vyskytujících se přímo v těle viru, může dojít pro českého občana k humorné situaci. Existuje několik případů, kdy jméno viru působí směšně až vulgárně.

3.1.1 VGrep (www.virusbtn.com/resources/vgrep)

Jde o speciální projekt, jenž má uživateli říci, jaká jiná jména uživatelem uvedeného viru používají ostatní antivirové systémy. V podstatě jde o obrovskou databázi údajů, která vznikla skenováním rozsáhlé virové sbírky rozličnými antivirovými skenery. Reporty použitých antivirových systémů jsou upraveny tak, aby bylo zřejmé, jaké jména byly přiděleny dotyčnými antiviry pro každý konkrétní soubor ve sbírce.

Kupříkladu zadáním řetězce Opaserv.A na výše uvedené adrese zjistíme, jaká „synonyma“ používá pro tohoto populárního červa celá řada antivirových systémů.

ALWIL	Win32:Opas [Wrm]	*H+BEDV*	Worm/OpaSoft
GRISoft	I-Worm/Opas.A	*Kaspersky Lab*	Worm.Win32.Opasoft.a
SOFTWIN	Win32.Worm.Opaserv.A	*Dialogue Science*	Win32.Opasoft
Frisk Software	W32/Opaserv.worm.A	*McAfee*	W32/Opaserv.worm.a

[30] Některé antivirové produkty mohou používat odlišná označení, popřípadě zkrácená označení.

IKARUS	Worm.Win32.Opasoft.A	*MKS*	[undetected]
Symantec	W32.Opaserv.Worm	*ESET*	Win32/Opaserv.A
Panda	W32/Opaserv	*Trend Micro*	WORM_OPASERV.A
GeCAD RAV	Win32/Opaserv.A.worm	*Sophos*	W32/Opaserv-G
CA VET	Win32.Opaserv.A	*CA InoculateIT*	Win32/Opaserv.A.Worm
VirusBuster	Worm.Opaserv.A	*HAURI*	Worm.Win32.Opaserv

I když se to vysloveně nabízí, podle údajů uvedených v databázi VGrep není vhodné testovat kvalitu antivirových skenerů, jelikož jde úmyslně o silně nevytříděnou sbírku malwaru a dalšího „smetí“ !

3.2 Činnosti po identifikaci

Antivirový systém dokáže aplikovat celou řadu činností, které mu uživatel předem definuje, popřípadě je zvolí až při samotné identifikaci škodlivého kódu. Jednou z krajních a zároveň nejspolehlivějších metod je smazání dotyčného infikovaného souboru. Kromě souborového viru tak často dojde i ke ztrátě důležitých dat, které daný soubor obsahoval. Dočasnou alternativou je přejmenování souborů tak, aby ho nebylo v budoucnu možno spustit a opět z něj aktivovat virus. Velice podobným zákrokem je přesunutí souboru do karantény, pokud to daný antivirus nabízí. Každého bude jako první zajímat, zda je možné škodlivý kód odstranit nedestruktivně a to formou „léčení“ infikovaného souboru. Léčení můžeme rozdělit na základní skupiny:

- **algoritmické**
- **heuristické**
- **speciální**

Kromě výše uvedených činností můžeme definovat i informační.

3.2.1 Algoritmické léčení

Při aplikaci prvně jmenované metody je plně spoléháno na přesnost informací o identifikovaném viru (kam si ukládá původní údaje z hlavičky ? jaká je jeho délka ? jaká je jeho pozice v souboru ?), které jsou uloženy ve virové bázi. Na základě těchto informací a obecných postupů je identifikovaný virus „vystřihnut“ ze souboru a ten je následně zrekonstruován do původní podoby (především jde o rekonstrukci hlavičky souborů a navrácení původního vstupního bodu – entry pointu). Blíže je k tomuto pojednáno v kapitole o virech.

K úspěšné rekonstrukci může dojít pochopitelně pouze v případě, že jde o nedestruktivní virus, co do způsobu napadení souboru. Ani tím však není zaručen bezproblémový chod souboru do budoucna. Problémy mohou kupříkladu nastat u tzv. „mezerových virů“ (cavity viruses), které napadají oblasti souborů, které neobsahují data, což nemusí nutně znamenat, že jsou v těchto oblastech pouze hodnoty 00h (například). Pokud antivirus takový virus „vystřihne“, na jeho místo musí nutně vložit jiný kód (například znaky „X“). Díky tomu nemusí být vyléčený soubor přesně v těchto místech shodný s původní neinfikovanou verzí a v případě, že je program, obsažený v tomto souboru, vybaven interní sebe-kontrolou, nelze vyloučit, že bude o tomto rozdílu informovat a následně i odmítne spolupracovat.

K podobným činnostem dochází i v případě starých boot virů, avšak s tím rozdílem, že je rekonstruována příslušná systémová oblast. Je nutno navrátit původní obsah systémové oblasti, kterou si nedestruktivní boot virus musel k zajištění své budoucí replikace někam uložit. Záleží opět na antivirovém systému, zda dokáže tento obsah najít a boot virus tak úspěšně odstranit.

V krajních případech lze systémové oblasti nahradit obecně platným kódem, což dokážou jak některé antivirové systémy, tak i příkaz DOSu: FDISK ve spojení s nedokumentovaným parametrem /MBR. Tímto se odkrývá jeden ze špatných zvyků a tím je řešení infekce boot virem formátováním. Ani sebelepším formátováním nelze boot virus z pevného disku odstranit !

Samotnou kapitolou jsou makroviry. Vzhledem k tomu, že formát dokumentů (MS Word) či sešitů (MS Excel) je obecně známý, antivirus dokáže přesně vymezit oblast, kde jsou uložena jednotlivá makra a tedy i případná makra viru. Pokud by nedokázal antivirový systém odlišit, jaká makra jsou škodlivá a která patří k původnímu dokumentu, může v krajním případě odstranit veškerá makra a to bez větších škod, jelikož nejcennější oblast, kterou je beze sporu samotný text dokumentu, zůstane nepoškozen.

Speciální kapitolou jsou pak trojské koně, backdoory, doprovodné viry. „Léčení“ probíhá formou mazání infikovaných souborů, jelikož z předchozích kapitol plyne, že soubory trojských koní, backdoorů a doprovodných virů neobsahují žádná jiná data, než škodlivý program.

3.2.2 Heuristické léčení

Kromě heuristické detekce virů existují i pokusy o jejich heuristické odstraňování. Virus se totiž po svém spuštění dříve nebo později pokusí předat řízení původnímu programu. Pokud se podaří odsimulovat jeho běh až k tomuto bodu, stačí napadený soubor správně zkrátit a všechno je v nejlepším pořádku.

Heuristické léčení sice umožňuje vyléčit i případný neznámý virus, ale vzhledem k tomu, že jde o odstraňování na základě informací získaných až v průběhu samotné heuristické analýzy, nelze touto operací vyloučit poškození souboru, ať už jde o známý či neznámý virus.

Heuristické léčení bylo fenoménem některých starších antivirových systémů (především AVG 3-4, TBAV) a dnes již ho nelze běžně spatřit.

3.2.3 Další metody léčení

3.2.3.1 Očkování souborů

V dávných dobách byly některé antivirové systémy (CPAV) vybaveny možností „očkování“ souborů. Takový soubor, dosud neinfikovaný, byl antivirem prodloužen o krátký kontrolní program. Pokud byl takto upravený soubor v budoucnu infikován, po jeho spuštění uživatelem byl tento upozorněn, že je soubor zmodifikován a následně mu bylo nabídnuto léčení sebe sama. Pokud si vložený kontrolní program před infekcí uložil důležité informace o původní zdravé variantě souboru, pak mu nečinilo problémy virus odstranit a soubor zrekonstruovat do původní formy.

Problémy mohly nastat opět v případě, že šlo o interně se kontrolující programy. Navíc mohly tyto kontrolní programy způsobovat falešné poplachy některých heuristik, jelikož sami o sobě připomínaly formou připojení k souboru virus.

3.2.3.2 Kontrola Integrity

Viry lze úspěšně odstraňovat na základě informací, které si automaticky ukládá kontrola integrity. Pokud je uloženo dostatek informací o původním souboru před infekcí, může být nejen s velikou úspěšností rekonstruován do neinfikované podoby, ale dle původního kontrolního součtu může být i ověřeno, zda je vyléčená forma souboru 100% shodná s původní.

3.2.3.3 Sebeléčení

Jde o velice ojedinělou vlastnost malé skupiny virů a jim podobných. Konkrétním příkladem může být starý polský virus Pieck.4444.A (Kaczor.4444.A), který sám sebe odstraní s počítače, jakmile uživatel při startu PC vypíše dané slovo. Konkrétně v tomto případě to na dnešních rychlých počítačích není možné, jelikož chvíle, během které je slovo očekáváno je až příliš krátká.

4 Srovnávací testy antivirových skenerů

Vzájemné srovnávání antivirových skenerů provází antiviry od samého počátku. Srovnávací testy se často zaměřují na samotnou výkonnost skenerů (on-demand i on-access) a nejsou tak porovnávány ostatní vlastnosti produktů (jednoduchost používání, služby atd.).

Pro srovnávací testy skenerů jsou rozhodující dva faktory, kvalita detekce a rychlost. Kvalitou rozumíme procentuální úspěšnost detekce, co do počtu zachycených škodlivých kódů v poměru ku celkovému počtu škodlivých kódů. Skenery by měly být testovány i na rozsáhlé sbírce zcela nezávadných souborů. Kvalitní skener totiž musí nejen spolehlivě detekovat malware, ale zároveň i minimalizovat množství falešných poplachů a to vše stihnout v rozumně krátkém čase.

Po stránce rozsáhlosti bývají testy prováděny odděleně na sbírce In The Wild virů a tzv. „zoo" virů. V prvním případě jsou do testu zařazeny reálně se šířící viry a to ty, které jsou prezentovány v pravidelně vydávaném seznamu „PC Viruses In The Wild" (viz. první část publikace). Ve druhém jde o obrovskou směs škodlivých kódů, z nichž celá řada neznamená vůbec žádnou reálnou hrozbu.

Úspěšnost detekce ITW virů je tak nutno hodnotit velice přísně a tak každou větší odchylku od 100% hodnotit velice negativně. Úspěšnost detekce „zoo" sbírky lze pochopitelně hodnotit volněji.

Sbírka malwaru musí být udržována, soubory infikované konkrétním virem je vhodné držet v několika exemplářích a to především u polymorfních virů (kde mohou nastat komplikace při detekci). Sbírka musí obsahovat jen regulérně namnožené viry, žádné germy apod. Majitel takové sbírky si musí být zároveň 100% jist počtem infikovaných souborů. Z výše uvedeného je patrné, že podobně kvalitní sbírky mohou vlastnit jen antivirové společnosti a několik málo specializovaných organizací. Výsledkům testů antivirových skenerů, které se jednou za čas objevují v některých časopisech a nepocházejí od těchto organizací, tak nelze vůbec věřit.

Každý srovnávací test vyžaduje časový harmonogram. Jde především o rozeslání „pozvánky" antivirovým společnostem, stanovení datumu, do něhož mohou antivirové společnosti zasílat aktualizace pro své produkty, ale i datum, kdy dojde ke zmražení sbírky. Tím rozumíme moment, počínaje kterým již nebudou ve sbírce prováděny žádné úpravy a doplňovány žádné další viry. Pro dosažení reálné šance detekce 100% ITW virů je potřeba zmražení provést dříve, než dojde k zákazu přijímání aktualizací antivirových skenerů.

4.1 Virus Bulletin (www.virusbtn.com)

Nejuznávanější srovnávací testy pravděpodobně publikuje anglický časopis Virus Bulletin (www.virusbtn.com). Dochází k nim obvykle každé dva měsíce a použitá cílová platforma je neustále měněna. Díky tomu lze v průběhu roku narazit na testy pod Windows NT, Linux, Windows XP, Novell atd. Antivirovým systémům, kterým se podaří detekovat 100% ITW virů v obou kategoriích (on-access i on-demand) a zároveň projít na „Clean Files Test Set“[31] bez falešného poplachu je uděleno ocenění „Virus Bulletin 100% Award”. Celková rychlost ani úspěšnost v zoo testech (rozděleno do kategorií makroviry, souborové viry, polymorfní viry) nerozhoduje, ale případné úspěchy v těchto oblastech jsou vyzdviženy v krátkých článcích věnovaných každému antiviru.

Ocenění je bráno velice prestižně a v době psaní této publikace vévodil celému pelotonu slovenský antivirový systém NOD32 v počtu ocenění VB100% Award.

Najít jde bohužel i několik stinných stránek. Redaktoři nedovolují publikaci kompletních výsledků jinde, než v samotném časopisu[32]. Vzhledem k tomu, že jde o časopis zcela bez reklam, jeho cena je až přehnaně vysoká – roční předplatné (12 čísel) vychází v přepočtu přes 10 000 Kč ! Běžný smrtelník se tak dozví pouze několik málo informací přímo ze stránek www.virusbtn.com, kde je naneštěstí uvedeno pouze jméno antiviru a jeho verdikt ve stylu USPĚL / NEUSPĚL.

4.2 GEGA IT-Solutions (www.av-test.org)

Srovnávací testy této organizace bývají využívány řadou časopisů (především německých) a velké množství jich je dostupných přímo na stránkách www.av-test.org.

Narozdíl od testů Virus Bulletinu bývají srovnávány i další vlastnosti antivirových systémů, které jsou bližší běžnému uživateli (možnosti nastavení, způsob aktualizace, podpora archivačních programů, délka produktu…).

4.3 Universita Hamburg

Srovnávací testy z tohoto zdroje bývají často zatracovány přímo antivirovými společnostmi a to díky velice špatné komunikaci s vedením. Navíc řada lidí, podílejících se na „fyzickém“ vyhodnocování testování této problematice nerozumí.

(http://agn-www.informatik.uni-hamburg.de/vtc/naveng.htm)

5 Konkrétní antivirové společnosti

Následuje přehled nejznámějších antivirových společností z pohledu obyvatele České Republiky.

5.1 Alwil Software (avast!)

[31] Velká sbírka neinfikovaných souborů pro odhalování případných falešných poplachů.

[32] Jako autorovi této práce a stránek www.viry.cz bylo dokonce doporučeno ze strany šéf-redaktorky časopisu Virus Bulletin, kompletní výsledky okamžitě z www.viry.cz stáhnout.

Pravděpodobně nejstarší tuzemský výrobce antivirových produktů. avast! antivirus způsobil před lety rozruch, když hned při vstupu do srovnávacích testů Virus Bulletin dokázal porazit řadu antivirů zvučných jmen a nedlouho poté jako první v historii detekoval 100% virů ve všech kategoriích. V dnešní době je jednou z mála společností, která nabízí velice kvalitní řešení pro domácnosti zdarma (avast! 4 home edition).

5.2 Grisoft (AVG)

Druhý z tuzemských výrobců, tentokrát produktu AVG. Pojem AVG se stal zřejmě navždy legendou v ČR i na Slovensku a AVG tak jednoznačně vládne v používanosti.

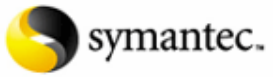
5.3 ESET (NOD32)

Antivirový systém NOD32 slovenské společnosti ESET se dostal do podvědomí taktéž až se vstupem do srovnávacích testů časopisu Virus Bulletin. Drží rekordní počet ocenění Virus Bulletin 100% Award a nechává za sebou i taková zvučná jména jako Symantec, McAfee či F-Secure. Skenovací „motor" je po technologické stránce tím nejlepším, ojedinělé binární prohledávání umožňuje v průměru až několikrát rychlejší skenování v porovnání s konkurencí.

5.4 McAfee VirusScan (NAI)

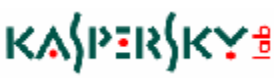
Zřejmě nejlegendárnější antivirový systém. V době MSDOSu bylo možné najít tento antivirus na velké spoustě PC v nepředstavitelně velkém množství verzí. Příchod polymorfního viru One_Half v roce 1994 odhalil velkou slabinu tohoto systému a tím byla právě detekce polymorfních virů. Tento a další problémy, včetně mírné technické zaostalosti za konkurencí, vyřešilo až odkoupení tehdy nejuznávanějšího antivirového systému Dr. Solomon AVTK. Již při letmém pohledu na čtvrtou generaci bylo zřejmé, že z původního McAfee VirusScanu zůstal na místě leda tak název a vše ostatní skrytě zařizovaly technologie převzaty z Dr. Solomona.

5.5 Symantec (Norton Antivirus)



Softwarový gigant, jehož produkty Norton Commander či Norton Utilities zná snad každý pokročilejší uživatel. Mezi obrovskou škálou produktů najdeme i Norton Antivirus. Současné verze nabízejí nejkomplexnější řešení jak pro jednotlivce tak i společnosti. Během své existence pohltil IBM Antivirus.

5.6 Kaspersky Lab (Kaspersky Antivirus)



Ruská společnost se sídlem v Moskvě. Zakladatelem je Eugen Kaspersky, jeden z nejuznávanějších virových odborníků. Během dlouhé historie měnil antivirus dvakrát svoje jméno. Původní název Doktor Kaspersky byl ještě za dob nízké popularity změněn na AntiViral Toolkit Pro (AVP), se kterým prorazila společnost do světa. Mezi vedením a distributory došlo později zřejmě k velice ostrým sporům, takže řada z nich odešla ke konkurenci. Největší strategickou ztrátou byl distributor pro USA, majitel domény www.avp.com. Využití této domény pro prodej konkurenčního antiviru rumunské společnosti Softwin, byl dokonalou pomstou,

obzvláště když byl narychlo přejmenován na AVX (silně připomínající původní AVP). Celá tato krize vyvrcholila až druhým přejmenováním, tentokrát na Kaspesky Antivirus (KAV). Kaspersky Lab nabízí v dnešní době velice širokou škálu řešení, ale bohužel vcelku často působí některé z nich nedodělaným dojmem.

5.7 RAV Anti-Virus (GeCAD)

Byl to antivirus s nejpropracovanější antivirovou ochranou platformy Linux a zároveň antivirem lidově řečeno „za hubičku“[33]. V roce 2003 byla rumunská společnost GeCAD pohlcena Microsoftem a došlo tak k situaci lidově popsané větou „dvě mouchy jednou ranou“. Microsoft získal velice kvalitní odborníky přes bezpečnost a zároveň zničil nejkomplexnější antivirové řešení pro Linux. Distributoři a samotní uživatelé RAV antiviru se o této skutečnosti dozvěděli až z tiskové zprávy, předem nebyli o ničem informováni.

6 Praxe

Co se týče praxe v používání antivirového software, pak je zřejmě nejzajímavější proces odstraňování virů. U starých virů pro DOS na většinu antivirů čekala nástraha v podobě paměťově rezidentních virů, které mohly samotný proces léčení ovlivnit (a taky se to často stávalo). Tehdy bylo nutno použít tzv. systémovou disketu, ze které byl zaveden operační systém namísto z infikovaného pevného disku. Tím se zajistilo čisté prostředí, ze kterého bylo antivirový program bezpečné spustit a viry odstranit. Tuto metodu bylo možno úspěšně aplikovat i u virů pro Windows 9x. Jistou alternativou bylo v tomto případě restartování do systému MSDOS prostřednictvím nabídky START / Vypnout, popřípadě zvolení adekvátní volby po stisku klávesy F8 během zavádění Windows.

První komplikace nastaly s příchodem Windows Me, které se pyšnily totálním odchodem od OS DOS. Ve skutečnosti šlo opět o nástupce řady 9x s uměle blokovaným DOSem. Bylo tedy možné použít už jen první variantu – systémovou disketu.

6.1 NTFS

Daleko větší problém představují souborové systémy NTFS, které dokážou využívat operační systémy na bázi Windows NT (nejčastěji 2000, XP, 2003). Souborový systém NTFS totiž není z běžné systémové diskety viditelný (narozdíl od FAT, FAT32). Tuto překážku lze obejít například následovně:

6.1.1 Speciální ovladače

Existují speciální ovladače (www.wininternals.com), které sice zaberou dosti operační paměti (to by nebylo problémem, nebýt oné magické hranice 640 KB) a NTFS partition pevného disku zpřístupní.

6.1.2 Přesun disku do jiného PC

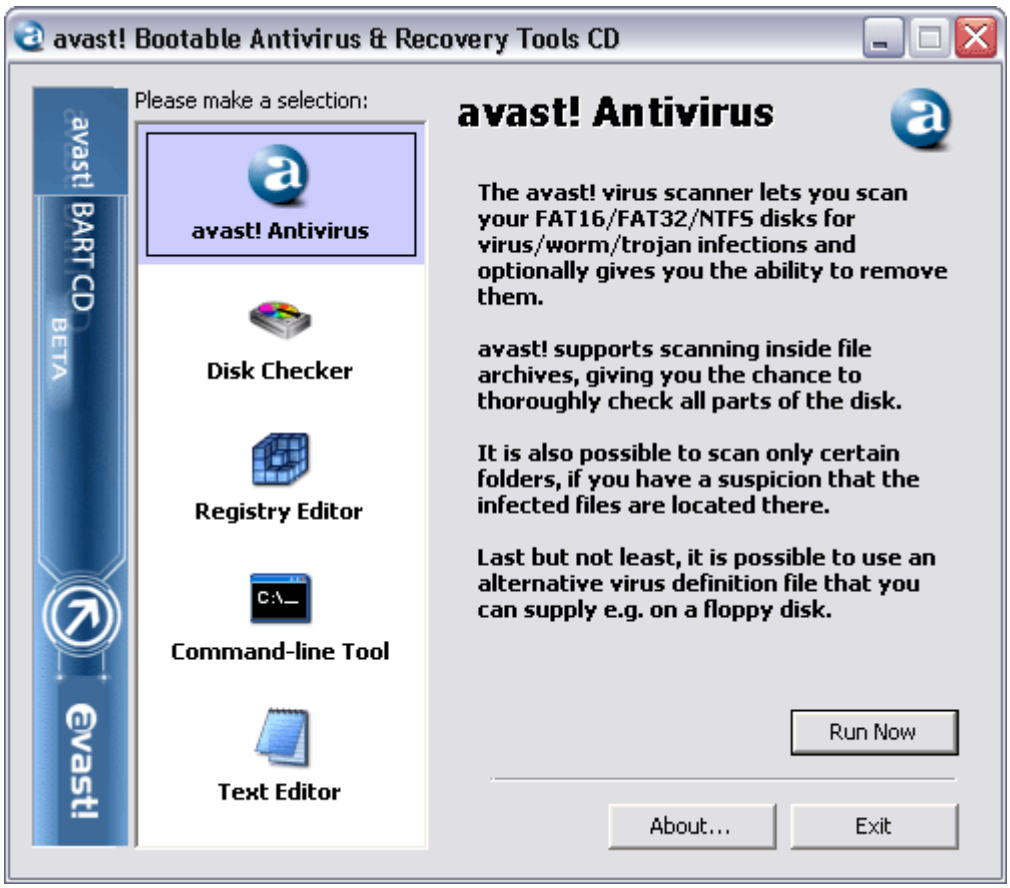
Přesunem disku s NTFS systémem do jiného PC s Windows NT nebo Linuxem je další, méně pohodlnou variantou.

[33] Především díky modelu licencování dle počtu domén, nikoliv dle počtu mailboxů – poštovních schránek.

6.1.3 Záchranné systémy

Jedním z prvních byla „záchranná disketa“ společnosti Kaspersky Lab. V antiviru AntiViral Toolkit Pro 3.5 (AVP 3.5) se objevila aplikace pro tvorbu záchranných disket na bázi Linuxu, ze kterých bylo možno zavést operační systém Linux (systémové diskety) ze kterého byly jednak viditelné NTFS partition, ale bylo možno odtamtud pustit i AVP antivirus pro Linux.

Zřejmě nejlepší řešení nabízí tuzemský výrobce antiviru avast! – Alwil Software. Ve spolupráci s Microsoftem vyvinuly speciální bootovatelné CD s operačním systémem Windows PE – avast! BART CD. Vzhledově se to velice podobá operačnímu systému Windows XP, ale s tím rozdílem, že Windows PE vůbec ke své činnosti nepotřebuje pevný disk ! Kromě toho, že můžeme přistupovat k diskům NTFS, je k dispozici pohodlné „windousojdí“ prostředí pochopitelně s avast! antivirem. Mezi další interní software pak patří například Servant Salamander a jednoduchý textový editor. Speciální pozornost si ale bezpochyby zaslouží editor registrů, který se dokáže vzdáleně napojit k registrům na inkriminovaném pevném disku, ale především podpora řady RAID řadičů a sítě ! To vše bez zápisu na pevný disk, jen médium CD a RAM.



Obrázek 32 avast! BART CD, hlavní menu

6.2 Obnova systému (restore)

Operační systém Windows ME (+ Windows XP a výše) uvedl novinku v podobě funkce „Obnova systému“ (v anglické verzi „Restore“). Jde o funkci, která umožňuje v případě nestability OS návrat ke stavu, kdy bylo všechno v pořádku. K tomuto účelu je vytvořen adresář _RESTORE (u Windows XP – System Information Volume), do kterého si Windows průběžně ukládá veškeré provedené změny (například instalací nového

softwaru) a to v takové formě, aby byl na žádost uživatele schopen vrátit stav Windows například o 14 dní zpět. Nápad je to skvělý, avšak velice často se stává, že se do tohoto složitého procesu zapletou i případné infikované soubory, které se v době vytváření "zálohy stávajícího stavu" pohybovaly na disku. Jmenované adresáře jsou pečlivě chráněny a nelze z nich nic běžnými prostředky odmazat. Tedy ani případné infikované soubory, které tam jsou v „pasti" a zcela neškodné (do doby, než dojde k případnému obnovení stavu, kdy bylo všechno „v pořádku" a zároveň šlo o stav infikovaný...), na které občas upozorní rezidentní štít antivirového systému.

Řešením je využití jedné z výše uvedených metod a ručního odmazání takto postižených souborů, popřípadě vypnutí funkce „Obnova systému" / „Restore":

Klikněte pravým tlačítkem myši na ikonu TENTO POČÍTAČ (MY COMPUTER) a zvolte z nabídky VLASTNOSTI (PROPERTIES). Přepněte se do záložky VÝKON (PERFORMANCE) a stiskněte tlačítko SOUBOROVÝ SYSTÉM (FILE SYSTEM). Zde se přesuňte na záložku PŘI POTÍŽÍCH (TROUBLESHOOTING) a zaškrtněte poslední volbu - ZAKÁZAT OBNOVU SYSTÉMU (DISABLE SYSTEM RESTORE). Vše potvrďte tlačítkem OK, PC se restartuje.

Pod Windows XP je vypnutí této funkce snažší, stačí kliknout pravým tlačítkem myši na ikonu TENTO POČÍTAČ (MY COMPUTER), zvolit VLASTNOSTI (PROPERTIES) a nalistovat záložku OBNOVENÍ SYSTÉMU (SYSTEM RESTORE). Tam už se nachází jediné zaškrtávací políčko pro vypnutí.

Prevence jiná, než softwarově-antivirová

Antivirový software není jediným prostředkem prevence před infiltrací. Zároveň nemusí jít o dostačující prostředek.

O prevenci v podobě informovanosti a školení uživatelů PC pravděpodobně nemusíme detailněji rozebírat. Smutnou skutečností ovšem zůstává, že informovanost je velice nízká i když zdrojů odkud čerpat lze najít požehnaně. Jistým důkazem může být průměrná návštěvnost stránek viry.cz v porovnání s tou, která následuje při významném virovém incidentu (rekordem je 13x vyšší návštěvnost – souviselo s virem Win32/BugBear.B). Lidé tak ve většině případů přicházejí na stránky až v době po infekci PC a hledají poslední naději v podobě jednoúčelových antivirů.

1 Formy prevence

1.1 Inteligence

Inteligenci lze uplatnit především u podezřelých příchozích e-mailů. Koho by pak napadlo spustit přílohu e-mailu, který je napsán anglicky a adresa odesílatele je viděna prvně...

1.2 Informovanost

Sledování webových stránek antivirové společnosti, jehož produkt vlastníme by mělo být na denním pořádku, stejně jako sledování nezávislých stránek v podobě www.viry.cz, www.hoax.cz či www.virusy.sk.

V případě vlastních stránek viry.cz doporučuji objednat službu, která zadarmo rozesílá novinky přímo do e-mailové schránky.

Obrázek 33 Část úvodní strany www.viry.cz

1.3 Aktuální verze softwaru

Případnou infekci lze významně omezit pravidelným stahováním aktuálních verzí softwaru (stahování aktualizací pro antivirové systémy bylo rozebráno výše) a to bez ohledu na to, o jakou platformu jde. Nás zajímá především operační systém Microsoft Windows, kde kromě aktualizací pro samotný systém existují i velice důležité aktualizace pro aplikaci Internet Explorer. Šíření řady virů díky bezpečnostním chybám v aplikaci Internet Explorer by bylo možno ihned omezit v případě, že by takovou pravidelnou aktualizaci prováděl každý uživatel. Realita je bohužel smutnější.

Nejjednodušší formou stahování aktualizací je stránka:

http://windowsupdate.microsoft.com

Průvodce již sám provede kontrolu aktuální verze operačního systému na uživatelově PC a navrhne vhodné aktualizace a záplaty ke stažení.

Alternativou jsou samotné stránky www.microsoft.com odkud je jednotlivé aktualizace – záplaty možno stahovat manuálně ve formě hotfixů (řeší jeden konkrétní bezpečnostní problém), kumulativních záplat (řeší všechny dosud objevené bezpečnostní problémy pro URČITOU VERZI softwaru) a „service packů" (obsahují jak vylepšení daného produktu, tak i celou škálu záplat).

1.4 Nastavení softwaru

Omezit průnik lze i vhodným nastavením používaného softwaru.

1.4.1 Internet Explorer

Vypnutím technologie ActiveX (Nástroje / Možnosti / Zabezpečení / Vlastní úroveň) lze značně omezit průnik škodlivých kódů a to především dialerů. Úroveň zabezpečení by přitom neměla klesnout níže než na Střední.

1.4.2 Outlook & Outlook Express

Zde došlo v posledních verzích k rapidnímu zlepšení situace, takže paradoxně poskytnu návod, jak se těchto přehnaných opatřeních zbavit. V případě Outlook Expressu jsou standardně blokovány přílohy v příchozích e-mailech, nelze je ani uložit na disk. Tuto vlastnost lze zrušit v Nástroje / Možnosti / Zabezpečení. Podobná situace je i v případě plného Outlooku z kancelářského balíku Microsoft Office, jsou zakázány některé přípony příloh a k nápravě může dojít pouze zásahem do registrů. Naštěstí existuje několik speciálních programů (stačí vyhledat přes google.com).

1.4.3 MS Office

Počínaje verzí Microsoft Office 97 jsou jednotlivé produkty tohoto kancelářského balíků vybaveny zabezpečením před aktivací maker. Je dobré se v Nástroje / Makra ujistit, že je toto zabezpečení aktivováno.

1.4.4 Poštovní servery

Velice účinným řešením je filtrace nebezpečných přípon příloh. Skriptové VBS viry lze zcela spolehlivě a bez vedlejších účinků „odstřihnout" filtrací přípony VBS, jelikož ji běžně nikdo nepoužívá. S filtrací přípon EXE, PIF, SCR je to velice podobné. Pokud to poštovní server umožňuje, lze bez vedlejších účinků filtrovat i některé MIME typy a dvojité přípony.

1.4.5 Firewally

Doporučeno je na základě routovacích pravidel povolit směrem ven i dovnitř pouze takové pakety - služby, které jsou nezbytně nutné. Otevřené porty ve směru dovnitř je vhodné povolit jen z určitých IP adres či rozsahů (například pouze z IP adresy pobočky v případě, že se připojuje k centrále).

Dobrým nastavením se bylo zcela reálné vyhnout červu SQLSlammer, který útočil prostřednictvím otevřených portů 1433-34 (Microsoft SQL Server).

Virová scéna

Osobně bych si tak dovolil lidi z virové scény („vxers“) rozdělit do několika skupin.

1 „Vxers“

1.1 Fanatici

Obvykle jde o lidi s cílem uškodit. Troufám si říci, že naprostá většina nejrozšířenějších virů pochází právě od této skupiny lidí. Snaží se tedy vytvářet takové viry (popřípadě upravovat existující), které mají reálnou šanci na úspěch. V dnešní době tak především viry šířící se elektronickou poštou. Většina lidí této skupiny zůstane veřejnosti navždy utajena a to i z jednoho prostého důvodu – zdrojový kód není dotyčnou osobou podepsán přezdívkou.

1.2 Umělci

Druhou skupinou jsou „umělci“, kteří vytvářejí svá virová dílka pro radost, ale nikoliv z případného úspěchu po stránce celosvětového rozšíření nebo z napáchaných škod. Programují viry jako koníček. Svá díla umisťují společně se zdrojovým kódem na vlastní webové stránky a v řadě případů je i sami rozešlou některým antivirovým společnostem. Sami o sobě viry nešíří a pod své dílo se podepisují přezdívkou. V řadě případů jde o „technické skvosty“, viz. v průběhu publikace popsané viry českých autorů Benny/29A, Ratter/29A, či autora z ruska - Z0mbie. V řadě případů nejde o reálné hrozby, protože udržet taková rozsáhlá díla bez chyby je velice obtížné.

1.3 Sběrači

Vytvářejí vlastní sbírku virů stahováním všemožných infikovaných souborů z Internetu. Následně na této sbírce vytvářejí reporty antivirových skenerů, které publikují na vlastních webových stránkách a s ostatními organizují vzájemné výměny vzácných chybějících virů. K případným srovnávacím testům antivirových skenerů jsou pochopitelně takové sbírky zcela nevyhovující.

1.3.1 Virus Collectors

Speciální software, jehož úkolem je setřídit nově vznikající virovou sbírku do přijatelně přehledné podoby. Příkladem může být software, který na základě reportů vybraných antivirových skenerů vytvoří názvy adresářů totožné s názvy virů a do nich pak příslušné infikované soubory roztřídí.

2 Skupiny – Groups

Někteří „vxers“ zůstávají samostatně, jiní se sdružují do skupin, v rámci níž členové spolupracují. Zřejmě nejznámější a nejuznávanější skupinou je původem španělská 29A, která má v době psaní tohoto článku následující členy:

Benny, GriYo, Mental Driller, Ratter, roy g biv, Super, Vecna, VirusBuster, Z0mbie

2.1 VX-meetingy

Jednou za čas jsou pořádány různé několikadenní společné akce – meetingy. Osobně jsem byl účastníkem jedné z nich, která se konala v roce 2000 v Brně.

Obrázek 34 Jeden z výsledků VX meetingu v Brně 2000

Seznam všech účastníků je k vidění na výše uvedeném letáku, kterým bylo dokonale oblepeno okolí vchodu do budovy, ve které sídlí společnost Grisoft software, výrobce antivirového systému AVG.

2.2 eZiny

eZiny jsou elektronickou podobou magazínů - často ve formátě HTML). Řada skupin vydává vlastní eZin, který je složen z článků členů a obsahuje jak různé návody (tutorials) k tvorbě virů, tak i zdrojové kódy virů, které byly danou skupinou vyprodukovány od posledního vydání eZinu. Populární jsou i rozhovory a úvodníky se vzkazy k antivirovým odborníkům.

Seznam použité literatury

- **MICHAL DANILÁK, *Svet počítačových vírusov*, Grada, 1992**
- **PETR ODEHNAL, PETR ZAHRADNÍČEK, Praktická *sebeobrana proti virům*, Grada, 1996**
- **PÉTER SZÖR, *Attack on Win32, Data Fellows*, 1998**
- **PÉTER SZÖR, *Attack on Win32 – Part II, Symantec (SARC Division)*, 2000**
- **PÉTER SZÖR, PETER FERRIE, *Hunting for metamorphic*, Symantec Corp., 2001**
- **NÁDENÍČEK PETR, TOMÁŠ PŘIBYL, PŘIKRYLOVÁ OLGA, VOBRUBA TOMÁŠ, *Chip Special*, Vogel Publishing**
- **PAVEL BAUDIŠ, *Makroviry*, Alwil Software**
- **JOSEF DŽUBÁK, *www.hoax.cz***

- **REPTILE & VICODINES, *Cross infection I-II*, 29A Issue #3**
- **BENNY/29A, *EPO techniques under Win32*, 29A Issue #4**
- **BENNY/29A, *InterProcess Communication*, 29A Issue #4**
- **BENNY/29A, *Threads and fibers under Win32*, 29A Issue #4**
- **BENNY/29A & RATTER, *Win2k.Stream*, 29A Issue #5**
- **BENNY/29A & RATTER/29A, *Win2k.SFPDisable*, 29A Issue #6**
- **BENNY/29A, *Win32/Linux.Winux*, 29A Issue #6**
- **ENDER, *TMC:Level6x9.A*, Asterix #1 (*zine #1)**
- **MGL, *Stealth*, Asterix #1 (*zine #1)**
- **FLUSH, *A brief history of virii vs. antivirii war*, Asterix #2 (*zine #2)**
- **FLUSH, *Main target: heuristics*, Asterix #2 (*zine #2)**