TÓTH J. SZABOLCS

PC VÍRUSOK

LSI OKTATÓKÖZPONT

Tóth J. Szabolcs

PC VÍRUSOK

Nyitott rendszerű képzés – Távoktatás –
Oktatási segédlete
Felsőoktatási Tankönyv

LSI Informatikai Oktatóközpont
A Mikroelektronika Alkalmazásának
Kultúrájáért Alapítvány
Budapest, 2001

Lektorálta: Dr. Kovács Magda PhD
főiskolai tanár
Dr. Hosszú Gábor
a műszaki tudomány kandidátusa, egyetemi docens
Mátrai Mátyás Béláné

A könyv megrendelhető vagy megvásárolható.

LSI Oktatóközpont
1037 Budapest, Bécsi út 324.
Tel./fax: 250-6013

ISBN 963 577 259 9

Kiadó: **LSI Oktatóközpont**
Felelős vezető: **Dr. Kovács Magda**
Témafelelős: **Flier István**

Szedés:
OFFICE-CAR Bt.
2040 Budaörs, Csata u. 17.

TARTALOMJEGYZÉK

A JELÖLÉSEK MAGYARÁZATA

! – Fontos kérdésekre hívja fel a figyelmet.

 – Súlyponti rész, figyelmesen kell olvasni.

 – Magyarázat, ötlet.

 – Számítógép-használatot igénylő „élő" illusztráció, megoldandó feladat. (Szimuláción alapuló vizsgálat vagy feladat.)

 – Kiegészítő ismeretek.

? – Ellenőrző kérdés(ek).

(1)2.3 – Irodalmi hivatkozás: [Sorszám az irodalomjegyzékben] (al)fejezet azonosító.

 – Aprólékos, figyelmes áttanulmányozást, önálló munkát igénylő ismeretanyag, fontos összefüggéseket tartalmazó logikai vázlat.

§ – A kifejtettekre vonatkozó törvény, jogszabály.

 – Felkeresni ajánlott Internet cím, WEB lap

1. ELŐSZÓ

Mindössze pár hónap telt el azóta, hogy könyvem első kiadásához írt kiegészítés megjelent, azonban úgy éreztem, ezt a két művemet össze kell dolgoznom, hogy átláthatóbb képet kapjon mindenki a PC vírusok fajtáiról, a vírusvédelmi módszerekről és rendszerekről. Éppen ezért nem kell meglepődni, ha az első kiadás (és a kiegészítés) különböző fejezeteinek egy-egy részét vagy teljes egészét ebben a könyvben is megtalálja a Kedves Olvasó, hiszen az első kiadás óta a bináris vírusokkal kapcsolatban semmi lényeges változás nem történt, annál több viszont a makróvírusok területén.

Ezen nem szabad csodálkozni, hiszen hatalmas léptékű fejlődés történt a PC-k hardver és szoftver rendszerében is: ma a kereskedelemben kapható legkisebb teljesítményű gép 3 évvel ezelőtt szinte még nem is létezett, azóta új, grafikus operációs rendszerek és a felhasználó igényeit jobban kiszolgáló alkalmazások jelentek meg. Ez a fejlődés a PC vírusok fejlődési irányát is befolyásolta, hiszen alkotóiknak - ha továbbra is szeretett volna magáról hallatni - alkalmazkodniuk kellett a megváltozott számítógépes környezethez. Ennek eredményeképpen jelentek meg a különböző, makrónyelvi lehetőséggel ellátott alkalmazásokhoz kifejlesztett makróvírusok. Ráadásul ezen vírusok száma jóval dinamikusabban növekszik, mint a régebbi, bináris vírusok száma. Igaz ma is megjelennek pl. Windows 95 alá kifejlesztett filevírusok, de nem ez a jellemző.

Hasonlóan az első kiadáshoz ebben a bővített kiadásban is kizárólag PC kompatibilis gépek vírusaival foglalkozom, újra áttekintést nyerhetünk az egyre inkább bővülő PC vírustípusokról és működési elveikről, a különböző vírusvédelmi szabályokról, néhány antivírus programról illetve vírusgyűjtemények létrehozásáról és karbantartásáról.

Könyvemnek nem szándéka, hogy azt átolvasva bárki könnyedén megírjon egy vírust, ellenben célom, hogy mindenki más, aki nem ilyen programok írogatásában éli ki programozói fantáziáját, megértse a vírusokkal kapcsolatos fogalmakat, igaz ehhez szükséges egy kis számítógép felhasználói tudás is.

Köszönetemet szeretném kifejezni mindenkinek, aki hozzájárult tudásom gyarapodásához, így könyvem megírásához is. Külön köszönöm a könyveimmel kapcsolatos kritikákat és elnézést kell kérjek a kisebb tévedésekért.

Bár többen is kifogásolták, de az aktuális antivírus programok magyar nyelvű leírását továbbra sem tartom jó ötletnek beletenni egy könyvbe, hiszen mire a könyv megjelenik, addigra az antivírus programokat részletesen ismertető fejezet már nem aktuális.

Bízom abban, hogy az egyes fejezetekben mindenki megtalálja a számára érdekes és hasznos információkat.

Budapest, 1998. december

Tóth J. Szabolcs
E-mail: tszabi@freemail.c3.hu

2. PC VÍRUS TIPOLÓGIA

Mielőtt részletesen belekezdenénk a tipológiába, tekintsünk egy picit vissza, hogy honnan indult az egész. Sokan úgy képzelik, hogy a vírusok és a velük kapcsolatos programok a 80-as és a 90-es évek számítástechnikájának átkai. Pedig ez nem így van: már a 60-as években, amikor tömegesen megjelentek a számítógépek, egyes katonai körök már foglalkoztak vírusok és trójai programok megírásával. Az akkori "ellenség" gépeibe próbálták ezeket a programokat eljuttatni. A cél minden esetben rongálás vagy bizonyos információk megszerzése volt.

"Hivatalosan" csak jóval később, az 70-es évek közepén kezdtek ilyen jellegű programok kutatásába, többek között a Rank Xerox laboratóriumaiban. Ne felejtsük el, hogy ekkor a PC-k még nem jelentek meg és ezekben a kísérletekben Unix alapú rendszerek lehetőségeit kihasználó un. férgekkel foglalkoztak. Már most szeretném megjegyezni, hogy a számítógépvírusok és a programférgek – bár rokoni kapcsolatban állnak egymással – mégis eltérően szaporítják önmagukat. A vírushoz mindig kell egy hordozó program, amihez képes hozzákapcsolódni és így továbbszaporodni, míg a féreg egy komplett program.

Ezekről a "férges" kísérletekről számítástechnikai újságokban különböző publikációk jelentek meg, aminek eredményeképpen egyes amerikai egyetemeken a számítógép-biztonsággal foglalkozó tantárgy keretein belül ismertetésre kerültek a kísérleti eredmények.

Egy ilyen szemináriumon 1983. november 3.-án, a PhD hallgató Fred Cohen-nek egy "kitűnő" ötlete támadt, amelyet kísérlettel akart bebizonyítani. Az ötlet nagyon egyszerű: tervezni kell egy olyan programot, amely képes más programokat módosítani olyan módon, hogy a módosított programok tartalmazzák a módosító program egy másolatát. Ez a másolat is futóképes kell hogy legyen, vagyis ugyanúgy képesnek kell lennie más programokat módosítani és újabb másolatait beleírni további programokba, mint az eredeti. Gyakorlatilag ezzel Cohen-t tekinthetjük a vírusok atyjának, hiszen az összes ma létező vírus ezen az elven működik.

A kísérletet egy VAX 11/750-es számítógépen végezték el. Az a program, amibe beletették ennek a módosítgató programnak az első példányát, egy segédprogram volt. Természetesen a felhasználók semmit sem tudtak a kísérletről. Az eredmény döbbenetes volt: minden felhasználónak a programjai megfertőződtek, még azok sem tudtak ellene tökéletesen védekezni, akik tudtak a kísérletről. Persze a vírusprogramba nem volt beépítve semmiféle romboló rutin, sőt úgy volt megírva, hogy könnyen felfedezhető legyen.

A kísérlet befejeztével minden fertőzött file-t töröltek és a rendszeradminisztrátor sem engedett meg több kísérletezést. Azt hiszem, ez elég érthető.

A nemcsak számítástechnikával foglalkozó nagyközönség ezzel a gonosz technikával csak később, a PC-k széleskörű elterjedése után ismerkedhetett meg. Az első PC vírus megjelenését 1987-re tehetjük, ekkor szabadult ki egy számítógépes laborból a Pakistani Brain. Nem sokkal ezután további PC vírusok (Vienna, Lehigh, Stoned, Jerusalem) tűntek fel, nem kis riadalmat okozva a felhasználók között. Erre válaszul sorra jelentek meg a különböző antivírus cégek.

A PC-k elterjedésével egy új szó került a számítástechnikai szótárakba: a szoftverlopás. Ekkor a nagyobb szoftvergyártó cégek olyan eljárásokat dolgoztak ki, melyek megakadályozták a programok törvénytelen másolását: büntető algoritmust vagy egy komplett vírust szerkesztettek a programba és ha valaki megpróbálta illegálisan lemásolni a szoftvert, akkor a másolásvédelem megsemmisítette a felhasználó programjait, adatait vagy kiszabadította magából a vírust. De nemcsak a másolásvédelemből kiszabaduló vírusok okozta botrányok miatt hagyták abba ezt a fejlesztési irányt az egyes cégek, hanem azért is, mert a fiatalabb nemzedék megalakította saját crackercsoportjait, akik ezeknek a másolásvédelmeknek a feltörésével foglalkoztak és foglalkoznak.

A 90-es évek elejéig csak a vírusok száma növekedett erőteljesen, majd ekkor megjelentek először a lopakodó, majd nem sokkal később a polimorf, az önmaguk kódját folyamatosan változtató vírusok.

Már 1993-tól kezdődően foglalkoztak vírusíró csapatok ún. makróvírusok megírásával, egy amerikai egyetemen a makróvírusokkal kapcsolatos kísérleteket csak 1994-től kezdték el. Nem sokkal ezután egyre nagyobb számban kezdtek megjelenni a Word makróvírusok, hiszen ezáltal az egyes vírusfejlesztő csapatok elérték azt, amit a bináris vírusokkal nem voltak képesek: multiplatformmá tenni a megírt vírust, hiszen pl. a Word-nek több operációs rendszer alá írott változata is létezik és a változatok közötti eltérés minimális. A makróvírusok számának növekedését az is elősegítette, hogy 1995-ben megjelent a Windows 95 operációs rendszer, ahol egészen másképp kellett lekezelni egy vírushordozó programot, jóval egyszerűbb volt makróvírust írni a megjelenő Office 95 alá, amelyben a Word makrónyelve teljesen megegyezett az előző verzióéval.

Nem csodálkoznék rajta, ha a súlypont újra visszahelyeződne a Windows 95 és a Windows 98 alá írt file-vírusokra.

Sokak számára nem tisztázott jó néhány fogalom a számítástechnika vírusokkal kapcsolatos részéből. Éppen ezért fontosnak tartom, hogy mindenkinek elmagyarázzam az önreprodukáló és kártékony programok működési elvét, azok típusait.

2.1. VÍRUSPROGRAMOK

A számítógépvírus az önreprodukáló programok legnépesebb családja. Valójában az élő anyag működőképes modellje, a számítógépben a programok, adatok felelnek meg a valóságos életbeli egészséges szervezetnek, amit megtámadhat a vírus.

A számítógépvírusok olyan programok, amelyek képesek önmagukat másolni mindenféle operátori ellenőrzés nélkül, adathálózatokon vagy floppy-kon átkerülhetnek más számítógép(ek)re, ahonnan szintén tovább terjedhetnek más számítógép(ek)re. A számítógépvírus sosem önmagában indul el, mindig kell egy olyan program (boot-szektor is az!) vagy egy makrókat is tartalmazható adatállomány, ami hordozza. Tehát a vírusok szaporodnak – hasonlóan a programférgekhez – míg más kártékony programok nem.

Nagyon sokan úgy képzelik, hogy a vírus valamilyen programhibánál vagy pedig valamely számítógép-alkatrész meghibásodásakor keletkezik. Egy vírus nem csak úgy létrejön a semmiből, hanem etikátlan programozók írják, legtöbbször tudásuk fitogtatásának céljából.

Vírusírók, fejlesztő intézmények lehetnek:

- katonai laboratóriumok: bár a két pólusú világ már megszűnt, a világ nagyobb katonai hatalmai továbbra is működtetnek ilyen laborokat, ahol a kifejlesztett vírusokat kisebb-nagyobb helyi konfliktusoknál fel is használják (pl. Sivatagi Vihar),
- egyetemi laboratóriumok: ma már inkább az egyes vírustípusok fertőzési mechanizmusait, jellemzőit vizsgálják, de egyetemi kutatási céllal készült el az első Word és az első Excel makróvírus is,
- vírusíró klubok: ma is nagy divat ilyen klub tagjának lenni, ahol a tagok minden egyes kifejlesztett vírusukat, különböző fertőzési mechanizmusokat taglaló írásaikat publikálják a többi tag részére,
- másolásvédelemmel foglalkozó laboratóriumok: manapság nem jellemző a büntető, sem a vírusos másolásvédelmek használata, de nagyon sok olyan vírus található, amely ilyen védett programokból szabadult a világra,

- számítógépet ismerő terroristák: akik eszméiket ilyen modern formában is szeretnék terjeszteni (nem hinném, hogy ez alapján több hívet találnának),
- bosszúálló emberek: pl. akiket elbocsátottak a munkahelyükről, vagy pedig valakire nagyon haragszanak,
- antivírus fejlesztő szakemberek: volt már rá példa, hogy ugyanaz írta a vírust és a vírus ellenszerét is, de ezek a cégek nem sokáig éltek,
- diákok: akik tudásukat társaik előtt ilyen módon szeretnék csillogtatni.

Adott vírus írásának az oka a víruskódban – néha – megtalálható szövegből kikövetkeztethető.

Egy számítógépvírus 3 részből áll:
I. reprodukciós rutin, amely:

!

1. keres valamilyen célpontot (programok, boot-szektor, partíciós tábla, makrót is tartalmazható állomány) amit megfertőzhet,
2. amennyiben talál valamilyen célpontot, ellenőrzi, hogy azt nem fertőzte-e már meg, igaz léteznek olyan vírusok is, amelyek ezt az ellenőrzést nem végzik el, így lehetséges, hogy ellenőrzés nélkül egy adott célpontot akár többszörösen is megfertőzzenek,
3. ha a fertőzést is ellenőrző vírus szerint nem fertőzött a célpont, akkor – vagy a fertőzést nem ellenőrző vírus esetén rögtön – a víruskódot utána fűzi, kicseréli vele, vagy felülírja a célpont egész vagy egy részét, makróvírus esetén a fertőzést tovább hordozó részbe (ami nemcsak az adott állomány makróterülete lehet!) másolja bele kódját (ennél a pontnál történik a többszörös fertőzés, ha nincs ellenőrzés),
4. ha bináris (nem makró) vírusról van szó, akkor a végrehajtási sorrendet a programon belül úgy állítja be, a víruskód mindenképpen végrehajtódjon, ha makróvírusról van szó, akkor úgy manipulálja az adatállományt, hogy annak betöltésekor vagy a felhasználó figyelmetlenségéből adódóan kapjon vezérlést a víruskód,
5. ha a fertőzést is ellenőrző vírusról van szó, akkor valahogy megjelöli a célpontot, hogy az már fertőzött, így elkerüli, hogy azt többszörösen megfertőzze.

A reprodukciós rutin 2.–3. pontjához annyit szeretnék hozzá tenni, hogy nagyon kevés olyan vírus létezik, amely 2. pontban leírt ellenőrzést kihagyja. Ugyanis az igaz, hogy a vírusból a fertőzöttség ellenőrzésnek a kihagyása

a vírus méretét csökkenti – ami minden vírusírónak a célja – ugyanakkor a vírus lebukásának veszélyét jobban megnöveli, hiszen így a többszörösen megfertőzött állományok hatalmas méretűre is növekedhetnek, amit esetleg a felhasználó könnyen kiszúr. Ráadásul ez növeli a vírus hibázási lehetőségeit is, hiszen ha pl. egy .com állományt többszörösen fertőz a vírus, akkor a .com program 64 Kbyte méret felett nem töltődik be és hibaüzenetet ad vissza indításkor.

A 4. ponthoz is lenne egy megjegyzésem: a mai bináris vírusok többsége a végrehajtási sorrendet úgy állítja be, hogy legelőször a víruskód hajtódjon végre, de elméletileg olyan vírus is elképzelhető, amelyik a **gazdaprogram** után fut le. Ez boot vagy partíciós vírusoknál könnyen megvalósítható, de programvírusoknál már problémát okoz, hiszen nem igazán lehet tudni, hogy hol és hogyan fejeződik be egy program, hiszen a mai programok nagy része interaktívan kommunikál a felhasználóval és annak döntései alapján folytatja és/vagy fejezi be futását. Tehát egy programon belül az összes lehetséges befejezési módot meg kellene vizsgálnia egy vírusnak ahhoz, hogy a program tetszőleges befejezése után a végrehajtás a víruskódra kerüljön. Ha nem mindegyik befejezési módhoz állítható be az a végrehajtási sorrend úgy, hogy a víruskód is végrehajtódjon, akkor a vírus nem lesz életképes.

A reprodukciós rutint minden vírus tartalmazza, az már más kérdés, hogy ez szinte minden víruskódban másképp van megvalósítva.

II. aktivizálódási feltétel ellenőrző rutin, amely:

1. ellenőrzi, hogy valamilyen esemény bekövetkezett-e vagy bizonyos feltételek teljesülnek-e,
   A vírus aktivizálódhat:
   - az év bizonyos napján vagy napjain, pl. dec. 5.-én
   - a nap egy bizonyos időpontjában, pl. 17 órakor
   - amikor egy bizonyos program fut, pl. ha egy DOS program fut
   - miután n db példányban sokszorosította magát, pl. 1000-szer
   - ha valamilyen billentyűzet-kombinációt érzékelt, pl. Shift-L
   - a számítógép újraindítása esetén
2. ha igen, akkor indítja a vírus objektív részét.

Még jó néhány feltétel lehetséges, csak a legismertebbeket soroltam fel.

III. objektív rutin, amely:

1. rendszerint valamilyen romboló utasítások sorozata,
2. ha nem romboló típusú, akkor általában valamiféle "reklámszöveg"-et jelenít meg.

Sok olyan vírussal találkoztam már, amely csak egy reprodukáló rutinból állt, a másik két részt nem tartalmazta. Tehát nem vizsgál meg semmiféle feltételt vagy eseményt, hogy objektív részét aktivizálja. Ez rendszerint akkor szokott előfordulni, ha a vírusíró csak tesztelési célból engedte el leendő vírusának szaporodó rutinját. Az "eredmények" után tökéletesíti a rutint, majd a másik két részt is hozzákapcsolja.

A PC vírusokat többféle szempont szerint is osztályozhatjuk.

PC VÍRUSOK OSZTÁLYOZÁSA

2.1.1. A fertőzés célpontja szerint

2.1.1.1. Boot-szektor és/vagy partíciós tábla vírusok

A **boot-vírus** lényege az, hogy egy PC-n nemcsak az **.exe, .com, .bat** kiterjesztésű file-ok futtathatóak, hanem a bekapcsolás utáni bootoláskor betöltődő **Master Boot Record** (**MBR** = partíciós tábla programja) és a bootszektor is. A vírusok ezen csoportja fertőzi ezt úgy, hogy ezek helyére írja magát és az eredeti boot-programot vagy partíciós táblát egy nem használt szektoron tárolják.

Más típusaik nem teljesen cserélik le az MBR-t vagy boot-szektort, azok helyére csak egy rövid betöltőprogramot tesznek, amely betölti a megfelelő helyen lévő teljes víruskódot. Ezt a módszert általában akkor alkalmazzák a vírusírók, ha a vírus méreteinél fogva nagyobb, mint a fent említett célpontok.

Nagyságrendekkel kevesebb ilyen típusú vírus készült, mint egyéb vírustípusokból, ma már nem is jellemző, hogy szaporodnának. Ez utóbbi a Windows 95 lemezkezeléséből valamint az alaplapokba beépített hardveres boot védelemből adódik.

A lényeg: boot/partíciós tábla vírus csak akkor válhat aktívvá, ha ilyen vírussal fertőzött lemezről próbáljuk indítani a gépet. Boot-vírus pl. a Stoned/Marijuana.

2.1.1.2. Programvírusok

A vírusok egy másik része a végrehajtható állományokba épül be vagy ezekhez az állományokhoz kapcsolódik. Bár – mint feljebb is írtam – elképzelhető olyan vírus, amely a hordozó programkódja után fut le, nem ez a jellemző, így ha elindítunk egy fertőzött állományt, akkor legelőször a víruskód hajtódik végre.

Ezeket a vírusokat kétféle szempont szerint osztályozhatjuk: milyen típusú állományokba, illetve azokba hogyan másznak bele.

Megfertőzhető végrehajtható állományok a következők lehetnek:

- io.sys-msdos.sys vagy ibmbio.com-ibmdos.com és command.com

Az első két legfontosabb **DOS** rendszerállományra rátelepülő vírusból nagyon kevés van, igaz ez érthető is valamennyire: ha a vírus ezekre a rendszerfile-okra is rátelepszik, ez a **DOS** "halálához" vezethet.

A command.com a parancsértelmező shell. Vírusok sokasága veszi ezt rögtön célba, mivel így a bekapcsolás után máris aktívvá válhatnak, vagyis az Autoexec.bat-ból esetlegesen induló védelmet ki tudják játszani.

- .com állományok

Itt a vírusnak illik ügyelni arra, hogy a file mérete ne legyen nagyobb, mint 64 Kbyte, ui. .com állományok maximális mérete csak ennyi lehet, mivel egy szegmensre töltődnek be a memóriába. Ellenkező esetben hibaüzenetet kapunk a Dos-tól és nem is tölti be a programot, így a vírus sem válhatna aktívvá.

- .exe állományok

Ez a file speciális felépítésű: a program elején van egy fejléc, ami alapján tölti be a memóriába a programot a Dos. Éppen ezért a vírusnak ezt a fejlécet korrekten kell kezelnie, ha szaporodóképes akar maradni. Ha rossz a fejléc, az könnyen a program lefagyásához vagy szintén hibaüzenethez vezethet.

- .exe állományok: NE – new executable
- .exe állományok: PE – portable executable

Az ilyen típusú programok Windows alatt futtathatóak, ezért az ilyen rendszer alatt szaporodni kívánó vírusoknak ezt a speciális file-formátumot jól kell ismernie.

- .bat állományok

Egy batchfile-ban szaporodó vírust elég nehezen lehet elképzelni, de többféle típusuk is létezik, attól függően, hogy milyen módon szaporodnak. Bővebben róluk a 3. fejezetben.

- .sys állományok

Ezek eszközmeghajtó programok. Ha ilyen file-t fertőz meg a vírus, akkor már a Config.sys végrehajtásánál lefuthat a kódja, ha a fertőzött eszközmeghajtó használva van. Nemcsak a .sys állományok, hanem a .com és .exe programok is lefuthatnak a Config.sys-ből – az Install parancs segítségével – de vannak olyan vírusok, amelyek futásához elengedhetetlen egy pa-

rancsértelmező shell – pl. a command.com memóriában való jelenléte – márpedig a Config.sys-beli eszközmeghajtó programok betöltésekor a parancsértelmező még nincs betöltve, sőt itt lehet hivatkozni arra, hogy esetleg a Command.com helyett milyen másik shell-t szeretne használni a felhasználó. Tehát egy parancsértelmező shell-t igénylő vírusnak nem igazán lehet a célpontja olyan .com vagy .exe program, ami a Config.sys-ből van indítva, mert ez az esetleges lefagyással vagy a hibaüzenettel szintén növeli a vírus lebukásának esélyeit.

- .ov? állományok (Ovl, Ov1, Ov2 stb.)

Programok részei, melyek a program indításakor nem töltődnek be a memóriába, csak később és akkor is csak ideiglenesen, de a kódjuk lefut...

- .bin állományok

Bináris állomány, melynek kódja végrehajtatható. Nagyon ritka az olyan vírus, amely ezekre rámászik.

- .obj, .lib állományok

Elég ritkák az ilyen vírusok, amelyek képesek megfertőzni tárgykódos vagy könyvtárállományokat, azonban mégis "érdemes", hiszen a tárgykódból egyszer futtatható program lesz, a könyvtárrutinokból pedig szintén. Elvileg lehetne olyan vírust is írni, ami különböző forrásállományokat – pl. Pascal, C – is megfertőz oly módon, hogy azokba beleilleszti saját, vagy más vírus forráskódját, hiszen ezekből a forrásállományokból is egyszer futtatható program lesz. Csakhogy, míg egy tárgykód állomány vagy könyvtárállomány fertőzés kevésbé tűnne fel a programozónak, addig egy esetleges ismeretlen és idegen forráskód belekerülése az eredeti forráskódba felkelthetné a figyelmét.

- .nlm állományok

A Novell Loadable Module szintén tartalmaz olyan kódot, ami végrehajtható, így ez is megtámadható.

- .dll állományok

A Dinamic Link Library állományok is tartalmaznak végrehajtható kódot, de az ilyen állományokat fertőző vírusokból szintén kevés van.

- .pif állományok

A .pif állományok a nem-Windows alkalmazásokhoz készült programinformációs file-ok, amelyek többek között tartalmazzák a futtatni kívánt program nevét. Ezt közvetetten úgy fertőzheti egy vírus, hogy a file-ban lévő programnévre való hivatkozást egy fertőzött állomány nevére cseréli le, így Windows-ból való indításkor először a fertőzött program fog lefutni.

Mint látható, nagyon sok megfertőzhető végrehajtható állomány létezik és minden egyes file-t más és másképp kell kezelni, ezért egy olyan vírus megírásához, ami többféle file-ra is képes rámászni, sok idő és tudás szükségeltetik. Ezenkívül azt is figyelembe kell venni a vírusíróknak, hogy ezzel (mármint, hogy több file-t képes megfertőzni a vírus) a mérete is növekszik, hiszen más file-típushoz más rutin kell. Igaz viszont, hogy ha minél több futtatható file-ba képes belemászni, akkor jóval gyorsabban elterjedhet a vírus.

File-vírusok szaporodási szempontok szerinti csoportosítása:

1. NEM FELÜLÍRÓ TÍPUSÚ

Ezek a vírusok úgy telepednek bele a programba, hogy azt nem bántják. Tehát a megtámadott program a legközelebbi futtatáskor is le fog futni, hacsak aközben a vírus nem aktivizálja magát (vagyis rombol).

1.a. *Hozzáfűző típusú (appendelő)*

A programok végéhez fűzi hozzá magát, majd a program elejére betesz egy ugróutasítást, ami a víruskód elejére mutat, hogy a víruskód hajtódjon végre először futtatáskor. Miután a vírus lefutott, az ugróutasítás helyére azokat a byte-okat helyezi vissza a vírus (csak a memóriában), ami eredetileg ott volt a programban, hiszen csak így futásképes a program. Erre legjobb példa a Vienna 648 vírus.

Appendelő vírusként kellene, hogy működjön az a vírus is, amit elméleti szinten emlegettem a reprodukciós rutin leírásánál (a "gazda" program lefutása után induló vírus).

1.b. *Programkód elé (beszúró típusú)*

Az is lehetséges, hogy a vírus a megtámadott program elejére teszi magát, így nem kell ugróutasítás, viszont a programnak ezt a részét arrébb kell tolnia vagy a file végére kell tennie, majd futtatáskor – miután a víruskód lefutott – a szabályos helyére teszi (a memóriában) a programkódot. Ilyen pl. a Péntek 13 (Jerusalem) vírus.

2. FELÜLÍRÓ TÍPUSÚ (OVERWRITE)

Ez egy elég durva megoldás a vírus részéről. Ebben az esetben a vírus a programba írja magát úgy, hogy az ott lévő információt (programkódot) nem menti el, tehát a program teljesen használhatatlan lesz, mert ha ilyen fertő-

zött programot elindítunk, akkor csak a víruskód fog lefutni, a program nem, mivel a programkód "elveszett". Mindig annyi byte veszik el a programból, ahány byte-os a vírus.

2.a. *Program elejére író (felülíró típusú)*

Ezen vírusok mindig csak a program elejére írják magukat, tehát ebben az esetben a programkód eleje fog elveszni. A legelső ilyen típusú vírus a V-405 volt.

2.b. *Véletlenszerű helyre író (felülíró típusú)*

A felülíró vírusok kis hányada mindig véletlenszerű helyre írja magát a megtámadott programba. Az előző esetben nem kell, míg ebben az esetben a program elejére kell írni egy ugróutasítást, hogy a vírus elinduljon (mivel itt nem a program elején van a vírus).

3. FELÜLÍRÓ ÉS NEM FELÜLÍRÓ HIBRID TÍPUSÚ

3.a. *Exe fejlécbe telepedő*

Még 1994 áprilisában jelent meg Magyarországon egy olyan vírus, amely bizonyos .exe file-ok hibáját használja ki. Mint tudva levő, jónéhány .exe file header részében van egy stack terület, ami üres. A vírus erre az üres területre telepedik be, a megtámadott program nem sérül és a vírus is "sikeresen" fertőzött.

3.b. *Com "üres" területre telepedő*

Hasonlóan az előzőhöz, ezek a vírusok a .com állományok lehetséges üres helyeire telepednek be. Bár az "átlagos" .com állományokra nem jellemző, érdekes módon a Command.com-nak van egy olyan területe, ami csupa 00-ból áll (a Command.com legvégén). Manapság már nagyon sok vírus ezt kihasználja és ezen részbe telepedik bele, a vírus aktívvá is tud válni és a felhasználónak sem tűnik fel a méretnövekedés.

Ez utóbbi 2 vírus nem igazán felülíró típus, hiszen a megtámadott program továbbra is futóképes – nem úgy mint az "igazi" overwrite típusoknál, ugyanakkor mégis a programok belsejébe telepednek bele.

4. COMPANION (CEB – COM, EXE, BAT) TÍPUSÚ

A **CEB** vírusok a **DOS**-nak azt a tulajdonságát használják ki, hogy ha a parancssorban nem adjuk meg a file kiterjesztését, akkor a rendszer először

egy .com állományt keres és indít el, ha ilyen nincs, akkor .exe állományt, ha ilyen sincs, akkor .bat állományt keres. Egy **CEB** vírus úgy fertőz .exe vagy .bat file-okat, hogy velük azonos néven létrehoz egy rejtett, .com kiterjesztésű file-t, ami magát a vírust tartalmazza. **Kiterjesztés nélküli programindításnál az eredeti program helyett mindig a vírus fog elindulni.**

5. FAT TÍPUSÚ

Mindössze egyetlenegy példányban rakja föl magát a lemezre, annak egy tetszőleges clusterébe. Minden végrehajtható fertőzött file kezdő clusterszámát a katalógusbejegyzésben átirányítja magára, a további FAT láncolást kódolja. Ha elindítunk egy fertőzött file-t, akkor a clusterszám átállítása miatt a vírus indul el. Tipikus példája a Dir2/Fat vírus.

Nagyon sok víruskutató a FAT és CEB típusú vírusokat megjelenésükkor nem a vírusok közé sorolta, mert akkori véleményük szerint a programvírusok a megtámadott file-ba ilyen vagy olyan módon beépülnek, míg ezek nem. Ugyanakkor mások szerint mégiscsak programvírusok, hiszen programokat támadnak meg, igaz egy kicsit trükkös módon fertőznek. Ezekkel a módszerekkel nagyon könnyen átverhető az integritás és egyéb vírusellenőrző programok többsége, hiszen a program tartalma és hossza sem változott!!

2.1.1.3. Makróvírusok

A makróvírusok olyan 5. generációs, közvetlenül nem végrehajtható, de makrókat tartalmazható állományokba épülő vírusok, melyek az adott alkalmazás makró nyelvi lehetőségeitől függően tudnak **szaporodni, rombolni vagy "jópofáskodni".**

Egészen a makróvírusok megjelenéséig élt az a tévhit, hogy a vírusok nem képesek adatállományokon keresztül szaporodni. Sajnos ez már nem így van, ez a kijelentés csak annyiban változott, hogy a vírusok az olyan adatállományokban nem képesek szaporodni, amelyek felépítésükből adódóan nem tartalmazhatnak makrókat. Tehát az ilyen adatállomány binárisan sem fog végrehajtódni, tehát a vírusnak nem "érdeke" ezeket a file-okat megfertőznie. Amennyiben egy vírus beleírja magát egy ilyen adatállományba, akkor valójában "csak" megrongálta, használhatatlanná tette a filet. Léteznek olyan vírusok is, hogy ha ilyen fertőzött adatállományt átnevezünk végrehajthatóvá, akkor abból képesek elszaporodni.

2.1.1.4. Hibrid vírusok

A vírusoknak egy nagyon kicsi része képes **boot-vírusként és/vagy programvírusként és/vagy makróvírusként is működni.** Bár elég bonyolult egy ilyen vírust megírni és a kód is elég termetes, viszont jóval szaporodóképesebb, mivel a fertőzési célpont szerinti vírustípusok "előnyeit" egyesíti magában.

Ha figyelembe vesszük az egyes típusok adottságait, akkor megállapíthatjuk, hogy jóval valószínűbb egy boot- és programvírus, illetve egy program- és makróvírus összekapcsolás, mint egy boot- és makróvírus kombináció, igaz ennek sem zárhatjuk ki a lehetőségét.

Hibrid vírusokra az egyik legjobb példa a **Onehalf**, mely Magyarországon 1994 októberében jelent meg. Ez a vírus floppy-n file-vírusként, winchesteren partíciós vírusként viselkedett. Bár a vírus mérete nem túl nagy, nagyon szaporodóképes.

2.1.2. Objektív rutin szerint

Mint azt az előzőekben már említettem, sok olyan vírus van, amely csak **egy reprodukciós rutin**ból áll. Ezekből a vírusokból **hiányzik az objektív rutin**, tehát nem vár semmiféle feltételre, **közvetlen károkat nem okoz**, viszont ha a szaporító rutinja helytelenül működik, akkor a megfertőzni kívánt célpontok károsodhatnak. Ha jó is a szaporodó rutinja, akkor is **drága és időigényes rombolásokat** okozhat azzal, hogy megfertőz gépeket.

Ha objektív rutin szerint szeretnénk jellemezni egy vírust, akkor nem elegendő azt mondani róla, hogy rendelkezik-e ilyen rutinnal vagy sem, hanem azt is meg kell határozni, hogy milyen közvetett vagy közvetlen károkat okozhat.

Ezek alapján az objektív rutinnal rendelkező vírusok két csoportba sorolhatók:

2.1.2.1. Nem romboló szándékú ("jópofa")

Ezek a vírusok különböző módon "**szórakoztathatják**" a felhasználót, pl. szöveg- vagy grafikamegjelenítéssel, zenéléssel, billentyűzet átdefiniálással stb.

Szöveg megjelenítés esetén általában a vírus íróját dicsőítő szöveg, vagy a vírusíróhoz közel álló (vírusfejlesztő, -terjesztő) csapatok üdvözlése található a kódban, de az is elképzelhető, hogy a vírust valaki számára

"címezték": pl. nem egy vírus nevesebb antivírus szakemberek nevére van címezve – a lejáratás érdekében, mintha ők írták volna a vírust.

A "zenélő" vírusok közül a legtöbb csak egy zenét tud lejátszani (pl. Yankee Doodle), de léteznek olyanok is, amelyek több zene közül szoktak egyet lejátszani (pl. 8 Tunes). Egy-két "zenélő" vírus minden billentyűhöz egy zenei hangot rendel, így ha gépelünk, az úgy tűnik, mintha zongoráznánk – persze korántsem egy igazi zenei darabot.

Egy "grafikus" vírusnak köszönhetően akár bohókás grafikák is megjelenhetnek a képernyőn, pl. ha lenyomunk valamilyen billentyűt, akkor egy alak néz szembe velünk, vagy egy lepke repül át a képernyőn, vagy egy gömbölyded női feneket rajzol ki a képernyőre, majd az elkezd mozogni (ez utóbbi a **Popo vírus**ra jellemző...), vagy a képernyő színeit megváltoztatja stb. Általában ezek a grafikák legalább EGA képernyőn működnek, monochrom esetében gyakran a gép lefagyásához vezethetnek, feltéve, ha a vírusírója nem programozta bele a vírusba a monitorvezérlő vizsgálatát.

Szeretném felhívni mindenkinek a figyelmét, hogy a fentiekben korántsem az összes lehetőséget írtam le – csak a legjellemzőbbeket, a vírus "pajkos kedve" az író fantáziájától függ. Gyakorlatilag az ilyen "jópofa" vírusok közvetett rombolásra is képesek: képzeljük el, amint a felhasználó a gépe előtt ül és hirtelen valami történik. Ha nincs "lelkiekben" felkészülve, akár ki is kapcsolhatja a számítógépet vagy formázhatja a fertőzöttnek hitt lemezeket – mindenféle adatmentés nélkül... Ha nem ijed meg, akkor viszont nem használja vagy nem használhatja addig a gépet, amíg a vírus – megfelelő szakemberek segítségével – ki nem lesz irtva a gépről. Ilyenkor pedig a kieső gép- és munkaidőt, valamint a költségvetésbe előre bele nem kalkulált vírusirtás költségeit ki fizeti ki?

2 1 2 2. Romboló szándékú

Egy komplett számítógépes rendszer hardver és szoftver elemekből épül fel, éppen ezért fontos megvizsgálnunk, hogy milyen mértékben tudja károsítani egy vírus az egyes elemeket.

- *Szoftverromboló*

Bátran kijelenthetjük, hogy a **PC-k jellemző operációs rendszerei** – Dos, Windows 3. 1, Windows for Workgroups 3. 11, Windows 95, Windows 98 – közül egyik sem rendelkezik komolyabb védelemmel, így **a rendszer bármely fontosabb része** (*boot szektor, partíciós tábla, FAT tábla, könyvtárbejegyzések, rendszerállományok*), a *telepített programok és adatok összessége vagy egy része minden probléma nélkül* **törölhető, felülírható, a rendszer bármely szoftveres része megsemmisíthető**.

A szoftverromboló vírusok között csak abban tudunk különbséget tenni, hogy azok **hogyan pusztítanak**: *erőteljesen, részlegesen, szelektíven vagy véletlenszerűen.*

Az erőteljesen romboló vírusok rendszerint formattálják – akár alacsony szinten is – vagy valami "byte-szeméttel" töltik fel az egész lemezt. A felhasználó számára az a "legjobb" ha a vírus "simán" formázta a lemezt, mert ez esetben egy **Unformat** paranccsal az **eredeti tartalom** rendszerint visszaállítható.

A részlegesen pusztító vírusok szintén felülírják, módosítják a lemezt, de annak csak bizonyos területeit. Az egyes területeket hibás szektoroknak is bejelölheti a vírus. A **Badsectors** vírus innen (is) kapta nevét.

A szelektíven romboló vírusoknak – mint a nevük is mutatja – előre meghatározott megsemmisítendő célpontjaik vannak, pl. az egyes rendszerprogramok megrongálása, rendszerterületek szétrobbantása. A legtöbb vírus mielőtt törölné, **szeméttel szokta feltölteni** a programot, így egy esetleges **fertőzés-elhárítás után Undelete-tel visszahozott programok már nem működőképesek:** ilyen módon törölt program elindításakor a gép biztos, hogy lefagy. Az egyik legismertebb vírus a "Péntek 13" minden 13-ra eső pénteken törölte az elindított programokat.

A **véletlenszerűen romboló vírus** pedig véletlenszerűen, nem sűrűn cserélgetheti az adatokat a lemezen vagy a memóriában, a ki- és beviteli eszközökre irányuló adatokat megváltoztathatja, ezzel könnyen elérheti, hogy az *egyes programok futás közben lefagynak, ami néha elég kellemetlen lehet (különösen ha nincs mentés a számítógépen folyó munkáról).*

- *Hardverromboló*

Sajnos a PC hardverelemei is rombolhatóak szoftveresen úton. Még mielőtt bárki is nem merne PC-t venni a továbbiakban, meg kell hogy nyugtassam, a vírusok **nagyon kicsi része képes hardver részek elpusztítására**. Ennek egyik oka az, hogy nagyon profi hardver tudást igényel egy ilyen vírus megírása, másrészt a különböző hardverelemek elpusztításához szükséges információkat a gyártók – érthető módon – nem adják ki.

Vizsgáljuk meg egyenként az egyes PC hardverelemeket, hogy azokat hogyan károsíthatja egy vírus.

1. PROCESSZOR

A Motorola 68000-as processzort megalkotó mérnökök már gondoltak arra, hogy a processzor tudását valamilyen úton szoftveresen bővítsék. Ezt olyan módon szerették volna elérni, hogy bizonyos utasításkódokhoz nem

rendeltek utasítást, hanem azt szabadon hagyták. A későbbiek folyamán ezekhez a szabadon hagyott kódokhoz lehetett hozzárendelni tetszőleges új utasításokat. A szoftveresen "felbővített" processzorokkal viszont az volt a gond, hogy a kibővített utasításkészletben nem voltak egymással kompatibilisek, így ezt a fejlesztési irányt teljesen leállították. Tudom, a mai PC-k nem Motorola, hanem x86-os processzorokkal működnek, de nem tartanám elképzelhetetlennek, hogy egy x86-os processzor is rendelkezzen ilyen lehetőséggel, csupáncsak ez a "feature" nem publikált...

Lehet, hogy a kedves Olvasó is találkozott már olyan jobb minőségű alaplappal, amelyben nem jumperek segítségével lehetett beállítani a processzor sebességét illetve a processzor működéséhez szükséges feszültséget, hanem mindezt **BIOS**-on keresztül lehet megvalósítani. Így processzorcsere esetén nem kell bajmolódni az alaplap jumperelésével, hanem csak elegendő beállítani a megfelelő értékeket a **BIOS**-on belül. Mint tudjuk a PC a **BIOS** beállításait egy bizonyos **CMOS** memóriában tárolja, ami szoftveresen szintén megváltoztatható: pl. a különböző "Bios jelszó kinullázó" programok a **CMOS**-ban tárolt jelszót törlik. Ezek után már nem nehéz elképzelni egy olyan vírust, ami a processzor sebességét vagy a feszültséget nagyobb értékre állítja és mint tudjuk, a processzorok a túlhajtást csak egy bizonyos ideig bírják... Igaz egy ilyen vírus eléggé alaplapfüggő lenne, hiszen nem mindegyik jumper nélküli.

A **Pentium II** processzorok rendelkeznek egy EEPROM-mal, melyben olyan konfigurációs adatokat tárolnak, hogy az adott processzort milyen gépbe építették bele. Ennek az EEPROM-nak a tartalmát szoftveresen is lehet változtatni, tehát akár egy vírus is átírhat benne tetszőleges információkat, igaz, ez nem számítható teljesen hardverrombolásnak, hiszen csak a konfigurációs adatok nem lesznek helyesek.

2. WINCHESTER

Egy winchestert többféleképpen is tönkre lehet tenni: az olvasófejjel valamilyen úton "szántást" csinálni a lemezen, a winchester flash memóriáját állítgatni, control szektort átírni.

A winchester fejét nem lehet közelebb állítani a lemezhez, viszont a „szántást" lehet csinálni régebbi konstrukciókkal: a fejet olyan rezgésszámon lehet ide-oda mozgatni, hogy a fokozott igénybevétel következtében a fej felfüggesztése elfárad. Ehhez azonban rengeteg idő kell – mire a fej leszakad, az pedig a kevésbé hozzáértő felhasználónak is feltűnhet, ha állandóan kerreg a winchester.

Véleményem szerint "**szántást**" nemcsak így lehet csinálni, hanem egy kicsit trükkösebb formában is: a gép bekapcsolásakor a winchester feje parkoló állásban van addig, amíg a lemezek nem pörögnek fel, a megfelelő fordulatszám elérése után a fej bemozdul a lemezek fölé. Így egyfajta "légpárna" van a lemezfelület és a fej között, tehát fizikailag nincs közvetlen kapcsolat közöttük. Ám mi lenne akkor, ha a fej a lemezek felpörgése előtt bemozdulna? "**Szántás**" keletkezne a lemezen. Ezt szoftveresen bekapcsoláskor nem lehet megvalósítani, hiszen ekkor még nem fut semmi szoftver a gépen, de szerintem energiatakarékos üzemmódnál igen. Amikor ugyanis egy gép energiatakarékos üzemmódra vált át – ami szoftveresen is elérhető, gondoljunk csak a Windows 95-re, akkor rendszerint a winchestert is leállítja. Ha ilyenkor egy program hirtelen bemozdítja a fejet a lemezek fölé, majd visszavált nem energiatakarékos üzemmódra, akkor a winchester sérülhet, hiszen a lemezek úgy pörögnek fel, hogy a fej a lemezek felett tartózkodik...

A winchesterek többsége nagyobb kapacitású, mint ami rájuk van írva, de vannak olyan területeik, amelyek **bad szektorosak**, ezeknek a szektoroknak a helyét a **winchester flash memóriája tárolja**. Ez a memória **szoftveresen is állítható**, így akár azt is be lehet állítani, hogy az **egész winchester bad szektoros**.

A winchestert a controlszektor átírásával is tönkre lehet tenni, ugyanis ez a szektor létfontosságú a boot-oláshoz, mert ez tartalmazza a winchester különböző adatait.

3. ALAPLAP

A mai gépek már Flash BIOS-szal vásárolhatóak, ez lehetővé teszi a BIOS program frissítését szoftveresen. Egyes alaplapoknál egy jumper segítségével lehet engedélyezni/tiltani a Flash BIOS írását, de sajnos rengeteg olyan is létezik, amelynél nincs ilyen jumper vagy a jumper állásától függetlenül engedi a BIOS írását. Ez a szoftveres frissítés viszont visszaélésekre is ad lehetőséget: a kevésbé neves alaplapok BIOS programja úgy módosítható, hogy a gép a legközelebbi bekapcsoláskor ne induljon el. Ezeknél a gépeknél ez együtt jár egy BIOS cserével vagy egy BIOS újraírással, igaz ez utóbbi "házilagosan" is megoldható, de ez nagyon veszélyes és a kevésbé hozzáértő embernek jobban megéri egy BIOS égetővel rendelkező céget felkeresni.

Hogy ne legyen teljes a vírusírók öröme, azt is meg kell említenem, hogy a nevesebb és "high-tech" alaplapok rossz BIOS-szal is képesek elindulni: a Flash mellett van egy ROM, ami biztosítja, hogy még a BIOS kisze-

dése esetén is boot-oljon a gép. Ha nem BIOS kivételről van szó, akkor egy jumper áthelyezésével beindul egy recovery program, amely lemezről képes visszaállítani a BIOS tartalmát. Az első ilyen lehetőséggel felruházott alaplap a Micronics volt.

A Flash BIOS pusztítás további lehetőségeihez tartozik a maradandó károsítás: mivel a Flash BIOS-ok csak néhány ezer átírást bírnak ki, így elegendő, ha csak egyetlen byte-ot rendszeres időközönként változtatna egy program – adott esetben egy vírus. Ez már csak egy komplett BIOS cserével oldható meg.

4. FLOPPY

A floppy meghajtók egyenáramú motorja felépítésükből adódóan a hosszabb idejű működést nem bírja, ezen okból kifolyólag rendszeresen le kell állnia ("pihenni"). Mivel azonban ez a leállás szintén szabályozható szoftveresen, ezért a floppy meghajtót is tönkre lehet tenni.

5. MONITOR

PC-n lehet szoftveresen szabályozni a videojel frekvenciáját, aminek "segítségével" megoldható a nagyon gyenge minőségű monitorok tönkretétele.

6. MONITORVEZÉRLŐ KÁRTYA

A monitorvezérlő kártyák is tönkretehetőek "össze-vissza" küldött IN-OUT utasításokkal illetve bizonyos kép-előállítási paraméterek rossz meghatározásával.

7. NYOMTATÓ

Szoftverből szintén meg lehet keverni az Epson nyomtatók EEPROM-ját, aminek eredményét nyomtatáskor lehet majd tapasztalni: csak zagyvalékot kapunk.

Bármilyen hihetetlennek is tűnik, a jövőben jóval több szoftverből is rongálható hardver fog megjelenni a piacon, hiszen a vezető hardvergyártók egyre inkább ráállnak a Flash-es technikára, ami a legolcsóbb és leggyorsabb megoldás az adott hardvereszköz tudásának folyamatos bővítéséhez.

Szintén a romboló szándékúak közé sorolandóak azok a vírusok, amelyek miközben elkápráztatják valami "jópofa" dologgal a felhasználót, valami gonoszat is cselekszenek.

2.1.3. Memóriával való "viszony" szerint

Ennél a szempontnál azt vizsgáljuk, hogy a víruskód lefutása után a víruskódja bent marad-e a memóriában vagy sem, ha bent marad, akkor az egész kód vagy annak csak egy része lesz aktív.

2.1.3.1. Rezidens

Amikor egy rezidens vírus kódja először lefut (pl. fertőzött program indítása), akkor "beül" a memóriába, s egészen addig fertőz, amíg a rendszert újra nem indítjuk vagy magát a vírust inaktívvá nem tesszük valamilyen antivírus programmal.

A memóriába való "beüléskor" a vírus kódja a memória egy megfelelő részébe bemásolódik (ezt a reprodukciós rutin intézi), majd a géptől valamilyen vezérlést (vagy akár vezérléseket) elvesz, vagyis **meghatározott interruptokat** magára irányít. Ennek eredményeképpen bizonyos feladatok végrehajtásakor nem az operációs rendszer eredeti rutinja, hanem a vírus megfelelő interrupt kezelő rutinja fog lefutni. Ezáltal **egy vírus szinte mindent képes ellenőrizni,** pl. hogy milyen programot indítottunk el. Az már vírustól függő, hogy ha egy adott vezérlés lekezelő rutinja fut, akkor mi fog történni, a legutolsó példánál maradva a **rezidens file-vírusok többsége** az elindított programokat azonnal **megfertőzik.**

Szinte valamennyi vírus rezidens, ezek a vírusok szaporodóképesebbek is, hiszen a gépet gyakorlatilag irányítják és kényük kedvük szerint fertőznek.

Minden boot-vírus rezidens, de nem minden file-vírus rezidens!

Nagyon sokféleképpen lehet megoldani, hogy egy vírus rezidenssé váljon. Az antivírus programok többsége ellenőrzi a memóriát, éppen ezért a vírusírók egyre furfangosabb ötleteket eszeltek ki, hogy a program ne érzékelje a vírust a memóriában. Néhány forma, ahogy rezidenssé szoktak válni:

- egyszerű **TSR** (**T**erminate and **S**tay **R**esident) programként ül be a memóriába, vagy
- a felső memóriába installálja magát, vagy
- a víruskód nagy része a videomemória olyan részébe ül be, amely nincs használva, vagy
- **MCB**-n (**M**emory **C**ontroll **B**lock) keresztül válik rezidenssé, vagy
- a **DOS** puffereiben lesz rezidens, vagy
- mindig ugyanarra a címre tölti be magát minden gépen, vagy
- a 640 KB fölötti szabad részre tölti magát.

A **rezidens vírusok**at úgy lehet inaktívvá tenni, hogy az interrupt kezelő rutinoknak az eredeti operációs rendszerbeli címét visszaállítjuk, vagyis a legközelebbi interrupt híváskor a víruskód már nem futhat le. A víruskód a legközelebbi rendszerindításig a memóriában benn marad, de mégsem aktív (halott vírus).

A rezidens típusok közé sorolnám azokat a vírusokat is, amelyeknél a szaporító rutin csak a végrehajtás idejére kerül be a memóriába, ám a víruskód egy kisebb része, az aktivizálódási feltétel ellenőrző rutin és az objektív rutin rezidenssé válik, de ez a rezidens rész már nem szaporodik tovább (mivel azt már elintézte a nem rezidens szaporító rutin). A jelenleg létező vírusok alig pár ezreléke szaporodik így.

2.1.3.2. Nem rezidens

A nem rezidens vírus csak a fertőzött program végrehajtása idejéig kerül be a memóriába, a géptől semmiféle vezérlést nem vesz el a kód lefutása után. Az ilyen típusú vírusok futásuk folyamán azonnal célpontot keresnek, és rögtön fertőznek. Sajnos így is meg lehet oldani, hogy a winchesteren az összes program fertőzött legyen: addig nem engedi futni az eredetileg indított programot, amíg az összes lehetséges célpontot meg nem fertőzte a lemezen.

Nem rezidens vírus kizárólag file vírus lehet.

Többekben felmerülhet a kérdés, hogy a makróvírusokat memóriával való viszony szerint hova lehet besorolni. Igazság szerint mindkét helyre, hiszen a makróvírusok többsége "rezidens" módon átmásolja makróit egy globális makróterületre és bizonyos vezérléseket (pl. mentés) elvesz az adott alkalmazástól. Ugyanakkor léteznek olyanok is, amelyek nem rezidens módon azonnal fertőznek. De minderről bővebben a 4. fejezetben.

2.1.4. Generáció szerint

Az itt felsorolt vírusok közül szinte valamennyi működési elvét és viselkedését a 3. és a 4. fejezetben bővebben tárgyalja.

2.1.4.1. Első generációs vírusok

Mint a nevükből is látható ezek voltak az első vírusok, melyek PC-ken voltak képesek szaporodni. Egyszerű terjedő rutinnal rendelkeztek, melyet könnyű volt megérteni, visszafejtésük nem volt túl bonyolult. Egy-két család szinte teljesen kihalt pl. a hibás terjedő rutin miatt, vagy a megváltozott számítástechnikai környezetből fakadóan.

A vírusírók a későbbiek folyamán ezeket a vírusokat már valami különleges dologgal ruházták fel, mint pl. lopakodási stílus.

2.1.4.2. Lopakodó (stealth) vírusok

A **lopakodási technika** abból áll, hogy a lopakodó típusú vírus a **fertőzése során bekövetkezett változásokat elrejti, mindent az eredeti, fertőzetlen állapotban mutat:** *programokat, azok hosszát, vagy boot-szektort, partíciós táblát stb.* Persze, hogy mit mutat eredetiben, az attól is függ, hogy mit fertőz, **mivel vannak lopakodó típusú boot- és file-vírusok** is.

Minden lopakodó vírus egyúttal rezidens is, hiszen csak így képes a felhasználó elől elrejteni a változásokat. Ha nem lennének azok, akkor rögtön észre lehetne venni a fertőzés során bekövetkezett változásokat, pl. a file-ok hosszváltozását.

2.1.4.3. Polimorf, mutációs vírusok

A **mutációs technika lényege** az, hogy a vírus teljesen átírja magát, véletlenszerűen előállított **változatokat készít önmagából,** amelyek tetszőleges hosszúságúak lehetnek. Visszafejtés elleni trükköket is tartalmazhatnak. Az állandó alakváltoztatás miatt nehezebb megtalálni ezeket.

Az eddig megjelent összes **polimorf** vírus **file-vírus**!

2.1.4.4. FAT és CEB (companion) vírusok

Ezek a vírustípusok egy új generáció tagjai is, ugyanakkor más szaporodási módszereket alkalmaznak, mint mondjuk pl. a lopakodó vagy a mutációs vírusok. Azok ugyanis rendszerint a file-hoz hozzákapcsolódnak, míg ezek valamilyen trükkös módon támadják meg a programokat. Éppen ezért, ha egy vírus CEB vagy FAT vírusok közé tartozik, akkor már generációs típus szerint és szaporodási szempont szerint is osztályoztuk.

2.1.4.5. Makróvírusok

A makróvírusokat a fertőzés célpontja szerinti osztályozásnál már említettem, most ugyanazt nem kívánom még egyszer leírni. A 4. fejezetben ezekről a vírusokról is bővebben tájékozódhat a Kedves Olvasó.

2.1.5. Szaporodási sebesség szerint

Szaporodási sebesség szempontjából is van különbség az egyes vírusok között, ráadásul a makróvírusokat egészen a többféle alkalmazás alatt is futni képes típusok megjelenéséig nem igazán lehetett hova besorolni ezen szempont alapján.

2.1.5.1. Lassú fertőző

Azok a rezidens vírusok tartoznak ebbe a csoportba, amelyek csak az elindított, fertőzetlen programot támadják meg, vagy pedig nem fertőznek minden egyes állomány hozzáféréskor.

A nem rezidens vírusok közül szinte valamennyi ebbe a csoportba tartozik, hiszen a fertőzött program elindítása után keresnek egy áldozatot, azt megfertőzik, majd a programkód kezd el futni.

2.1.5.2. Gyors fertőző

Gyors fertőzők azok a vírusok, amelyek, ha aktívak a memóriában, nemcsak az elindított programot fertőzik meg, hanem más programokat is egyidejűleg, pl. azokat, amelyeket megnyitnak.

Szintén a gyors fertőző vírusok közé tartoznak azok a nem rezidens vírusok, amelyek kódjuk lefutása után addig nem engedik futni a hordozó programot, amíg az összes lehetséges célpontot meg nem fertőzték.

Ha ilyen vírus van a memóriában, akkor nagyon könnyen előfordulhat, hogy miközben egy víruskereső programot futtatunk, az összes programunk megfertőződik, mivel az ellenőrző program minden programot megnyit, hogy azt ellenőrizze. Ezért fontos mindig vírusmentes lemezről indítani a gépet...

A makróvírusok közül azok tekinthetőek gyors fertőzőnek, amelyek több alkalmazás alatt is képesek szaporodni, vagy amelyek aktivizálódásukkor megkeresik az összes lehetséges célpont állományt és megfertőzik azt. A többi makróvírus "lassú" fertőzőnek tekinthető, bár a makróvírusok aktuális terjedési üteme alapján ez a jellemző nem teljesen fedi a valóságot.

2.1.6. Visszafejthetőség szerint

A számítógép "lelkivilágával" komolyabban és mélyebben foglalkozó felhasználók az elkapott vírusokat szeretnék is visszafejteni, megtudni annak

működési elvét. Egynémely vírusoknál ez könnyűszerrel megoldható, míg másoknál már kevésbé.

2.1.6.1. Könnyen visszafejthető, "nem ellenálló"

Az egyszerűbb bináris vírusok nem alkalmaznak semmiféle trükköt, hogy kódjuk visszafejtése nehezebb legyen. Gyakran a **DOS**-féle **DEBUG** elég hozzájuk. Főleg 1. generációs vírusokra jellemző.

Makróvírus esetén az adott alkalmazás makrónyelvi szerkesztőjében tanulmányozható a vírus forráskódja (hiszen a makróvírusok "forrás" szinten szaporodnak).

2.1.6.2. Nyomkövetéskor billentyűzetet letiltó

Bináris vírusoknál, ha lépésenként nyomkövetéssel próbáljuk visszafejteni a víruskódot, akkor az képes lehet rá, hogy letiltja a billentyűzetet, majd a kód lefutva "átveszi a hatalmat" a gép felett. Ráadásul egy ilyen rutin megírása nem ütközik nagy nehézségekbe, mert ilyen rutint bárki találhat jobb assembly könyvekben.

Makróvírus esetén ez a technika nem alkalmazható, de egyéb trükkökkel (pl. menüátdefiniálással) "védekezhet" a vírus.

2.1.6.3. Víruskód titkosító

Nagyon sok bináris vírus saját kódjának nagy részét mindig titkosítja – rendszerint más-más kulccsal, így egy disassemblerrel visszafejtett lista nem lesz értelmes, vagy abból csak a dekódoló rutin értelmezhető. Ilyenek voltak már a polimorf vírusok legelső típusai is, az oligomorf vírusok. Ez esetben a visszafejtésre megoldás lenne a dekódoló rutin lefuttatása, ez azonban nem mindig alkalmazható.

Makróvírus egyes **makrói is titkosítva** lehetnek, de itt a titkosítást nem a vírus intézi, hanem maga az alkalmazás, amiben a vírus fut.

Mint látható, jó néhány szempont szerint lehet osztályozni a vírusokat. Egy adott vírusra minden egyes szempont szerint mondható legalább egy jellemző, sőt az is előfordulhat, hogy egy szempont szerint többféleképpen is jellemezhető: pl. a **Whale vírus generáció szerint polimorf és lopakodó vírus is.**

Ezen szempontok szerinti **vírus-meghatározás nagyon fontos**, hiszen a vírus detektálása, irtása, adott családbeli besorolása és rendszerezése másképp is történhet típustól függően.

2.2. TRÓJAI PROGRAMOK

Mint Trója ostrománál, ezeknél a programoknál is az **álcázás a lényeg**. Az ilyen típusú programok onnan kapták nevüket, hogy valójában nem azt cselekszik, amire a felhasználó szeretné használni, hanem egészen mást, írójuk céljának megfelelően.

Egy számítógéprendszerbe kétféle úton juthat be egy trójai program: felmásolják ártatlan felhasználók, vagy pedig szándékosan gonosz munkatársak által, valamilyen felderítési vagy pusztító szándékkal. *Ellentétben a vírusokkal a trójai programok nem képesek önreprodukálásra, tehát nem szaporodnak!!*A "**Vírusvédelmi tanácsok**" című fejezet próbál ötleteket adni arra is, hogyan „szúrjuk ki" ezeket a programokat.

2.2.1. Általános trójai programok

Nagyon gyakran valamilyen közismert és gyakran használt programba építenek be valamilyen pusztító rutint, de nem a gyártók, hanem hackerek. Szaporodó rutin nincs bennük, ezt megoldják a felhasználók. Hiszen ha nagyon jó programról van szó, akkor természetesen a felhasználók odaadják egymásnak, terjesztik egymás közt. Aztán persze futtatni is fogják, aztán egyszercsak...

Az általános trójai programok közé sorolják a trójai makrókat is, melyek ugyanúgy aktivizálhatják pusztító makrójukat, ahogy a makróvírusok, csak itt szó sincs szaporodásról. A makróvírusokról, illetve a makrók pusztítási lehetőségeiről bővebben a 4. fejezetben lehet olvasni.

2.2.2. Másolásvédelmi trójai programok

Léteznek olyan programok, amelyeket a kifejlesztő cég úgy akar megvédeni az illetéktelen másolástól, hogy a programot másolásvédelemmel ruházza fel. Ez azt jelenti, hogy a program valamilyen feltételek között hajlandó csak működni, pl. a program futása alatt egy, a cégtől kapott eredeti lemeznek benn kell lenni a floppymeghajtóban, vagy pedig valamilyen kártyával hajlandó csak működni. *Persze a cégtől kapott lemez egyáltalán nem, vagy csak nehezen másolható.* Ha a program *nem a felinstallált környezetet érzékeli* vagy *pedig nem tetszik neki valamilyen feltétel (pl. nincs lemez)*, akkor a **lemezen lerombolja az adatokat, tönkreteszi a programokat,** mondván: a bűnös hadd bűnhődjön (aki programot lopott). *Ma már egyre ritkább az ilyen típusú másolásvédelem.*

A másolásvédelmeknek a humánusabb fajtájánál nincs rombolás, itt egyszerűen ha a program nem az installált környezetet érzékeli, akkor nem fog lefutni.

2.2.3. Gazdasági trójai programok

Régebben könyvelői és egyéb, pénzért eladott programok sokasága volt ellátva ilyen gonosz rutinokkal. **A garanciaidő lejárta után szokott "adatvesztés"** előfordulni, s hiába a reklamáció a cégnél: a garanciaidő lejárt, tessék megvásárolni a következő verziót (ami szintén ilyen időzített bombát tartalmazhatott).

2.2.4. Hálózat felderítő trójai programok

Pusztítás más és más típusú lehet, nem kell mindig az adatok elvesztésével számolni. Előfordulhat az is, hogy egy olyan felhasználó, akinek semmiféle joga nincs egy hálózathoz, ír egy trójait, ami ha fut, a futtató felhasználó file-jainak a hozzáférési jogait állítja át. Ezután bárki hozzáférhet a mit sem sejtő felhasználó file-jaihoz, egészen addig, amíg az hosszú idő múlva észre nem veszi. De lehet, hogy addigra már késő lesz...

Pl. ha ráültetünk egy rutint a Novell hálózati operációs rendszer Loginjára, ami nézi, hogy milyen jelszóval jelentkezik be valaki, majd egy általunk is elérhető állományba ezt a jelszót lejegyzi, akkor ez hálózatfelderítő trójainak számít.

2.2.5. Beépített vírust tartalmazó trójai programok

Trójainak nevezzük azokat a programokat is, amelyekben víruskód van, amely tömörítve is lehet (lehetőleg egy ismeretlen tömörítő eljárással, mivel az ismertebb algoritmusokkal összenyomott állományokba a legtöbb víruskereső/irtó bele tud kukkantani). Valamely feltételek teljesülése esetén szabadjára engedi a vírust. Pl. ilyen volt régebben az Eagle.exe, ami egy sast rajzolt ki a képernyőre, de tömörítve egy Péntek 13 (Jerusalem B) vírust tartalmazott.

Vírusterjesztők durva tréfája szokott lenni, hogy ismertebb vírusellenes programra egy vírust "ültetnek", s ezt a vírussal fertőzött verziót kezdik el terjeszteni más, "magasabb" verziószámmal. Pl. a 45-ös verziószám előtti McAfee féle VirusScan soha nem ellenőrizte önmaga épségét, amikor a file-

okat ellenőrizte, azokat meg kellett nyitnia. Éppen ezért, szinte érdemes volt szántszándékkal egy olyan vírussal megfertőzni, amely minden egyes file-nyitáskor megfertőzi a megnyitott file-t. **A Scan45-öt megfertőzték V-2000 vírussal, majd Scan65-ként kezdték elterjeszteni.** Hasonlóképpen járt a **Sysdoki őse, a Prgdoki** is.

Nemcsak antivírus programokra szoktak ráültetni vírust és "magasabb" verziószámmal terjeszteni, hanem más közkedvelt felhasználói programokra, mint pl. tömörítő programokra vagy játék programokra. **Operációs rendszerek** is estek már áldozatul vírusterjesztők gonoszságának: pl. az **MS-DOS 3.30 installációs** lemezeire *gonosz emberkék* egy **Michelangelo boot-vírust ültettek,** s így terjesztették, még 1992 elején. A Novell hálózati operációs rendszer lemezeire **Stoned-3** vírus került, 1991 végén.

2.3. PROGRAMFÉRGEK

PC-ken a vírusok az operációs rendszer védelmének hiányát használják ki, a nagyobb számítógépes rendszereken alkalmazott rendszerekben viszont már operációs rendszer szinten beépítenek annyi védelmet, hogy az adott rendszer alá vírust írni szinte képtelenség vagy legalábbis nem érdemes. Igaz létezik már PC-re is ilyen komolyabb védelemmel felruházott operációs rendszer a **Linux**, de ennek **használata nem jellemző az "átlagos" felhasználók** között.

Ám a legkomolyabb védelemmel felruházott operációs rendszerben is maradhat biztonsági rés és ezt kihasználva tudnak gépről gépre szaporodni az un. programférgek, amelyek **(eltérően a vírusoktól)** nem hordozó programokhoz csatolódnak hozzá vagy makrós állományokba épülnek bele, hanem komplett programok. Tehát nincs hordozó program, a **fertőzés kiinduló pontja egy komplett féreg program.**

A programféreg belépve a rendszerbe elkezdi feltérképezni a fertőzött géppel kapcsolatban álló gépeket, majd ha a megfertőzni kívánt távoli gépen is létezik a szaporodáshoz kihasznált biztonsági rés, akkor belép a távoli gép rendszerébe, majd innen szintén tovább szaporodik. Vagyis egy féreg a hálózati kapcsolatokat használja fel arra, hogy rendszerről rendszerre szaporodjon, tehát csak azokat a rendszereket támadhatja meg, **amelyek valamilyen kommunikációs vonalra vannak kapcsolva**. Ezekből is következik, hogy a féreg terjedése korlátozva van a hálózat méretétől és a teljes hálózatra kapcsolt összes gép rendszerbiztonsági résének létezésétől függően.

Az eddig megjelent **programférgek pusztító rutinnal nem voltak felszerelve**, hacsak azt nem vesszük annak, hogy a **megtámadott rendszeren saját kódjukat megsokszorozták lemezen és memóriában egyaránt**, tehát a rendszer erőforrásaiból rengeteget felhasználtak, más hasznos program futását korlátozva ezzel. Elképzelhető olyan programféreg is, amely nem sajátkód sokszorozásra (és erőforrás korlátozásra) van beprogramozva, hanem információ szerzésre, pl. *jelszótábla kihozatalára, tehát ez a technika igazán veszélyes is tud lenni. Gondoljunk csak bele, ha pl. egy banki nagygépes rendszerből sikerülne mondjuk a jelszótáblát kihozni...*

Nagygépes rendszereknél a védekezés egészen más filozófián alapul, mint PC-k esetében, hiszen itt nemcsak tüneti kezeléssel (tehát féregirtással) vetnek gátat az újrafertőződésnek, hanem azzal is, hogy a **biztonsági rendszeren található rést "befoltozzák"**. Ezt rendszerint az adott operációs rendszer fejlesztő cég valósítja meg, többnyire ingyen, próbálva így kiköszörülni a tekintélyen esett csorbát.

A biztonsági rés befoltozását követően a programféreg fejlesztőknek **újabb ilyen rést** kell keresni, de mivel az ilyen rések egyre kevésbé felderíthetőek – a folyamatos operációs rendszer fejlesztéseknek köszönhetően –, ezért nagyon sok munkával és vesződséggel jár egy újabb, szaporodóképes féreg kifejlesztése. Ezért is egyre ritkábbak a programférges fertőződések.

Az eddigi leghíresebb, *"Internet Worm" nevű programféreg* 1988-ban szabadult rá az Internet-re, az akkoriban rákötött több mint 60.000 gép 10%-a (!!!) fertőződött meg és csak azért ennyi, mert ez a féreg elsősorban a Berkeley Unix sendmail-jének sajátosságát kihasználva tudott szaporodni gépről gépre. Vagyis azok az *Internet-re kötött gépek ahol más operációs rendszert használtak, nem fertőződtek.* Rendkívül gyorsan szaporodott, katonai kísérleti laboratóriumok számítógépeit is sikerült megfertőznie...

Azt is szeretném megjegyezni, hogy az "Internet Worm" szerzője, a Harvard Egyetem hallgatója 3 év felfüggesztett börtönbüntetést, 10 ezer dollár pénzbírságot és 400 óra közmunkát kapott "elismerésül".

2.4. SPECIÁLIS "VIRULENS" PROGRAMOK

Ezek a "**programok**" nem sorolhatók a vírusok és a férgek közé, mert szaporodni nem tudnak, ugyanakkor a trójai programok közé sem, mert nincs egy külső burok, ami elrejti a valóságot, vagyis azt, hogy hasznos programunk kártékony lehet.

2.4.1. ASCII vírusok

Minden számítógép Neumann-elv szerint működik, így a gép az adatokat és programokat nem tudja megkülönböztetni. Az adatok és programok tárolása az operációs rendszer feladata, ami szintén nem különbözteti meg a **futtatható és az adatfile**-okat: egy futtatható programot ki is nyomtathatunk, ennek az eredménye egy csomó értelmetlen zagyvalék lesz, de ez az átalakítás ellenkező irányban is megengedett, egy szövegfile-t átnevezhetünk végrehajtható kiterjesztésűre, majd elindíthatjuk. Ilyenkor a gép semmi értelmeset nem fog csinálni, valószínűleg lefagy, mivel számára a szövegfile olyan érthetetlen, mint számunkra a futtatható file kinyomtatott listája.

Ám lehetséges, hogy a futtatható kód és az adatok közötti ilyenfajta átalakítás következtében a gép számára értelmes kód lesz az eredmény. Ez az a lehetőség, ami megengedi Ascii vírusok létezését. Az **ASCII** vírus nem más, mint maga a víruskód szövegfile-ban, ami nem baj, hiszen egy *szövegfile-t nem tudunk elindítani, ebből programot kell csinálni.* Ez háromféleképpen tehető meg:

- **Szöveges file átnevezésével**

A szövegfile **kiterjesztését kell végrehajtható kiterjesztésűre** (.com, .exe, .bat) változtatni, majd elindítani. Természetesen ezt csak úgy a felhasználó nem fogja megcsinálni, éppen **ezért kell egy batch file vagy egy program, ami ezt elvégzi.**

Nagyon sok felhasználó mindenféle elővigyázatosság nélkül szokott elindítani különböző batch file-okat, anélkül, hogy megnézné ezek tartalmát. Pedig néha érdemes: vegyünk egy közismert és közkedvelt játékot, ehhez szokott lenni egy batch file, ami elindítja a programot (rendszerint runme.bat szokott lenni a neve). Ha ez a batch file nemcsak a játékprogramot indítja el, hanem elvégez egy ilyen átalakítást, akkor a felhasználó nagyon rosszul járhat!

Sajnos a **DOS** lehetőséget ad arra is, hogy ne lássuk, milyen parancsok hajtódnak végre a **batch file**-on belül: ehhez elég, ha **@echo off** parancs van beírva a batch program elejére vagy **minden egyes parancs előtt egy @ jel** található.

Természetesen egy ilyen szöveges-bináris átalakítás megoldható egy "**bináris**" programból is, bár ennek jóval kevesebb az értelme, hiszen ha a felhasználó ellenőrzés nélkül futtat programokat, akkor eleve nagy az esélye annak, hogy egy vírus bejusson a rendszerbe.

- **Szövegfile másolásával**

Egy szövegfile-ban lévő vírus akkor is elindítható, ha a szövegfile tartalmát átmásoljuk egy végrehajtható file-ba és ezt a file-t elindítjuk. Ez kétféleképpen tehető meg (xxx egy futtatható kiterjesztést takar):

```
Type virus.txt > virus.xxx
```

vagy

```
Copy virus.txt virus.xxx
```

Természetesen elég feltűnő lehet a felhasználónak, ha a batch program csak úgy elkezdene másolgatni bizonyos szövegfile-okat. Ezt onnan venné észre, hogy a második esetben a copy a következő üzenetet adná vissza: "1 file(s) copied", ami eltüntethető, ha át van irányítva a kiíratás a képernyőről a "Null perifériá"-ra a következőképpen:

```
Copy virus.txt virus.xxx > nul
```

Ekkor már másolás esetén sem kapunk visszajelzést!

- **Debug bemeneteként szöveges file felhasználásával**

Általában feltételezhető, hogy a Debug.com program – ami egy elég kezdetleges debugger – az elérési útban, rendszerint a **DOS** alkönyvtárban található. A Debug-nak sok lehetősége van: pl. képes file-ba írni, file-ból olvasni, egy programrészletet adott címtől elindítani, módosítani stb.
Ebben az esetben egyetlen parancsot elég a batch file-ba beletenni:

```
Debug < virus.txt
```

Itt a virus.txt egy "sima" ascii file, ami a Debug számára értelmezhető parancsokat tartalmaz. Ekkor elegendő, hogy a szövegfile-ban lévő vírust a virus.txt-ben lévő Debug parancsok betöltsék egy adott memóriacímre és ott elindítsák azt. Ha a vírusprogram .com típusú, akkor még meg sem kell adni, hogy mely memóriacímre kell betölteni, mivel a Debug a nem .exe file-okat mindig a hexa 100-as címre tölti be, ahonnan a .com programok szoktak indulni.

Ezek után már csak az a kérdés, hogy a víruskód lefutása után hogyan szaporodik tovább a vírus, ha nem szaporodik, akkor mit művelhet. Itt több eset is lehetséges:

1. "Szabványos" vírusként fog továbbszaporodni

Az ascii forma csak a rendszerbe való bejutást tette könnyebbé. Ez azt az előnyt használja ki, hogy szinte minden víruskereső csak a végrehajtható állományokat ellenőrzi, az egyéb file-okat nem, így a specifikus védelmi módszert máris sikerült kijátszani.

2. Trójaiként viselkedik

Rendszerünket az elinduló kód azonnal szétzilálja, pl. formattálja a winchestert, törli a rendszerfile-okat, stb. Ebben az esetben nem beszélhetünk szaporodásról és ilyenkor vírusnak sem illene nevezni. A lehető leggyakrabban ez szokott előfordulni.

3. Ascii vírusként fog továbbszaporodni

Létrehoz újabb – valószínűleg – rejtett állományokat, amelyekbe beleteszi önmagát, majd egyéb batch programokhoz hozzácsapja az áttranszformálási rutint. Ez a lehető legritkább szaporodási forma.

Gyakori trükk, hogy olyan állományok tartalmazzák az ascii vírust, amelyeknek kiterjesztése nem kelt semmiféle feltűnést. Ha mondjuk egy tömörítő, pl. a Pkzip féle .zip kiterjesztést használ a vírus, akkor az ember tekintete "elsiklik" felette, hogy az az állomány vajon hogy került oda. Természetesen ilyenkor valódi tömörített állományról szó sincs.

Más kiterjesztés használatnak másik előnye az, hogy az antivírus programok elsiklanak felette, mivel azok csak a végrehajtható állományokban keresnek vírusokat, hacsak más opciót nem használunk. Ugyanis az antivírus programok 99%-ának van olyan paramétere, amelynek hatására minden állományt megvizsgál, még a szövegesnek feltüntetetteket is.

2.4.2. ANSI vírusok

Az első PC-k jellemző operációs rendszere a 16 bites, grafikus felülettel nem rendelkező **DOS** volt. A grafikus felület hiánya miatt a Dos-hoz több kiegészítést is készítettek, többek között ún. ansi parancsok feldolgozására

képes meghajtóprogramot az Ansi.sys-t, ami többek között a **DOS** képernyőkezelését bővítette, de csak karakteres felületen. A kibővített utasítások segítségével – un. escape szekvenciák képernyőre való kiíratásával – a felhasználóknak lehetősége nyílt kurzorvezérlésre, képernyőtörlésre, videomód megváltoztatásra és billentyűzet átdefiniálásra. Ez utóbbi lehetőséget használják ki az ansi vírusok.

Ha ez a meghajtóprogram a Config.sys-ből be van töltve, akkor a szabványos kimenetre – képernyőre – történő escape szekvencia kiíratáson keresztül lehet kihasználni az Ansi.sys parancsait. Egy ilyen escape szekvencia kiíratás többféleképpen is történhet:

1. Echo paranccsal

Az echo parancs a Dos-ban szövegek képernyőre való kiírására szolgál – igaz át is lehet irányítani a kimenetét – és alkalmas escape szekvenciák kiíratására is, így ansi parancsok végrehajtására is.

2. Prompt paranccsal

A prompt parancs a **DOS** promptjának beállítására alkalmas, de rendelkezik escape szekvenciák kiírására szolgáló paraméterrel is, vagyis ez is alkalmas ansi parancsok végrehajtására.

3. Szövegfile type-olásával

Az Ansi.sys parancsait szöveges file-okban is el lehet helyezni: ha ezt az állományt type paranccsal kiíratjuk, akkor a file-ban található ansi parancsok végrehajtódnak.

4. Programból való kiírással

Egy "bináris" program is képes ansi vezérlésre, ha a **DOS** szabványos kimenetre való kiírási rutinját használja.

Mint fentebb már említettem, az ansi vírusok az Ansi.sys billentyűzet átdefiniálási lehetőségét aknázzák ki. Billentyűzet átdefiniálással meg lehet oldani, hogy más karakter jelenjen meg a képernyőn, mint amit leütött a felhasználó, valamint tetszőleges billentyűhöz bármilyen parancsot vagy futtatható programot lehet hozzárendelni. Pl. ha az F10-es funkcióbillentyűhöz egy 'Dir' parancsot rendelünk, akkor az átdefiniálás után akárhányszor lenyomjuk az F10-es billentyűt, az aktuális könyvtár directory listáját fogjuk megkapni.

Ezt a billentyűzet átdefiniálását azonban rossz szándékkal is ki lehet használni, amit meg is tesznek az ansi vírusok készítői: általában sűrűn használt billentyűkhöz romboló utasításokat rendelnek. Ebből adódóan az ansi vírusok nem szaporodóképességükről híresek, hanem inkább pusztításaikról, éppen ezért talán érthető is, hogy a szakemberek többsége az ansi-bomba kifejezést használja ezekre a "programokra".

Gondoljunk csak bele, milyen "fenomenális" érzés lehet az, ha az Enter billentyűhöz egy 'Format c: /u /autotest' parancs van rendelve.... Egyszerűen azt vesszük észre, hogy egy teljesen üres winchesterünk van, aminek a tartalmát még visszaállítani sem lehet. A **DOS** régebbi változataiban használható 'format'-nál még nem volt 'autotest' paraméter, de ez is kiküszöbölhető, ha a 'format' parancsnak egy szöveges állomány vagy az echo parancs kimenete van használva bemenetként.

Sajnos nem kell messze menni a másik, **DOS** parancsot alkalmazó hozzárendelésig sem: pl. 'echo Y | del *. *' paranccsal a gép nem kér engedélyt a file-ok törlésére.

Természetesen az is lehetséges, hogy a rombolást nem a **DOS** parancsaival végzi el: egy bináris állomány is tartozhat a bombához – ami szintén ugyanúgy aktivizálható, mint a fent leírt Format parancs – mely esetleg "csak" a Fat táblát, vagy a partíciós táblát teszi tönkre. Egy ilyen bináris program a vírusok objektív rutin szerinti kategorizálásánál említett bármilyen szoftveres vagy hardveres pusztítást véghez vihet.

Figyelembe véve az escape szekvenciák kiíratási lehetőségeit, ansi parancsok az alábbi állományokban lehetségesek:

1. Batch file

Az ilyen típusú állományokból 'prompt' vagy az '**echo**' parancs segítségével lehet elérni billentyűzet átdefiniálást. Sajnos nagyon sokan vannak olyanok, akik az ilyen billentyűzet átdefiniáló batch állományokat batch vírusoknak titulálják, csak egy valamit nem vesznek figyelembe: itt nincs szó szaporodásról. **A szaporodni képes batch vírusok a file vírusokon belül egy külön kategóriát alkotnak!**

2. Szöveges állomány

Régebben a programokhoz adott leírásokat nem egy külön állománynézegető program segítségével olvasták el a felhasználók, hanem a type parancs segítségével kiíratták a képernyőre, esetleg a more-t használva tördelték képernyőnkénti méretre. Ebből viszont rögtön adódott is a támadási le-

hetőség: a szöveges rész bármely részén – inkább az elején, mert azt jellemzően elolvassák a felhasználók – elhelyezett escape szekvencia segítségével át lehetett definiálni a billentyűzetet.

3. Végrehajtható program

Egy program ansi vezérlő karaktereket is kiírhat a képernyőre, tehát billentyűzetet is átdefiniálhat az Ansi.sys segítségével. Egy fontos megjegyzésem lenne, hogy programból nemcsak az Ansi.sys segítségével lehet billentyűzetet átprogramozni.

4. Tömörített állomány

Tömörített állomány kibontásakor a type rutinon keresztül kiíródó fejcímet is el lehet látni ansi kóddal.

Ha batch file-ból vagy szöveges állományból történik a billentyűzet átdefiniálás, akkor az esetek többségében pusztító utasításnak a **DOS** valamely parancsát használják fel, a másik két esetben lehetséges, hogy a program által létrehozott egyéb végrehajtható program illetve egy kitömörített program a pusztítást végző "bombarész".

Mint látható, sokféleképpen lehet elérni a billentyűzet átdefiniálását, egy megoldási módot azonban nem említettem feljebb, ez pedig maga a **Dir** parancs. Hónapról hónapra különböző Internet-es fórumokon felbukkan az a kérdés, hogy egy **Dir** paranccsal aktivizálni lehet-e egy vírust. Vírust nem, de ansi bombát igen: minden lemezen vannak olyan területek, amelyekben a könyvtár tartalmának adatait tárolják, pl. a könyvtárban lévő file-ok nevei, kiterjesztésük, létrehozási dátum, kezdő clusterszám stb. Ha azonban ez a terület nincsen teljesen kihasználva – nincs az adott könyvtár tele file-okkal, akkor az üres részekre lehet tenni ansi parancsokat.... Amikor directory listát kérünk egy lemezről, akkor azok az adatok kerülnek a képernyőre – a felhasználó számára érthető formában – amelyek fileneveket és a file-ok egyéb adatait tartalmazzák. Az ANSI parancsokat tartalmazó byte-ok azonban nem kerülnek a képernyőre, mert azokat az ansi driver "lenyeli". Vagyis el lehet érni, hogy egy DIR parancs hatására néhány billentyű átdefiniálódjon! Persze ehhez az kell, hogy egy idegen floppyról kérjünk listát, mivel a saját winchesterünk megfelelő részeibe csak nem teszünk ilyen parancsokat, hacsak valamilyen trójai nem csinál ilyet, másrészt ma már nem jellemző, hogy valaki **DIR** paranccsal nézi meg a lemez tartalmát, inkább használ valamilyen filemanager programot, pl. a Norton Commander-t.

Az ansi bombák ellen nem nehéz védekezni: ha nincs betöltve az **Ansi.sys**, akkor a bomba nem képes "robbanni", mivel nincs ami végrehajtsa a rombolást eredményező billentyűzet átdefiniálási utasításokat. Ebből rögtön adódik a védekezés egy módja: nem töltjük be a Config.sys-ben az Ansi.sys-t. Ha azonban mégis ragaszkodunk az Ansi.sys-hez, akkor próbáljunk beszerezni olyan "lebutított" **Ansi.sys**-t, ami nem teszi lehetővé ezeknek a kevésbé publikált parancsoknak a végrehajtását. A másik lehetőség az, hogy Philip Katznak, a Pkzip írójának programját, a Pkfansi-t beszerezzük, ami a billentyűzethez való hozzárendelést tiltja le, vagy pedig megvásárolunk egy komolyabb vírusvédelmi rendszert, ami más, közismert vírusok ellen is használható.

Most már csak az a kérdés maradt hátra, hogy honnan "szerezhető" ilyen bomba, esetleg hogyan szaporodik. Nem szaporodik, tehát csak úgy kaphatunk ansi bombát, ha valamilyen programcsomagok másolgatásakor egy bombát is tartalmazó file "keveredik" a file-ok közé. Persze ez a többi állomány közé keveredés szándékos. A bomba készítője a programcsomag részeként is terjesztheti az ansi "vírust". A többi már csak a felhasználó elővigyázatlanságán múlik...

2.5. VÍRUSTÁROLÓ PROGRAMOK

Ezek a programok valamilyen úton-módon vírust tárolnak önmagukban, más-más céllal, de több jellemzőben is megegyeznek:

- egyikük sem fertőzött, csak hordoznak bizonyos vírusokat,
- régebben egy-egy nagyobb vírusgyűjteményben többet is lehetett találni, ma már nem jellemzőek.

2.5.1. Injector

Az injector programok sem megfertőzött programok, csak maga a programkód tartalmazza a vírust, melyet ha valaki elindít, akkor annak a gépét "beoltja" a vírussal. Bár hasonlít a beépített vírust tartalmazó trójai programokhoz, azoktól az különbözteti meg, hogy a trójai programoknál van valamiféle esemény vagy feltétel, aminek hatására a trójai szabadjára engedi a hordozott vírust, itt pedig nincs. Tehát ha egy gyanútlan felhasználó elindít egy ilyen programot, akkor megfertőzte gépét. Nehezen detektálhatóak.

2.5.2. Germ

Vannak olyan vírusok, amelyek "normális" úton nem képesek szaporodni, pl. a vírus csak egy adott méret feletti programokra képes rámászni. De ha valakinek meg van a vírus forráskódja vagy egy jól visszafejtett listája, akkor könnyen szerkeszthet hozzá egy kisebb hordozó programot, aminek segítségével a vírus aktivizálni tudja magát, ha azt elindítják. Ezeket a programokat (a hordozott vírussal együtt) nevezzük germ programoknak. Rendszerint Com állományokban találhatóak.

2.5.3. Dropper

A dropper programok többnyire 360 Kbyte-os floppylemezekre installálnak boot-vírust, rendszerint a felhasználó beleegyezésével. Ez azt jelenti, hogy indításuk után a felhasználót figyelmeztetik, hogy vírust tesznek fel az "A:" meghajtóban található floppylemezre és megkérdezik azt is, hogy biztos-e a felhasználó a vírus installálásában. Amennyiben nem, a program nem telepíti a vírust és nem is csinál semmiféle destruktív dolgot. Ha a felhasználó kéri, akkor boot-vírust telepít a behelyezett floppyra, de winchesterre sohasem (igaz, ha a legközelebbi boot-olási próbálkozás a fertőzött floppyról történik, akkor könnyen meg lehet fertőzni a merevlemezt is!)

2.6. VÍRUSGENERÁTOR PROGRAMOK

A vírusírás automatizálása a 90-es évek elején kezdődött, az első vírusgenerátor program a "Virus Construction Set" nevet viselte. Erre a programra jellemző volt, hogy olyan vírusokat készített, melyek kódjaikban nem sok mindenben különböztek egymástól, csak más-más dátumokon a felhasználó által megadott szöveget írták ki a képernyőre és a Config.sys-t, valamint az Autoexec.bat-t tették tönkre.

A későbbiek folyamán már jóval komolyabb vírusíró automaták (Virus Creation Laboratory, IVP, G2, Virus Tutorial, Generateur Virus, PS-MPC) is kikerültek az alvilági műhelyekből, de a generátorok működése szinte mindig azonos volt: adva volt egy vírusváz, amire a generátor a felhasználó igényeinek megfelelően rárakta a megfelelő rutinokat.

Ezeknek az automatáknak a megjelenése nagy veszélyt jelentett, hiszen a vírusírásban avatatlanok is képesek voltak olyan vírusok előállítására, amelyek az "átlag" felhasználók számára komoly fejfájást okoztak. Érdekes

módon azonban kevésbé terjedtek el. Egyrészt mert mind a mai napig a vírusírók legfőbb hajtóereje a kihívás, az erőfitogtatás, ami az automaták alkalmazása révén elvész, másrészt az antivírus programfejlesztők is rájöttek arra, hogy ha a vírusvázra aggatott rutinok állandóak, akkor könnyen felfedezhetőek az adott automata által készített vírusok, hiszen csak a megfelelő rutinokat kell keresni a gyanúsnak vélt állományokban.

Ezek után természetesen megint a vírusfejlesztők léptek egyet előre, hiszen a víruskód állandó változásának érdekében kifejlesztették a mutációs rutinokat...

Modern korunkra jellemző, hogy már megjelentek a makróvírusgenerátor programok vagy makrók is, melyek hasonlóan a bináris vírusgenerátor programokhoz makróvírusok sokaságát képesek legyártani rövid idő alatt.

2.7. VÍRUSMUTÁCIÓS RENDSZEREK

A számítástechnika alvilágában több olyan programrendszer is létezik, amelyek lehetővé teszik assembly rutinok megváltoztatását, mutálását. A vírusmutációnak az a lényege, hogy a vírus minél jobban változzon, hiszen ekkor kevésbé lehet felfedezni, leirtani.

Ezek a rendszerek két részre oszthatók attól függően, hogy a mutáció készítő rendszer önálló program, vagy csak egy rutin, ami egy vírus forráskódjához is csatolható.

2.7.1. Vírusmutáló programok

A vírusmutáló programok a bemenetként megadott forráskódot vagy egy tökéletesen visszafejtett kódot képesek megváltoztatni annyira, hogy még a forráskód eredeti szerzője sem mindig ismer rá. Az "új" program is ugyanazt a feladatot fogja elvégezni, mint az eredeti, de binárisan egészen másképp néz ki mint az eredeti. Az ilyen programokat azonban nemcsak vírusok mutálására lehetett felhasználni: ha ugyanis valakinek forrásszinten is sikerült valamilyen programot ellopnia, akkor elegendő volt egy ilyen mutáló programot felhasználnia és máris nem hasonlított a nem törvényes úton szerzett forrásprogram és az újonnan "kifejlesztett" program.

A legelső és leghíresebb vírusmutáló program, a "Virus Mutator" 1992-ben jelent meg. Az egész programcsomag egy 5525 byte hosszúságú összetömörített file-ban (Mutator.arj) volt megtalálható. Az arj-s állományban ta-

lálható Mutator.exe program segítségével lehetett elvégezni az assembly kód mutálását. Futtatásakor meg kellett adni a bemeneti és kimeneti forrásfile nevét, valamint a mutálásához szükséges néhány paramétert:

Σ

- a forráskódot hányadik sorától kezdje el mutálni,
- a forráskód hányadik soráig végezze tevékenységét,
- milyen lépésközzel mutálja,
- hány új sort tegyen bele.

Ez a program lépésköznyi, egymás utáni NOP-okat (semleges utasítások assembly-ben) szórt el a forráskódban a beolvasott sorok között, valamint a kért számú új sort szúrta be, mely szintén hatástalan, akárcsak az egymás utáni NOP-ok. A célfile nagyobb volt, mint az eredeti, mivel NOP-ok és új sorok lettek beleszúrva, viszont ezáltal a végeredmény olyan kód volt, amely minimálisan hasonlított az eredeti kódra.

2.7.2. Vírusmutációs engine-ek

A mutációs rutinok biztosítják, hogy az a program, amelybe beszerkesztik, folyamatosan változzon. Ezeket a rutinokat angolul mutating engine-eknek hívják, azokat a vírusokat, amelyekben ilyen "mutációs motor" van, mutációs motor alapú vírusoknak nevezzük, pl. MtE alapú vírus. Bár az ilyen rutinok hasznos programokba is beleszerkeszthetőek, ennek értelmét csak akkor látom, ha a program írója azt akarja, hogy nehezen tudják csak visszafejteni programját.

A legelső ilyen szörnyszülemény a "Dark Avenger Mutation Engine" (MtE) 1992 tavaszán jelent meg. Jóval több hozzáértést igényel, mint az előző pontban említett "Virus Mutator" program, erre a dokumentációban az írók fel is hívják mindenkinek a figyelmét. Ha ez a Mutation Engine – ami valójában egy object modul – bele van szerkesztve a mutálási képességgel felruházni kívánt kódba, akkor megfelelő paraméterezéssel a regisztereken keresztül csak egy külső szubrutint kell meghívni a programnak vagy a vírusnak, ezután a programkód mindig más és más kulccsal lesz elkódolva.

Az MtE után sorra jelentek meg a különböző mutációs motorok: TridenT Polymorphic Engine (TPE), Nuke Encryption Device (NED), Dark Slayer Mutation Engine (DSME), Dark Angel Mutating Engine (DAME), Mark Ludwig Visible Mutating Engine (VME), Simulated Metamorphic Encryption Generator (SMEG) stb., sőt nemrégen megjelent az első makróvírus mutációs motor is!!

Mennyire kell komolyan venni a számítógépvírusokat és egyéb kártékony programokat? Erre a kérdésre válaszként álljon itt egy összehasonlítás: egy 1992-es amerikai felmérés szerint a megkérdezett cégek és szervezetek 5 százaléka szenvedett el valamilyen súlyos víruscsapást, a fertőzések nyomán keletkező kár pedig elérte az évi egymillárd dollárt. Egy 1997-es felmérés szerint már 99.33%-uk átesett számítógépes vírusfertőzésen... Hadd ne számoljam ki a 1997-es károkat, de ezenkívül lenne még egy megjegyzésem: csak amerikai cégek körében történt a felmérés...

Most, hogy ennek a fejezetnek a végére értem, jóslatokba kellene bocsátkoznom, hogy merre fog tartani a vírusok fejlődése, de nem teszem. Véleményem szerint ugyanis nincs az az antivírus szakember, aki előre tudná, hogy mi lesz a következő lépés. Ezt nem ők fogják eldönteni, hanem a vírusfejlesztők! Ők azok, akik meghatározzák a "fejlődés" irányát, egészen addig, amíg a PC-kre is olyan operációs rendszerek és alkalmazások jelennek meg (és széles körben terjednek el), melyek kevésbé teszik lehetővé vírusok létezését.

3. KÜLÖNLEGES VÍRUSTÍPUSOK

A „PC vírus tipológia" című fejezetben a generáció szerinti vírusfajtáknál nem jellemeztem bővebben az egyes típusokat, melynek oka az, hogy a róluk szóló anyag nagyon terjedelmes: ha mindent leírtam volna róluk előzőleg, akkor a tipológiai rész csak ezekről a vírusokról szólt volna és nem általánosan a kártékony programok tipológiájáról.
Ebben a fejezetben **a mutációs, a lopakodó, a companion, fat és a batch vírusokról** lehet találni érdekes dolgokat, a **makróvírusoknak** a következő fejezetet szántam.

3.1. POLIMORF, MUTÁCIÓS VÍRUSOK

A vírusírók hamar rájöttek arra, hogy ha a vírus saját kódjának nagy részét kódolja, akkor nehezebben lehet belőle olyan byte-sorozatot kiválasztani, ami alapján a vírus megkereshető. Persze ahhoz, hogy a vírus végrehajtódjon a legközelebbi fertőzött program indításakor, vissza kell kódolnia magát. Ezt a kikódolást intézi a kikódoló rutin, amit angolul „**decryptor**"-nak nevezünk. Egy ilyen technikát alkalmazó vírusban legelőször mindig a decryptor fog lefutni, ami kikódolja a vírus többi részét, majd a kikódolt részek rutinjai ezután fognak lefutni. Fertőzéskor a vírus a decryptor rutin kivételével elkódolja magát, majd ezután történik a titkosított víruskód beleírása a célpontba. Ugye milyen egyszerű?

A **polimorf** vírusokban ezt a gonosz technikát úgy fejlesztették tovább, hogy decryptor részt a vírus állandóan átírja úgy, hogy az még mindig a vírus többi részének a kikódolását végzi, de állandóan másképp néz ki binárisan minden egyes fertőzésnél. Ez a decryptor változtató rutin a vírus elkódolt részében található, mindig akkor fut le, amikor a vírus egy újabb célpontot kíván megfertőzni, ezért változik a polimorf vírusok decryptor része fertőzésenként. Ezek a vírusok önmaguk kódját szinte határtalanul képesek módosítani, változtatni.

A polimorf vírusok tehát valójában leutánozzák a biológiai vírusokat, hiszen azok is mindig változnak és így tudnak újra és újra támadni: pl. az influenza vírus is mindig átalakul, ezért az emberi immunrendszer a megváltozott vírust nehezebben tudja kiszűrni. Egy ilyen biológiai vírus mindenféle mesterséges beavatkozás nélkül képes megváltozni, ugyanígy cselek-

szenek a mutációs vírusok, amelyek – miután szabadon bocsátotta az írójuk – maguktól változnak. Néhány kutató talán emiatt is a vírusokat mind a mai napig a mesterséges intelligenciák körébe sorolja.

A víruskereső programok mindig egy bizonyos vírusra **jellemző kódrészletet keresnek** a file-okban. Ha ilyet találnak, akkor az a file nagy valószínűség szerint fertőzött. A polimorf vírusoknál viszont éppen az állandó változás miatt nem lehet a vírusból ilyen kódrészletet választani, hiszen a **decryptor mindig változik, a vírus többi része pedig kódolt.**

A polimorfizmus több lépésben fejlődött ki, ezek közül csak az elsőnek adott nevet a szakirodalom. Nem minden polimorf vírus rendelkezik az összes polimorf technikával mégsem, egynémely vírusok „megragadtak" egy bizonyos szinten. Néhány lépésnél néhány általam kitalált és lehetségesnek tartott gyakorlati példát is leírok. Persze a felsorolt példák korántsem jelentik azt, hogy csak ezek léteznek, csupáncsak azoknak szeretném megmutatni egy vírus lehetséges átváltozását, „életét", akik eddig nem hittek benne.

1. lépés: oligomorf vírusok

Az ilyen típusú vírusok voltak az elsők, amelyek valamilyen újdonságot mutattak fel az antivírus – vírus fronton. Ebben az esetben a decryptor állandó, de a víruskód többi része mindig más és más kulccsal van elkódolva. Ezek a vírusok nagyon könnyen detektálhatóak, hiszen a kikódoló rutint vagy annak egy részletét kell csak keresni a megfertőzhető célpontokban. A legtöbb víruskutató nem is sorolja az ilyen vírusokat a polimorf vírusok közé. Ilyen vírus volt az *1701-Cascade, vagy közismertebb nevén potyogtatós vírus.*

2. lépés

A következő lépésben a decryptor még mindig ugyanazokból az utasításokból áll, csak ezeknek az utasításoknak a végrehajtása már megváltozik minden egyes fertőzéskor: pl. más regiszterek vagy adott utasítás más opcode-ja van használva a következő futáskor.

Pl.	Hexakód	Utasítás
	89D8	MOV AX,BX
	8BC3	MOV AX,BX

A MOV AX,BX-nek kétféle hexakódja is létezik, de ez néhány egyéb utasításra is igaz!

Ezzel a technikával támadó polimorf vírusok is könnyen detektálhatóak, hiszen csak a byte-mintáknak a számát kell növelni a víruskereső programon belül. De minderről bővebben az 5. fejezetben.

3. lépés

Itt a mutációs rutin a decryptor-ban valamilyen utasítás(oka)t cserél le másik, de eredményében ugyanolyan utasítás(ok)ra.

1. Gyakran előfordul az, hogy az egyik regiszter tartalmát szeretnénk átvinni a másik regiszterbe, ez regisztertípustól függően különbözőképpen oldható meg. Az egyik ilyen 3. lépésbeli mutációs trükk a regiszterek közötti adatátvitel verem segítségével.

Pl.	Hexakód	Utasítás
	8CC8	MOV AX,CS
	8ED8	MOV DS,AX
helyette:	0E	PUSH CS
	1F	POP DS
vagy:	0E	PUSH CS
	58	POP AX
	50	PUSH AX
	1F	POP DS

Pl.	Hexakód	Utasítás
	89D8	MOV AX,BX
helyette:	53	PUSH BX
	58	POP AX

Ezeknek a utasításoknak ugyanaz lesz a hatása, mint a MOV-nak: a „cél"-regiszter tartalma felveszi az új értéket, a „forrás"-regiszter tartalma sem változik. Az 1. példa 1. mutációjában a kód hossza csökken (!!), a 2. mutációjában vannak fölösleges utasítások is (**POP AX, PUSH AX**), melyekkel a mutációs lehetőségek száma tovább növekedik.

2. A regiszter kinullázást is többféleképpen tudjuk megtenni. Ha ezeket a módokat cserélgeti a vírus, akkor is folyamatosan változni fog a kód.

Pl.	Hexakód	Utasítás
	31C0	XOR AX,AX
helyette:	29C0	SUB AX,AX
vagy:	B80000	MOV AX,0000
vagy:	B80000	MOV AX,0000
	89C3	MOV BX,AX

Szeretném megjegyezni, hogy a **SUB** „lassabban" végzi munkáját, mint a **XOR**, a 4. kinullázásos módszernél az a lehetőség lett kihasználva, hogy egy nem használt regisztert nullázunk, majd abba a regiszterbe rakjuk a tartalmát, amit nullázni akartunk, viszont a regiszterek közötti átvitel a fent említett módon már mutálható...

3. Néha szükség van arra a címre, ahonnan az adatok kezdődnek és az adott címtől az egyes adatok eltolására. Ezt a kettőt (kezdőcím + eltolás) összeadva kapjuk a kívánt adat valós címét. A kezdőcímet általában egy külön regiszterben tároljuk, aminek tartalmát nem szabad változtatni.

Pl.	Hexakód	Utasítás
	89F2	MOV DX,SI
	81C21F00	ADD DX,001F
helyette:	BA1F00	MOV DX,001F
	03D6A	ADD DX,SI

Ebben az esetben SI tartalmazza a kezdőcímet, 001F az eltolás. Mint látható, a hossz változik, de az eredmény ugyanaz.

4. Más interrupt használattal is változhat a kód: más interruptot hívok meg, de ugyanazzal a hatással. Erre a lehető legjobb példa a **program rezidenssé** válásának átírása (**INT 21 és INT 27 csere**).

5. Interrupt hívás helyett utasítás is használható: amely interruptot meg akarjuk hívni INT utasítással, lekérdezhetjük a címét és CALL-al hívjuk meg a lekérdezett címet, miután minden szükséges regisztert feltöltöttünk.

6. Többféle lehetőség is van a MOV utasítás átírására:

Pl.	Hexakód	Utasítás
	BA1F00	MOV DX,001F
helyette:	BA0000	MOV DX,0000
	81C21F00	ADD DX,001F

Ebben a példában először kinulláztuk azt a regisztert, amelyikbe bele akarunk tenni valamit, majd egy **ADD** utasítással hozzáadjuk a betenni kívánt értéket. A nullázási módszerek közül az egyiket használtam, de bármelyik másik is betehető, ami megint növeli a mutációs lehetőségeket. Persze van egy „elegánsabb" **MOV** szétbontási lehetőség is:

Pl.	Hexakód	Utasítás
	BA1F00	MOV DX,001F
helyette:	B600	MOV DH,00
	B21F	MOV DL,1F

7. Ha egy vagy több regiszter tartalmára adott rutin lefutása után is szükség van, akkor rendszerint vermeléssel szokták megoldani. Ám ez megoldható másképpen is: az elmentendő regiszter tartalmát átrakjuk (**MOV**) olyan regiszterbe, amelyet nem használunk sem a rutin végrehajtása során, sem a programban (ha mégis használnánk a programban, akkor ezt el kell menteni valamilyen módon).

Pl.	Hexakód	Utasítások
	52	PUSH DX
	.	
	.	
	5A	POP DX

```
helyette:   89D0        MOV AX,DX
              .
              .
            89C2        MOV DX,AX
```

Itt éppen **DX** regiszter tartalmát kellett elmenteni, de más regiszter tartalma ugyanígy elmenthető.

4. lépés

A polimorfizmus negyedik fejlődési lépését lehet a legkönnyebben megérteni: véletlen helyekre véletlen számú semleges utasítás vagy utasítássorozatok vannak beszúrva a kódba, az utasítások közé. Ez(ek) a hatástalan utasítás(ok) szintén nem befolyásolják a decryptor algoritmust, de az eredmény mégis „óriási", hiszen nem vagy csak nagyon nehezen tudunk olyan byte-mintát venni a decryptorból, ami alapján kereshetnénk a vírust.

1. Assembly-ben hatástalan a **NOP** (**N**o **OP**eration = nincs művelet) parancs. Ha a program futása egy ilyen utasításra ér, akkor a program „nem csinál semmit", csak a futási ideje növekedik a másodperc töredékrészével. Éppen ezért nagyon „hatásos", mivel ha egyszerre több ilyen utasítást is elszórunk a programban és azt a fordító lefordítja, akkor a vírus bináris kódja már más lesz, a hossz is változik, a standard byte-minta szerinti víruskeresés már csődöt mondhat. Az esetleges hosszváltozás miatt a régebben írtható vírus már nem az lesz, így született a kitolós Potyogós vírus is, amit szinte minden antivírus program ki tudott irtani, ám „gondos" kezek **NOP**-os változtatása miatt az irtáskor maga a megtámadott program tönkremehet!

```
Pl.         Hexakód     Utasítás

            8BF2        MOV SI,DX
            81C60A00    ADD SI,000A
            BF0001      MOV DI,0100
            B90300      MOV CX,0003
            F3          REPZ
            A4          MOVSB

NOP-al:     8BF2        MOV SI,DX
            90          NOP
            81C60A00    ADD SI,000A
```

```
90          NOP
BF0001      MOV DI,0100
90          NOP
B90300      MOV CX,0003
F3          REPZ
A4          MOVSB
```

Mint a fenti példából látható, a vírus (hexa) kódja és a hossza megváltozott, de a funkciója nem. Amennyiben esetleg éppen ez a kód a vírus keresési mintája, akkor a minta szerinti víruskeresés már hatástalan lesz. Természetesen nemcsak egy **NOP**-ot lehet beszúrni a kódba, hanem esetleg többet is, a lényeg nem változik. Ugyanez a transzformáció visszafelé is eljátszható: a kód szintén megváltozik, ha a benne lévő **NOP**-ok „eltűnnek"!

2. Szintén hatástalan lehet az, ha egy olyan rutin van beszúrva a kódba, ami sem a regisztereket, sem az adatokat nem változtatja. Ilyen rutin lehet pl. ha egy regiszter tartalmát elvermeljük, majd kivermeljük, de fölösleges ugró utasítás beszúrása is lehetséges: közvetlenül az utána következő címre kell, hogy mutasson.

```
Pl.             Hexakód     Utasítás

                8BF2        MOV SI,DX
                81C60A00    ADD SI,000A
                BF0001      MOV DI,0100
                B90300      MOV CX,0003
                F3          REPZ
                A4          MOVSB

„Vermesen":     8BF2        MOV SI,DX
                81C60A00    ADD SI,000A
                50          PUSH AX
                58          POP AX
                BF0001      MOV DI,0100
                B90300      MOV CX,0003
                F3          REPZ
                A4          MOVSB
```

Fölösleges 2 byte-os ugróutasítással:

```
CS:0200  8BF2      MOV SI,DX
CS:0202  81C60A00  ADD SI,000A
CS:0206  EB00      JMP 0208
CS:0208  BF0001    MOV DI,0100
CS:020B  B90300    MOV CX,0003
CS:020E  F3        REPZ
CS:020F  A4        MOVSB
```

Fölösleges 3 byte-os ugróutasítással:

```
CS:0200  8BF2      MOV SI,DX
CS:0202  81C60A00  ADD SI,000A
CS:0206  E90000    JMP 0209
CS:0209  BF0001    MOV DI,0100
CS:020C  B90300    MOV CX,0003
CS:020F  F3        REPZ
CS:0210  A4        MOVSB
```

A három említett példán kívül, még jó néhány olyan rutin írható és beszúrható, melyek a regiszterekre és az adatokra nézve semleges hatásúak (pl. **MOV BX,BX**).

5. lépés

A mutációs rutinok fejlesztői rájöttek arra, hogy a decryptorban azokat az utasításokat fel lehet cserélni, amelyek egymásra nincsenek hatással, nem ugyanazzal az adattal, vagy regiszterrel dolgoznak, hiszen ezáltal a kód funkciója nem változik, de binárisan annál inkább.

1. Kód változásával jár, ha a kódban már meglévő **NOP**-okat az alattuk vagy a felettük levő utasításokkal cseréljük.

2. A lehető legjobban az egymást követő **MOV** utasítások cserélhetők fel egymással.

```
Pl.      Hexakód   Utasítás

         B8C103    MOV AX,03C1
         BB0101    MOV BX,0101
```

helyette:	BB0101	MOV BX,0101
	B8C103	MOV AX,03C1

Pl.	Hexakód	Utasítás
	B90300	MOV CX,0003
	B44F	MOV AH,4E
	BA1F00	MOV DX,001F
	CD21	INT 21

A második példában azt szerettem volna megmutatni, hogy az interruptok előtt is felcserélhető az egyes regiszterekbe való adatbevitel sorrendje, ráadásul itt a 3 **MOV** utasítást 6 különböző sorrendben tudnánk leírni (itt most egy mutációs példát sem írtam le, de az első példa alapján szerintem senkinek nem okozna nehézséget).

3. Vermelésnél hasonlóképpen felcserélhető a sorrend.

Pl.	Hexakód	Utasítás
	56	PUSH SI
	57	PUSH DI
	52	PUSH DX
	.	
	.	
	5A	POP DX
	5F	POP DI
	5E	POP SI

helyette:	57	PUSH DI
	52	PUSH DX
	56	PUSH SI
	.	
	.	
	5E	POP SI
	5A	POP DX
	5F	POP DI

A MOV-os példához hasonlóan itt is hatféleképpen lehetne megoldani a vermelést.

Bár könyvem első kiadásában említettem egy 6. lépést – amiben az összes előző lépés összeérett, most ezt inkább nem tenném, ugyanis nem jelenthetjük ki egyértelműen, hogy a polimorf vírusok csúcstípusai közé tartozó vírusok közül valamennyi alkalmazná a fentebb leírt 5 lépést: pl. 2. lépés kihagyásával is lehet nagyon komoly számú mutációra képes vírust írni!

További érdekességképpen megemlíteném, hogy a **polimorf vírusok egy véletlenszám-generátort is tartalmaznak.** Ezek a vírusok a generátor által képzett véletlen számtól függően választják ki a decryptornak azt a részét, amit átmutálnak, ezenkívül a címzési mód és a használt regiszterek kiválasztásánál is felhasználják a generált véletlenszámot, valamint ezzel a számmal kódolják a decryptoron kívüli víruskódot.

+1 lépés: permutáció

Egy különleges polimorf vírustípus is létezik, a permutációs vírusok. Ezek a vírusok nem titkosítják magukat – így könnyű belőlük byte-mintát venni, viszont képesek cserélgetni a víruskódon belüli főbb kódrészeket. Ez viszont régebbi antivírus programoknak alapvető problémát okozhat!

Például tegyük fel, hogy **egy 3 főbb rutinját cserélni képes permutációs vírussal** van dolgunk. A rutinok kezdeti sorrendje:

rutin 1
rutin 2
rutin 3

Lehetséges cserélések:

1.2.3.4.5.

rutin 1	rutin 2	rutin 2	rutin 3	rutin 3
rutin 3	rutin 3	rutin 1	rutin 1	rutin 2
rutin 2	rutin 1	rutin 3	rutin 2	rutin 1

A permutációs vírusok innen kapták nevüket, mivel a vírus a cserélhető rutinok számától függően változhat. A víruson belül az egyes rutinok közötti kapcsolat ugróutasításokkal van megoldva.

A Magyarországon is megjelent **One-half vírus ezt a permutációs technikát alkalmazta**, azzal kombinálva, hogy az egyes rutinokat a megtámadott programban véletlenszerű helyekre tette.

Az eddig megjelent polimorf típusú vírusok file-vírusok és makróvírusok voltak, ez utóbbinak a mutációs lehetőségei azonban egy kicsit eltérnek a fentebb leírt „bináris" vírusok mutációs lehetőségeitől, de erről bővebben a makróvírusokról szóló fejezetben lehet olvasni.

Az itt felsorolt mutálási technikák a vírusátíráshoz képest bocsánatos bűnre is felhasználhatóak: ha például ellopunk egy assembly rutint vagy programot (ez azért meggondolandó a mai **BSA**-s világban) és átmutáljuk, akkor még a szerzője sem fog ráismerni. Sajnos erre elég kevesen használják tudásukat, inkább vírusokat írogatnak át, megkeserítve ezzel az egyszerű felhasználók életét.

3.2. LOPAKODÓ VÍRUSOK

Az első generációs vírusok nem úgy lettek kifejlesztve, hogy a rendszerben a vírus miatt végbemenő változásokat a felhasználó ne vegye észre. Erre igazság szerint nem is volt nagyon szükség, mivel a felhasználók a vírusok fogalmával, a lehetséges támadási pontokkal és az ellenük való védekezéssel sem voltak tisztában. Később azonban egyre több olyan szoftvertermék jelent meg, amely futtatásakor ellenőrizte saját épségét, eltérés (vírusfertőzés) esetén a felhasználót figyelmeztette. Erre válaszul a víruskészítők kifejlesztették lopakodó vírusaikat: ezek a vírusok mindig elrejtik a fertőzés során bekövetkezett változásokat. *Ezt úgy tudják megvalósítani, hogy a vírus szempontjából kritikus interrupt hívásokat a vírus önmagára irányítja, majd az adott interrupt meghívása esetén a vírus, számára megfelelő értékeket ad vissza.*

Az első lopakodó vírus a Brain boot-vírus volt, ha aktivizálni tudta magát a memóriában, akkor mindig az eredeti állapotában mutatta a bootszektort és ez a tulajdonság minden más lopakodó boot-vírusra is igaz. Egy lopakodó file-vírust nehezebb megírni, mégis jóval több lopakodó file-vírus létezik, mint lopakodó boot-vírus! Érdekes módon a lopakodási technikát egyes makróvírusok is felhasználják: eltűntetik az adott alkalmazás azon menüpontjait, amelyen keresztül meg lehetne nézni az aktív makrókat... Természetesen erről is szó lesz a következő fejezetben.

A lopakodó file-vírusok egyszerűbb típusai általában csak a fertőzés előtti eredeti file-méretet szokták mutatni, ha fizikailag a file-hoz nyúlunk (pl. file megnyitás), már észrevehető a változás. A teljesen lopakodó típusú file-vírus mindent az eredeti, fertőzés előtti formában mutat: ha fertőzött programot indítunk el és a vírus még nem rezidens, akkor a vírus aktívvá válik a memóriában és leszedi magát a fertőzött programról. Ha már rezi-

dens, akkor nem történik semmi különös, hiszen a memóriában ülő példány automatikusan leszedi a másik példányt a fertőzött programról. Ezután vezérlést kap az elindított program, ha az esetleg ellenőrzi saját magát, akkor semmi változást nem észlel, a program lefutása után a vírus újra visszatelepszik a célpontra.

Ha egy lopakodó vírussal fertőzött programot nem elindítani szeretnénk, hanem egyéb fileműveletet szeretnénk rajta elvégezni, akkor annyiban bonyolódik a helyzet, hogy csak akkor látjuk fertőzöttnek a programunkat, ha a vírus nem aktív, tehát nincs ami eltávolítsa a fertőzést. Ha a vírus már aktív, akkor bármilyen fileművelet esetén eltávolítja magát a programról, majd a műveletek befejezése után (file lezárás) újra visszatelepszik a programra.

Vagyis egy lopakodó vírus csak akkor tud lopakodva fertőzni, ha aktív a memóriában. Ellenkező esetben – pl. ha tiszta rendszerlemezről indítottuk a gépet – már észrevehető lesz a rendszerben történt változás, pl. a file-ok méretének növekedése.

Sajnos a legtöbb esetben a vírusírók a lopakodási technikát a gyors fertőző technikával ötvözik, ilyenkor nagyon gyorsan és széleskörben képes terjedni egy vírus.

Mint fentebb említettem, minden lopakodó vírus a saját szempontjából kritikus interruptokat irányítja magára, de az alábbi 6 interrupt átirányítás szinte minden ilyen vírusra igaz: 21h = Dos funkciók, 24h = kritikus hibák, 25h = abszolút lemezolvasás, 26h = abszolút lemezírás, 1Ch = idő, 28h = Dos Ok, 13h = disk i/o műveletek.

Lopakodó boot/partíciós vírusok esetén a disk i/o műveletek, abszolút lemezolvasás és lemezírás átirányítása teljesen egyértelmű: lemezolvasás esetén az eredeti boot szektort vagy partíciós táblát mutatja meg, ha arra a területre történik az olvasás, lemezírás átirányításával egyrészt a további fertőzéseket intézi, másrészt önmagát védi felülírás ellen.

Lopakodó file-vírusoknál a Dos funkciók átirányítása is egyértelmű: ezen keresztül lehet lekérdezni egy file méretét – amire a vírus nem valós értéket fog visszaadni, ezen keresztül lehet minden fileműveletet elvégezni szabványosan – amire a vírus eltávolítja magát, ha a fertőzött programmal történne az adott művelet, de itt lehet lekérdezni azt is, hogy a felhasználó milyen programot indított el – amit rögtön a futás után meg- vagy újra lehet fertőzni.

A lopakodó vírusok az átirányított interrupt rutin elejére beszúrnak egy 5 byte-os hosszúságú ugró (jump far) utasítást – mely magára a víruskódra mutat, vagy magának az interruptnak a kezdő címét írják át arra a címre, ahol az adott interruptot lekezelő vírus rutin található. Ha egy olyan interrupt hívás történne, mely a vírus által ellenőrizve van akkor:

- a vezérlés a hívott interruptra kerül,
- az ugróutasítás vagy az átírt interrupt cím miatt a vezérlés a vírusra kerül,
- ugróutasításos módszer esetén az eredeti interrupt kódot visszahelyezi az 5 byte-os jump helyére, vagyis felülírja azt,
- a vírus ellenőrzi, hogy olyan funkció lett-e meghívva, ami a vírus szempontjából „kényes". Ha igen, akkor megteszi a számára szükséges lépéseket (pl. megnyitás miatt el kell távolítania magát a programról),
- a vírus átadja az igazi interruptnak a vezérlést,
- az interrupthoz tartozó rutin végrehajtja a kívánt funkciót és a vezérlés visszakerül a vírushoz,
- ha szükséges, akkor az interrupt értékeit manipulálja a vírus (pl. a file-hosszt csökkenti),
- ugróutasításos módszer esetén az 5 byte-os jump-ot visszateszi a „helyére",
- a vírus visszaadja a vezérlést az interruptot hívó utasítás utáni utasításra.

Mint látható, a vírus szinte mindent befolyásol és ez szükséges is neki, máskülönben nem lenne képes a lopakodásra.

Néhány lopakodó file-vírust a 4. pontban lehet beugratni: pl. 4096 vírus azokról a végrehajtható állományokról, amiket átnevezünk nem végrehajthatóvá, eltávolítja magát ha aktív a memóriában. Más lopakodó vírusoknál jól használható módszer a vírusos file-ok betömörítése: a tömörítő a tiszta programot olvassa be, hiszen a vírus eltávolította magát róla vagy pedig nem is volt fertőzött, ugyanakkor a kiírt tömörített kódot már nem fertőzi meg, hacsak nem olyan vírussal van dolgunk, amely a tömörített állományokba képes belemászni. Persze *önkicsomagoló állományt nem szabad létrehozni, mert akkor ez további fertőzési gócpontot jelenthet.* Sajnos nem minden lopakodó vírust egyszerű ilyen kézi módszerekkel eltávolítani, mert a legtöbb már kivédi az ilyen trükköket. *Ezeket csak a megfelelő vírusirtó programmal lehet eltávolítani a rendszerből.*

3.3. COMPANION VÍRUSOK

Ha a felhasználó a DOS-os parancssorban egy végrehajtható állomány nevét gépeli be, akkor a parancsértelmező – általában a **Command.com** – az aktuális könyvtárban egy olyan .com állományt keres, amelynek a neve megegyezik a felhasználó által

begépelt névvel. Ha talált ilyet, akkor azt lefuttatja, ha nem, akkor egy ugyanolyan nevű .exe állományt keres. Ha .exe-ként találja meg, akkor lefuttatja, ha nem, akkor .bat kiterjesztésűként keresi a programot. Ha megtalálja, akkor lefuttatja, ha nem, akkor az elérési útban – **PATH** – keresi a programot ugyanilyen sorrendben (**.com, .exe majd .bat**). Ha itt sem találja, akkor hibaüzenetet ad vissza. Persze a felhasználó ebből a keresgélésből nem lát semmit, csak azt, hogy elkezd futni a begépelt nevű program vagy hibaüzenet ír ki a gép.

Lássunk konkrétan egy példát és az algoritmus lépéseit:

```
Pl. C:\>demo (+ enter)
```

Erre a parancsértelmező a következőképpen reagál:

```
ha a demo egy belső parancs, akkor végrehajtja
 ↳különben ha van demo.com az aktuális könyv-
  tárban akkor indítja
 ↳különben ha van demo.exe az aktuális könyv-
  tárban akkor indítja
 ↳különben ha van demo.bat az aktuális könyv-
  tárban akkor indítja
 ↳különben ha van demo.com a path 1. könyvtár-
  ban akkor indítja
 ↳különben ha van demo.exe a path 1. könyvtár-
  ban akkor indítja
 ↳különben ha van demo.bat a path 1. könyvtár-
  ban akkor indítja
 ↳különben ha van demo.com a path 2. könyvtár-
  ban akkor indítja
 ↳.....
 ↳.....
 .
 .
 .
   ↳ különben hibaüzenet: „Bad command or file
  name"
```

Ez a programfuttatási sorrend egy új ötletet adott a vírusíróknak: ha egy vírus .exe vagy .bat állományt akar megfertőzni, csak létre kell hoznia egy ugyanolyan nevű .com állományt és ha a felhasználó csak a nevet gépeli be kiterjesztés nélkül, akkor a vírust indította el! Ehhez a fertőzéshez a megtámadott objektumokat sem kell megváltoztatni, ezzel az ellenőrző összeges vírusvédelmet sikerült átverni. Természetesen a vírusnak gondoskodnia kell

az eredetileg futtatni kívánt program indításáról is, de ez már nem nehéz feladat.

Ezeket a vírusokat **CEB** (**.com, .exe, .bat**) a külföldi szakirodalom szerint companion vírusoknak nevezik. Ez a vírustípus nagyon rendszerfüggő, főleg a DOS operációs rendszerre jellemző.

Ezek a vírusok azonban kevésbé terjedtek el, mert a legtöbb ember valamilyen file-manager programot használ – legyen szó DOS-ról vagy Windows-ról, ezek használatával a felhasználó gyakorlatilag a kiterjesztés „begépelésével" indítja a programokat, tehát kevésbé valószínű, hogy egy vírusos program induljon el.

A kevésbé elterjedtség másik okát talán éppen szaporodási elvükben kell keresni: ha egy „szabványos" file-vírussal fertőzött állományt átmásolunk egy másik gépre akár floppyn, akár hálózaton keresztül és ott a fertőzött program futtatva van, akkor az a másik gép megfertőződött. De ezeknél a vírusoknál hiába másolunk át egy programot, az nem fogja tartalmazni a víruskódot – hacsak nem másoltuk át magát a .com állományt tartalmazó vírust is. Pontosan ennek az átmásolásnak kicsi a valószínűsége. Persze véletlenül is átterjedhet a vírus más gépre: lemásolhat egy barátunk egy egész programcsomagot, amibe esetleg már belemászott a vírus és ezzel már lehetősége van a másik gépen a szaporodásra.

A companion vírusoknak több alfaja is létezik, melyeket a fertőzési mód szerint lehet egymástól elkülöníteni.

3.3.1. Általános companion vírusok

Ez a companion vírusfajta abba a könyvtárba teszi bele a vírust tartalmazó .com állományt, ahol megfertőzni kívánt .exe vagy .bat található: pl. ha a speed.exe-ét megfertőzi a C:\UTILS> alkönyvtárban, akkor a C:\UTILS> alkönyvtárban egy speed.com állomány fogja tartalmazni a vírust. A ma létező companion vírusok többsége ilyen típusú.

Ez a típus a fentebb említett okokon kívül talán azért is terjedt el kevésbé, mert egy átlag felhasználó részére is különösen feltűnő lehet, ha egy könyvtáron belül egy vagy több programnak is létezik .exe és .com verziója, ráadásul az összes ilyen program .com kiterjesztésű változata ugyanakkora (ezek tartalmazzák a víruskódot). Ezeknek a vírusoknak az irtása különleges szakértelmet sem igényel, hiszen egyszerűen csak a vírust tartalmazó .com állományokat kell törölni.

Az általános companion vírussal Novell hálózati operációs rendszer alatti, csak végrehajtható (**ExecuteOnly**) programokat is meg lehet fertőzni, ha arra a könyvtárra van írási jogunk, amiben a vírus által megfertőzni kívánt programok találhatóak.

3.3.2. Path companion vírusok

Ezek a vírusok a víruskódot tartalmazó .com állományt abban az elérési útban (**Path**) szereplő alkönyvtárak valamelyikébe teszi, amelyek előrébb vannak a Path-ban, mint az indított program könyvtára: pl. ha elérési útnak „**C:\Utils;C:\Norton;C:\System**" van beállítva és a megtámadott program a **C:\System** alkönyvtárban található, akkor a vírus vagy a **C:\Utils** vagy a **C:\Norton** alkönyvtárban hoz létre egy ugyanolyan nevű, fertőzött .com állományt.

Ha a megtámadott program könyvtára nem szerepel az elérési útban, akkor nem fertőzi meg a vírus, hiszen indításkor a rendszer mindig először azt vizsgálja meg, hogy az aktuális alkönyvtárban szerepel-e az elindított program. Ha igen, akkor onnan indítja, a Path-t egyáltalán nem vizsgálja. Vagyis ez a vírustípus csak olyan programokat képes megtámadni, amelyeknek az alkönyvtára benne van az elérési útban. Talán most már érthető, hogy ezek a vírusok miért terjedtek el még kevésbé, mint az általános companion típusok.

Bár nagyon furcsa szaporodási formáról van szó, ez a forma kevésbé rendszerfüggő, hiszen más operációs rendszerekben is van elérési út környezeti változó (még ha nem is így hívják). Vagyis más operációs rendszerekre kifejlesztett vírusok alkalmazhatják ez a szaporodási formát.

Egy path companion vírus képes „megfertőzni" minden file-t a Novellen, függetlenül azok védelmétől: mivel a felhasználónak a munkaállomáson biztos, hogy van írási joga, ezért a vírus ekkor képes lehet arra, hogy a Path környezeti változóban lévő könyvtárak elé beszúrjon a munkaállomáson egy írható alkönyvtárat, amibe a fertőzött .com állományokat rakja.

3.3.3. Alias companion vírusok

A **Doskey** program megengedi a felhasználónak, hogy parancssor makrókat definiáljon, amelyeket angolul alias-oknak nevezünk. Ha begépelünk valamit és az egy definiált makró, akkor a makróban definiált parancs(ok) fog(nak) végrehajtódni először, még mielőtt bármilyen végrehajtható állo-

mány elindulna. Lehetőség van arra is, hogy már meglévő parancsokat átdefiniáljunk: pl. a **Dir** parancs helyett egy **Cls** parancs is végrehajtódhat! Ez az a lehetőség, ami megengedi alias companion vírusok létezését, mivel nem egészen jó szándékú hozzárendelések is történhetnek....

Az ilyen típusú vírus első futásakor az **Autoexec.bat** file-t módosítja, amiben olyan makrókat definiál, hogy lehetőleg bármilyen belső parancs kiadásakor a víruskód fusson le. Ezáltal biztosította, hogy rendszerindítást követően nagy valószínűséggel a víruskód is lefusson. Aktivizálódás után pedig úgy fertőzi a programokat, hogy ugyanolyan néven makrókat hoz létre, amikben a víruskódja után van indítva az eredeti program és ezeket a makrókat egyúttal jegyzi is az **Autoexec.bat**-ba. Így bizonyos idő eltelte után az **Autoexec.bat** egészen tekintélyes méretűre fog növekedni, viszont ha ily módon fertőzött programot indítunk, akkor biztosan lefut a vírus is.

A makróból indított víruskód egy külön bináris állományban található, tehát egy víruskóddal megfertőzhető több program is. Más gépre úgy kerülhet át egy ilyen vírus, ha véletlenül belekeveredik egy programcsomagba. Igaz, ahhoz, hogy a másik gépen aktivizálódni tudjon, a felhasználónak is figyelmetlennek kell lennie, amikor elindítja (mivel azon a gépen még nincsenek makródefiniálások).

Az ilyen vírusok jelentik a legkevesebb veszélyt – igaz nem szabad alábecsülnünk a lehetőséget – hiszen eléggé rendszerfüggőek, ráadásul az egyre jellemzőbb Windows 95 alatt nincs is értelme az ilyen vírusoknak.

3.4. FAT VÍRUSOK

Igazság szerint eddig egyetlen **FAT** vírus a **Dir2-Fat** (vagy **Creaping Death** vagy **Cluster Buster** vagy ki hogyan nevezi) jelent meg, bár ennek rengeteg variánsa létezik. Lényegében egy nagyon intelligens vírusról van szó, amely ravasz módon képes szaporodni. Ha ezt a vírust követné még más **FAT** vírus is, akkor arra (és erre is) a következő tulajdonságok jellemzők:

- A **FAT** vírusok egy példányban teszik fel magukat a lemezre (floppy és winchester), annak tetszőleges clusterére (a **Dir2** az utolsóra, ha az foglalt, akkor az elé telepszik). Ha megtámadnak egy programot, akkor nem írnak bele és nem is változtatnak meg rajta semmit, csupán csak a program kezdő clusterszámát állítják át önmagukra, az eredeti program elveszett clusterekre „kerül". Ha egy fertőzött

- programot indítunk, akkor valójában legelőször a vírus indul el, a vírus lefutása után átadja a vezérlést arra az elveszett clusterre, ahol a valódi program kezdődik. Ez a fajta támadás szinte tökéletesnek mondható: a program megfertőződik, de nem változik, a víruskód végrehajtása után az eredeti program is lefut, tehát szinte semmit nem lehet észrevenni.
- A **FAT** vírusok a clusterszám-átírás miatt nem növelik meg a fertőzött file méretét, így az ellenőrző összeges filevédelmet könnyű kivédeniük.
- Rezidensek, ha aktívak a memóriában, akkor minden tökéletesnek látszik. Ha nem aktívak (pl. mert tiszta rendszerlemezről indítottuk a gépünket) és le akarunk másolni egy programot, akkor a directory bejegyzésnek megfelelően egy clusternyi szektort sikerül csak kimenteni, ráadásul az is csak a vírust tartalmazza. Ha ezt a file-t egy másik gépen elindítjuk, akkor azt a gépet „sikeresen" megfertőztük.
- Ha aktívak a memóriában, akkor a nem fertőzött lemezekre azonnal rámásznak a terjedési elvüknek megfelelően.
- Minden végrehajtható kiterjesztésű programra képesek rámászni.

Mint látható, a terjedési és szaporodási elv egyszerű és „nagyszerű". Sajnos egyesek nem vették komolyan a Dir2 vírust, mert úgy gondolták, hogy a vírus – mivel másolhatatlanná teszi a gépben lévő programokat – kitűnően véd programlopás ellen. Mégsem ajánlom alkalmazását, mert nemcsak szaporodik, hanem rombol is!

3.5. BATCH VÍRUSOK

A batch egy viszonylag korlátozott „**programozási nyelv**", melynek segítségével lehetőség adódik a rendszeres, napi, heti feladatok kötegelt végrehajtására, **DOS**-ban. Ezeket a kötegelt feladatokat egy szöveges batch állományban batch utasítások, Dos belső parancsok és külső programok végrehajtásával végezhetjük el. A .bat kiterjesztésű batch állományok közvetlenül futtathatóak – ha a nevüket a prompt-nál beírjuk, ekkor a Dos az állományban található parancsokat, utasításokat soronként végrehajtja, mint egy interpreter. A batch programok lehető legjobb példája az Autoexec.bat, amelyben a felhasználó igénye szerinti rendszer-beállítási utasítások minden egyes rendszer indításkor automatikusan végrehajtódnak (még Windows 95 alatt is).

Ez a batch programozási lehetőség annyira fejlett (tartalmaz feltételes elágazásokat és ciklusokat is), hogy sajnos vírusok készítésére is felhasználható, ráadásul éppen a Dos sajátosságai miatt több fajta szaporodási technikával fertőző batch vírust fejlesztettek ki.

3.5.1. Általános batch vírusok

Ezek a batch vírusok kizárólag batch parancsok és Dos parancsok felhasználásával szaporodnak. Mindenekelőtt lapozzunk vissza egy kicsit az előző fejezethez, ahol a vírusok reprodukciós rutinját jellemeztem: első lépésként a vírus keres valamilyen célpontot, amit megfertőzhet. Most erre sokan azt mondják, hogy már ez a lépés sem megvalósítható, válaszul egy létező batch vírus célpont kereső sorát hadd szúrjam ide:

```
for %%a in (*.bat ..\*.bat) do set _!=%%a
```

A célpont keresés után a Find parancs segítségével a vírus ellenőrizheti, hogy a megtalált célpont fertőzött-e. Ha nem fertőzött, akkor egy kiírás-átirányítással máris a célponthoz „ragaszthatja" magát.... Természetesen a célpont fertőzöttségének ellenőrzése itt sem garantált, tehát egy batch vírus akár felül is írhatja a megtámadott batch állományt!

Fertőzöttség-ellenőrzésen kívül pusztán batch parancsokkal megoldható, hogy a vírus egy futás alkalmával csak egy illetve tetszőleges számú célpontot fertőzzön meg, nem is beszélve a **PATH** felhasználásáról, mint lehetséges célpontok helyeiről.

Bár ezeknek a vírusoknak eddig csak hozzáfűző (appendelő) és felülíró típusaik jelentek meg, elméletileg megvalósíthatóak a batch program elejére beszúró és a batch programon belül véletlenszerű helyre beszúró illetve felülíró típusaik is.

Van azonban további két érdekes fajtájuk is ezeknek a vírusoknak:

- **„Call insert" batch vírusok**

Ezek a típusok a megtámadott batch állományba csak egy „call" utasítást szúrnak be – akár tetszőleges helyre, ami egy másik fertőzött batch állományra mutat. Ezzel azt érik el, hogy ha a megtámadott batch állományban erre a sorra kerül a végrehajtás, akkor az meghívja a vírust tartalmazó batch programot. A vírusok szempontjából ennek a fertőzési módszernek az egyedüli hátránya az, hogy ha egyszer egy másik gépen futtatunk egy „call"

utasítással „**fertőzött**" batch programot, akkor az hibaüzenetet fog visszaadni, ami feltűnő lehet.

A fertőzött batch program meghívása azonban másképp is lehetséges: ha a valóban fertőzött batch program neve – esetleg elérési úttal együtt – a megtámadott batch programban szerepel, akkor is le fog futni a vírus. Ennek azonban az a hátránya, hogy a vezérlés ezután nem fog visszakerülni a hívó batch állományra: ha a megtámadott batch állomány végén történik a fertőzött batch program indítása, akkor semmi nem fog feltűnni a felhasználónak, ellenkező esetben a megtámadott batch programban a fertőzött batch állományra való hivatkozás utáni utasításokból egyik sem fog végrehajtódni, ami növelheti a vírus lebukásának esélyeit.

Ezt a fajta fertőzést könnyen hasonlíthatjuk a **FAT** vírusokhoz, hiszen itt is egyetlen példányban van a gépen az egész rendszert megfertőző batch vírus.

- **Companion batch vírusok**

A batch vírusok ezen típusa a **DOS CEB** tulajdonságát használják ki: programokat átneveznek tetszőleges névre, majd a programnevén létrehoznak egy batch állományt, ami magát a vírust tartalmazza. Ezzel a vírus azt érte el, hogy ha a felhasználó csak programnév begépelésével indít programokat, akkor az átnevezett programoknál valójában a batch vírust futtatta le, a turpisságból természetesen semmit sem lehet észrevenni, hiszen a vírus a más névre átnevezett programot is elindítja.

A vírus szaporodási nehézségei csak abból adódhatnak, ha a felhasználó file-manager segítségével indít programokat, ilyenkor feltűnő lehet programok eltűnése (mivel át lettek nevezve), ha viszont azokat a programokat veszi célba, amelyeket a felhasználó azért rakott az elérési útban szereplő valamely könyvtárba, hogy könnyebb legyen az indításuk, akkor azok között jobban elterjedhet.

3.5.2. Hibrid batch vírusok

A hibrid típusok kihasználják azt a lehetőséget, hogy a batch programon belül bináris utasításokat, parancsokat lehet elhelyezni és végrehajtatni a géppel, az alábbi módszerekkel:

- Batch programból tetszőleges szöveges állományt lehet létrehozni, ha a képernyőre való kiíratás át van irányítva egy file-ba, amit fel

lehet használni a **DOS** alatt is megtalálható Debug program bemeneteként – hasonlóan az **ASCII** vírusoknál említettekhez.

- A kiíratás átirányításának eredménye nemcsak szöveges, hanem futtatható bináris állomány is lehet, a kiíratás után közvetlenül tudja futtatni bináris részét a batch vírus.
- A 3. bináris kód futtatási megoldás egy kicsit trükkösebb: a batch vírus átmásolja önmagát egy .com kiterjesztésű file-ba, majd ezt futtatja. Igen ám, de egy .com program futtatása egészen másképp történik, mint egy .bat programé, hiszen a .com programot betölti egy szegmensre a gép, majd a bináris kódot (mint utasításokat) végrehajtja, míg a .bat programoknál soronkénti értelmezés és feldolgozás történik. Válaszul szintén egy batch vírus részletet szeretnék ide rakni:

```
@ECHO OFF
REM „bináris kód 1."
copy %0 virus.com>nul
......
REM „bináris kód 2."
```

Nézzük meg mi történik, ha ezt .com programként betöltenénk (első oszlopba a batch állomány egyes karaktereit írtam, a második oszlop az egyes karakterek bináris kódjának megfelelő utasításokat tartalmazzák címmel együtt):

```
SzövegCím + Utasítás

@CS:100INC AX
ECS:101INC BP
CCS:102INC BX
HCS:103DEC AX
OCS:104DEC DI
„space" OFCS:105AND [BX+46],CL
FCS:108INC SI
„enter+köv. sor" RCS:109OR AX,520A
ECS:10CINC BP
MCS:10DDEC BP
itt jönne az 1. valódi utasítás
```

Vagyis a batch állományban található első 2 utasítás (**@ECHO OFF, REM**) különösebben nem befolyásolja a .bat-ból átvedlett .com prog-

ram futását. Az első bináris kód csak egy ugró utasítást tartalmaz a második bináris kódra, a két bináris kód közötti batch utasítások első futtatáskor másolják át a batch programot .com kiterjesztésű állományba, illetve tűntetik el a felhasználó elől az átmásolás nyomait.

A bináris kódot is alkalmazó batch vírusok ezáltal átlépték a batch programozás korlátait: lehetnek rezidensek és gyors fertőzőek, az összes lehetséges pusztításra képesek, mint file-vírus társaik, a bináris rész tartalmazhat olyan víruskódot, ami bináris programok fertőzésére is alkalmas...

A bináris kódot is alkalmazó batch vírusok szinte bármekkora pusztítást véghez tudnak vinni: itt a vírusok objektív rutinja szerinti osztályozásánál említett összes lehetőség adott, pl. hardverpusztítás is! *Csak batch utasításokból felépülő vírusok pusztítása szoftver szintre korlátozódik: lemez formattálás, nem szabványos partíciós tábla program (pl.* ***Linux*** *féle* ***LILO****) tönkretétel, file-ok törlése, könyvtárak törlése, Ansi bomba aktivizálása.*

A batch vírusok kevésbé elterjedtségének egyik oka az, hogy manapság egyre inkább jellemző a grafikus rendszer használata – ahol nem szoktak batch programokat használni, másrészt jóval kevesebb ilyen vírust fejlesztettek, harmadrészt a batch vírusokban használt utasítások nagy része régebbi **DOS** verziókban még nem volt ismert, tehát jóval kevesebb potenciális fertőzési célpont létezett.

Ami miatt viszont érdekes foglalkozni velük, az az, hogy manapság rengeteg olyan program létezik, ami batch állományokból .exe vagy .com programot generál....

Bár az itt felsorolt vírusok a számítógépes alvilág régebbi szörnyszülöttei, sajnos az újabb típusok az ezekben a vírusokban található tippeket, trükköket is felhasználják, elegendő a mutáló és lopakodó makróvírusokra gondolni. Valljuk meg őszintén: *ezek a vírustechnikai ötletek egyszerűek és egyben óriásiak is, a probléma csak az, hogy az ilyen ötletek ellen nehéz védekezni.*

4. MAKRÓVÍRUSOK

A makrók olyan utasítások és műveletek sorozata, amit egy csoportba összegyűjtve úgy használhatunk, mintha egyetlen parancs volna, vagyis könnyebbé és egyszerűbbé teszi a rendszeres napi, heti, havi feladatok végrehajtását, akár automatizálva is azokat, megspórolva ezzel rengeteg időt és pénzt.

A makrókra az egyik lehető legjobb példa az **DOS batch** „programozási" nyelve, igaz más operációs rendszerek (pl. **OS/2**) is rendelkeznek batch programozási lehetőséggel. Ezek alapján bárki joggal állíthatná, hogy makróvírusok már a 90-es évek előtt léteztek, hiszen a batch vírusok már ekkor megjelentek, de:

- minden egyes operációs rendszernek más-más a batch programozási nyelve, így nem írható olyan batch vírus, amely az egyes rendszerek között szabadon tud szaporodni,
- a makróvírusok felhasználói alkalmazások, programok makrónyelvi lehetőségeit használják ki, nem pedig operációs rendszerekét.

Makrók felhasználhatóságának másik legjobb példájaként bármilyen más irodai alkalmazást említhetnék, mint pl. **Word, Excel, Access, Ami Pro, WordPerfect, Lotus** stb: egy dokumentum kitöltése olyan adatokkal, amelyek rendszeresen ismétlődnek, vagy megadott értékek alapján különböző adatok vagy táblázatok kiszámolása, grafikonok megrajzolása mind-mind makrókkal megoldható. A grafikus alkalmazásoknál használható „varázslók" valójában szintén makrók!

Ezek az alkalmazások olyan fejlett makró programozási nyelvvel rendelkeznek, hogy azokban vírus készíthető, mely az adott alkalmazás által készített adatállományokban – ez lehet dokumentum, sablon, táblázat is – képesek szaporodni, természetesen alkalmazásonként más-más módszerrel, de erről lesz szó a későbbiek folyamán.

Sajnos makróvírusok megvalósíthatóságára különböző vírusíró csapatok is rájöttek és próbaképpen kifejlesztették első makróvírusaikat. A **Telix** kommunikációs program volt az első olyan program, melyre egy holland vírusíró makróvírust fejlesztett 1993-ban. A vírus a Telix script-jeit volt képes megfertőzni (egy Telix-es script olyan utasításokból áll, amelyeket a Telix hajtott végre, pl. ahhoz., hogy az adott Internet szolgáltatót a felhasználó elérje), s bár ez a vírus nem terjedt el széles körben, kifejlesztői számára igazolta kísérletük sikerét, ami egyúttal a makróvírusok jövendőbeli megjelenését is jelentette.

Nemcsak vírusírók készítettek azonban makróvírusokat, hanem víruskutatással foglalkozó szakemberek is: **az első Word makróvírust Joel McNamara készítette el 1994-ben,** kutatási céllal. Kutatási eredményeit a kísérletek elvégzése után, a vírus „forráskódját" az első nem teszt jellegű **Word makróvírus** (**Concept**) megjelenése után publikálta. A McNamara által készített **DMV** (**D**ocument **M**acro **V**irus) és a **Concept** vírus romboló rutinokat nem tartalmazott, az ezután megjelent Word makróvírus, a Nuclear viszont már nemcsak rombolt, hanem bizonyos feltételek teljesülése esetén már egy file vírust is szabadon engedett magából.

A **Concept** megjelenésekor széles körben elterjedt, s a nevetséges az, hogy ezt részben a Microsoft-nak is köszönheti: a **Microsoft kiadott két CD-t,** amelyen ezzel a vírussal fertőzött dokumentok (pontosabban sablonok, de erről majd később) voltak. Az egyik CD-t „**Microsoft Windows '95 Software Compatibility Test**" címmel több mint ezer OEM partner cégnek postáztak még 1995 közepén, a másik CD-t 1995 augusztus-szeptember környékén adták ki „**The Microsoft Office 95 and Windows 95 Business Guide**" címmel. A Microsoft közel két hónap (!) hallgatás után ismerte el hibáját, kifejlesztettek egy Concept vírusirtó makrót – amit bárki ingyenesen letölthetett a Microsoft honlapjáról – azonban ennek a vírusvédelmi makrónak az első verziója hibás volt, amit később korrigáltak.

Sajnos nem ez volt az utolsó ilyen eset: nem olyan rég a Microsoft egyik európai honlapjára egy Wazzu Word makróvírussal fertőzött dokumentum került fel és csak néhány hét (!) után cserélték le vírusmentesre.

Ehhez hasonló bakikat azonban nemcsak a Microsoft követett el, hanem más cégek, többek között az egyik hazai szaklap is, amely CD-jére szintén egy Word makróvírusos dokumentumot tett fel, azonban észrevéve a hibát, a következő havi újságban részletesen beszámoltak az előző havi makróvírus fertőzöttségről, a CD-n pedig mellékeltek ingyenesen terjeszthető makróvírus-irtó programot.

Beindult a „hadigépezet": antivírus cégek sokasága próbálkozott különböző makróvírus-kereső és -irtó programok kifejlesztésével, de a lehető legnagyobb problémát az jelentette, hogy a Word sablonfile felépítése nem volt publikus, így minden cég más-más úton próbálta megközelíteni a makróvírusok irtását egészen addig, amíg Concept megjelenését követő 5. hónap után (!) a Microsoft mégiscsak adott némi használható információt a felépítéssel kapcsolatban.

Nemcsak az antivírus programfejlesztők hadigépe indult be: egymás után jelentek meg a Word makróvírusok, melyek egyre fejlettebb technikával rendelkeztek, többek között lopakodó és mutációs lehetőségekkel. **A későbbiek folyamán a vírusfejlesztők figyelme más alkalmazások felé te-**

relődött, így fejlesztettek makróvírusokat Excel, Access, Ami Pro, Lotus, Quattro, WordPerfect, Autocad, CorelDraw és PageMaker alá is, bár mind a mai napig a legkedveltebb makróvírus fejlesztési környezet a Word.

Egy 1997-es amerikai felmérés szerint azoknál a cégeknél, amelyek átestek számítógépes vírusfertőzésen, az esetek 49%-ában makróvírusról volt szó!! A felmérés folyamán azt is vizsgálták, hogy az egyes makróvírusok hogyan jutottak a számítógépes rendszerekbe: legtöbb esetben a felhasználók a fertőzött dokumentumokat, táblázatokat belső hálózaton, vagy Interneten keresztül e-mail-hez hozzá kapcsolva kapták, további esetekben a gépek floppy-n való dokumentációcsere révén, illetve ismeretlen honlapról való dokument letöltéskor fertőződtek meg – rendszerint Word – makróvírussal.

A makróvírusok jövőjét talán az is jellemezheti, hogy míg a 80-as évek elején több mint 5 évnek kellett ahhoz eltelnie, hogy az összes, bármilyen típusú vírus száma 1000 fölé emelkedjen, addig ezt a számot a makróvírusok röpke 3 év alatt érték el. Igaz, ez az 1000 db makróvírus mindössze 300 makróvírus-család valamelyikébe sorolható be, de a számok mindenesetre elég riasztónak tűnnek.

Vagyis a súlypont áthelyeződött a régebbi file-, boot-vírusok írásáról a makróvírusok fejlesztésének irányába, mert

- az antivírus kutatók közül nagyon kevesen számoltak a makróvírusok megjelenésével, terjedt az a tévhit, hogy adatállományokban, szövegfile-okban nem lehetnek vírusok (ez ma már csak azokra az állományokra igaz, melyek felépítésükből adódóan nem tartalmazhatnak makrókat),
- ebből kifolyólag az átlag felhasználók mindenféle ellenőrzés nélkül cseréltek és cserélnek dokumentumokat, táblázatokat, adatbázisokat,
- a különböző állományok cseréjét segíti és felgyorsítja a belső vállalati hálózatok és az Internet egyre nagyobb elterjedtsége,
- az adott alkalmazások makró programozási lehetőségei nagyon jól publikáltak (ez különösen a Word-re és az Excel-re igaz),
- makró nyelven való programozáshoz nem kötelező ismerni a gép „lelki" világát – mint boot- és/vagy file-vírus írásához, vagyis jóval könnyebb ebben fejleszteni, mint assembly-ben (pl. 1 napos Word makró help tanulmányozás után könnyen megírható egy alap Word makróvírus!!!!),
- egyes irodai alkalmazások – Word, Excel – nagyon elterjedtek, használatuk szinte teljesen szabványosnak tekinthető, éppen ezért nagyon sok a potenciális áldozat,

- végül, de nem utolsó sorban a legtöbb irodai alkalmazásnak más operációs rendszerekre és platformokra írott változata is létezik, a változatok között szinte semmi különbség nincs, a legtöbb esetben a makrónyelv teljesen egységes: pl. a Word 6.0-nak létezik Windows 3.1, Macintosh és NT változata és azt is figyelembe vehetjük, hogy a Word 6.0 Windows 95 alatt is tökéletesen fut. Éppen ezért azokról a makróvírusokról, amelyek nem tartalmaznak operációs rendszerfüggő utasításokat és az adott alkalmazás (majdnem) minden verziójában és változatában futnak, elmondhatjuk, hogy multi-platform és/vagy operációs rendszerfüggetlen vírusok. A vírusíróknak ez régóta dédelgetett álmuk volt...

További problémát jelent az, hogy a legtöbb irodai programban rendelkezésünkre álló **OLE** technika segítségével más alkalmazásokban létrehozott információkat, objektumokat, file-okat vagy kijelölt blokkokat tudunk beilleszteni, beágyazni. A beillesztéstől kezdve, ha kétszer kattintunk a beágyazott objektumra, megnyílik az az alkalmazás, amelyben az objektumot létrehoztuk és benne megjelenik az objektum, ezáltal az azonnal szerkeszthető. Amikor visszatérünk az eredeti programba, akkor megjelenik az objektumon végrehajtott összes módosítás. Vagyis egy makróvírus nemcsak a saját alkalmazása makrókat is tartalmazható állományában tud terjedni, hanem az OLE segítségével más állományokban is (feltéve, ha beágyazásra kerül a fertőzött dokumentum).

De nemcsak előnyei, hanem hátrányai is lehetnek – a vírusírók szempontjából – egy makróvírus megírásának:

- az adott alkalmazás makró nyelve magas szintű programozási nyelvekhez hasonlítható, így a rendszer közvetlen irányítását a vírus nem veheti át, csak olyan rendszerközeli dolgokat tud megcsinálni a vírus, ami az alkalmazás
  makró nyelvében engedélyezett (sajnos ez a hátrány profi vírusírók részéről kiküszöbölhető: ha a vírus dropperként hordoz egy bináris kódot...),
- a **makróvírusok futásához viszonylag sok memória kell**, bár a manapság jellemző gépek már elegendő memóriával rendelkeznek ehhez,
- egy makróvírus a gépet annál jobban lelassítja, minél nagyobb, viszont a sok funkciót (pl. lopakodó, polimorf funkciók stb.) tartalmazó makróvírusok eleve nagy vírusok lesznek, a gép „lelassulása" növeli a vírus lebukási esélyeit, hiszen felkeltheti a felhasználó figyelmét.

Mindezeket összefoglalva, számítógépvírus tipológiailag a **makróvírusok olyan 5. generációs, közvetlenül nem végrehajtható,** de makrókat tartalmazható állományokba épülő vírusok, melyek az adott alkalmazás makrónyelvi lehetőségeitől függően tudnak rombolni, jópofáskodni, „rezidenssé" vagy „nem rezidenssé" válni, képesek lassú vagy gyors fertőzésre illetve könnyen visszafejthetőek vagy pedig titkosítják makrókódjukat. A makróvírusok által megfertőzhető állományok: **.doc, .dot, .xl*, .smm, .mdb, .wp*, .wk*, .wiz, .wzs, .cdr.**

§

Ezek után nézzük meg, hogy a makróvírusok az egyes alkalmazásokban hogyan tudnak terjedni, aktivizálódni, pusztítani vagy „csak" ijesztgetni a felhasználókat.

4.1. WORD MAKRÓVÍRUSOK

4.1.1. A Word makrónyelvi jellemzői

A Word makrók műveletek egymásutánjai, melyeket rögzítés vagy szerkesztés útján **Word 97 előtti verzióiban WordBasic, Word 97-ben Visual Basic for Application (VBA) nyelven tudunk létrehozni**. Makrórögzítés esetén a Word az adott verzió makrónyelvén írt utasítások sorozataként tárolja el a makrót, ezt a rögzített makrót a későbbiek folyamán ugyanúgy lehet szerkeszteni, módosítani, mint a felhasználó által létrehozott, szerkesztett makrókat.

A **makrókat eszköztárhoz, menühöz vagy gyorsbillentyűkhöz rendelhetjük,** így használatuk éppoly kényelmessé tehető, mintha egy standard Word-parancs lenne. Ilyenkor elegendő, ha rákattintunk az eszköztár megfelelő gombjára, kiválasztjuk a makróhoz rendelt menüparancsot, vagy megnyomjuk a billentyűzet-kombinációt. Természetesen ez a hozzárendelés nem kötelező, ekkor egy külön menüpontból – Word 6.0-ban Eszközök|Makró – tudjuk futtatni a számunkra fontos makrót.

Új dokumentumainkat gyakran használt, dokumentumtípusokhoz tervezett sablonok felhasználásával is elkészíthetjük. A sablon egy dokumentumterv, amely az adott típusú dokumentumok mindegyikében szereplő szövegeket, ábrákat és formátumokat tartalmazza. Ezenkívül stílusokat, makrókat, gyorsszöveg bejegyzéseket, eszköztár gombokat, valamint egyedi menü- és gyorsbillentyű-beállításokat tartalmazhatnak, melyek egyszerűsítik a munkát. A Word megfelelő könyvtáraiban különböző sablonok találhatók, pl. üzleti levelekhez, számlakészítéshez, melyeket használhatunk változtatás nélkül, de módosíthatjuk is őket, sőt akár saját sablonokat is készíthetünk.

!

A **Word** az új dokumentumokhoz automatikusan a normál **sablont kapcsolja**, hacsak nem választunk másik sablont. A normál sablon egy általános sablon, amely tetszőleges dokumentumhoz használható, ezt a **Word egy Normal.dot** állományban tárolja, annak tartalmát, így az esetleg benne lévő makrókat is mindenegyes indításakor automatikusan betölti (nem biztos, hogy végrehajtja, de erről majd egy picit később). A Word a **Normal.dot** állományon kívül szintén betölti automatikusan a felhasználó által is megadható „**Startup**" könyvtárban – ami alapértelmezésben a Word könyvtárában található \Startup alkönyvtár – lévő összes sablon állományt – a bennük lévő makrókkal együtt, így gyakorlatilag a felhasználó kedvenc Word beállításai tetszőleges sablon file-okban tárolhatóak.

A Word a sablonokat .dot kiterjesztésű állományokban tárolja, amitől persze el lehet térni, mert kiterjesztéstől függetlenül felismeri, hogy sablon file-lal vagy pedig egyéb állománnyal van dolga. A Word által létrehozott dokumentumok makrókat nem tartalmazhatnak!

A felhasználó által rögzített vagy létrehozott makrókat alapértelmezés szerint a normál sablon globális makró területére rögzíti a Word, hogy a makrók használatára valamennyi dokumentumban lehetőség legyen. Ha a makrók rögzítésekor illetve létrehozásakor más, nem a „**Startup**" könyvtárban is megtalálható sablont választunk, akkor csak ahhoz a dokumentumhoz használhatóak a makrók, amelyek ehhez a sablonhoz kapcsolódnak, de lehetőség van az egyes sablonok közötti makrók átmásolására is.

Word makróvírus-vizsgálatának szempontjából nem elhanyagolható, hogy a Word-nek 4 verziója is használatos manapság: a **Word 2.0, Word 6.0, Word 95 és a Word 97.** Az egyes verziók közötti makrónyelvi eltérések az alábbiak:

- Word 2.0, 6.0 és a Word 95 makrónyelve a **WordBasic**, de az egyes verziók makrónyelve tovább lett fejlesztve úgy, hogy azok egymással csak felülről kompatibilisek. Ez azt jelenti, hogy pl. egy Word 2.0-ban írt makró biztosan fog futni Word 6.0 és 95 alatt, de ez fordítva már nem biztos, hogy igaz, hiszen egy Word 6.0 WordBasic-ben írt makró olyan utasításokat is tartalmazhat, melyek csak a 6.0 verziótól kezdődően találhatóak a WordBasic-ben.
- A **Word 6.0 és a Word 95 sablon file felépítése teljesen eltér a Word 2.0-tól,** ha ezeknél a verzióknál egy makrókat is tartalmazó sablont file-t lementünk 2.0 formátumban, akkor a lementett állomány nem fog tartalmazni makrókat, mert ekkor a Word 6.0 és a Word 95 csonkolja a sablon file-t. Ez azért fontos, mert egy fertő-

zött Word 2.0 állomány megnyitásával, Word 6.0-ás lementésével, majd újra Word 2.0 formátumú lementésével a vírus többé nem tud szaporodni Word 2.0 alatt (egy-egy víruspéldányról van szó!).

- A **Word 6.0 és Word 95 makrónyelve között nagyon kevés utasításbeli eltérés létezik,** viszont a Word 6.0 alatt írt makrók csak 16 bites API-t tudnak hívni. Ebből kifolyólag, ha a makróban található 16 bites API hívás, akkor az nem fog működni Word 95 alatt, még akkor is hibát eredményez, ha Windows 95 alá feltelepített Word 6.0-ban szeretnénk futtatni! A Word 6.0 és a Word 95 sablon file formátuma megegyezik, ebből már nem adódnak problémák az előző ponthoz hasonlóan.
- A Microsoft egységesítette irodai programjainak makrónyelvét az **Office 97**-ben, éppen ezért a **Word 97 makrónyelve a Visual Basic for Application (VBA)** lett. A felülről való kompatibilitás érdekében Word 97-ben a régebbi verziókkal készített makrókat a Word 97 automatikusan átkonvertálja VBA-ra és az esetek többségében ezek a makrók ugyanúgy használhatóak maradnak. Ám lehetséges, hogy lesznek olyan részei a makróknak, amelyek átkonvertálás után nem futnak: ez a probléma jellemzően akkor áll fenn, ha a makró speciális rendszerhívásokat is végez. A makrók visszafelé konverziója (VBA-ról WordBasic-re) már nem működik, így ha makrókat tartalmazó sablon file-t szeretnénk lementeni Word 6.0/Word 95 formátumban, akkor a Word 97 figyelmeztet, miszerint a más formátum való lementés a VBA makrók elvesztését jelenti. Ebből adódóan azok a makróvírusok, melyek VBA-ban lettek kifejlesztve, soha nem tudnak majd működni Word 6.0 vagy Word 95 alatt, valamint ugyanaz a probléma előfordulhat, mint a Word 2.0 – Word 6.0 között (eredetileg WordBasic-es makróvírussal fertőzött Word 6.0 dokumentum, amely Word 97-ben is le lett menve, Word 6.0 alatt vírusmentes lesz).

A Word-nek nemcsak az egyes verziói, hanem nyelvi változatai között is van különbség. Általában nyelvi verziónként különböznek a Word belső parancsai, eltérhetnek az egyes makrónyelvi parancsok és függvények (a fordítástól függően), illetve a normál sablon globális makró területére nyelvenként másképp kell hivatkozni: pl. amíg az angol verzióban a globális makróterületre „Global" néven, addig a magyarban „Globális" néven lehet hivatkozni.

4.1.2. A Word makróvírusok működése

Mint már fentebb említettem, makrókat kizárólag sablon file-ok tartalmazhatnak, dokumentumok nem, éppen ezért a Word makróvírusok is csak sablon file-okat képesek megfertőzni.

Elhelyezkedés, betöltődés

A makróvírusok az alábbi két módszerrel tudnak beletelepedni dokumentumokba, pontosabban sablonállományokba:

1. (Ál)dokumentumban tovább fertőző Word makróvírus

Ennél a szaporodási módnál a Word makróvírusokat terjesztő „dokumentok" valójában lementett sablonok, csak a szokásos .dot kiterjesztés helyett .doc kiterjesztést kapnak. Mivel a Word az állományok kiterjesztésétől függetlenül felismeri, hogy milyen file-lal van dolga, emiatt a hamis .doc megnyitásakor „rájön", hogy az egy sablon file, majd ennek megfelelően tölti be.

A PC-s felhasználó mindebből nem lát semmit, hiszen ő a file kiterjesztése miatt szentül meg van győződve arról, hogy egy dokumentummal van dolga, mert – a sablonfile adottságai miatt – az tartalmazza az általa legutoljára beleírt vagy megváltoztatott szöveget.

A **Macintosh Word**-öt használóknak éppen itt van egy hatalmas előnyük a PC-sekkel szemben: ha a dokumentumállomány valójában sablon, akkor egy sablon ikonnal látjuk, nem pedig a dokumentumokhoz rendelt ikonnal. Vagyis egy Macintosh felhasználó még jóval a dokumentum megnyitása előtt eldöntheti, hogy gyanús állományról (sablon file-ról) van szó vagy sem.

2. Dokumentumhoz csatolt sablonfile-ban továbbfertőző Word makróvírus

Ez egy kevésbé jellemző szaporodási mód, melynek lényege, hogy a megtámadott dokumentumba csak egy csatolási megjegyzés kerül, miszerint az adott dokumentumhoz nem a normál sablont, hanem egy másik, a vírusmakróit is tartalmazó sablont kell betöltenie a Word-nek. Eltérően az előző módszertől itt a dokumentum továbbra is dokumentum marad, és a vírus nem konvertálja át sablonná.

Ettől kezdve, ha a felhasználó egy ilyen dokumentumot nyit meg, akkor a Word automatikusan betölti a dokumentumhoz tartozó, fertőzött sablont. Ez a szaporodási módszer egy picit hasonlít a **DOS-os CEB** (companion) vírusokéra.

Ugyanakkor ezzel a módszerrel megoldható lenne az is, hogy hasonlóan a **DOS**-os FAT vírusokhoz a makróvírus csak egyetlen sablon file-t fertőzne meg és valamennyi megtámadott dokumentumban a csatolási megjegyzést erre az egyetlen fertőzött sablon állományra irányítaná. Ilyen „FAT-es" Word makróvírus eddig még nem jelent meg.

További variációs lehetőséget ad a vírusírónak az, hogy a csatolási megjegyzés nemcsak .dot, hanem más kiterjesztésű állományra is mutathat, hiszen a Word – hasonlóan az előzőekhez – az állományok kiterjesztésétől függetlenül felismeri, hogy milyen file-lal van dolga.

Ennek a módszernek az egyedüli nagy hátránya az, hogy a felhasználók rendszerint csatolások nélkül szokták dokumentumaikat másolgatni, így viszont a vírus nem tud egyik gépről a másikra átjutni. Ennél a módszernél a makróvírus egyedüli tovább szaporodási esélye az, ha a felhasználó Word-ből floppy-ra vagy hálózatra menti ki dokumentumait, ekkor a vírus tudja csatolni a fertőzött sablont.

Vezérlés, átadás

Egy fertőzött dokumentum betöltése után a makróvírusnak valamilyen úton át kell vennie a vezérlést az adott alkalmazás felett – a felhasználó tudta nélkül, hogy szaporodni tudjon. Ez az alábbi módon történhet meg:

1. Automakrók használatával

A WordBasic-ben (már a Word 2.0 verziójától kezdődően!) és VBA-ben is lehetőség van úgynevezett automakrók használatára. Ez azt jelenti, hogy ezek a makrók valamilyen tevékenység végrehajtásakor automatikusan lefutnak, ha a normál sablon, vagy az éppen betölteni kívánt sablon file tartalmazza ezen makrók bármelyikét. A Word-ben alábbi automakrók fognak lefutni:

- AutoExec: Word indításakor,
- AutoNew: új dokumentum létrehozásakor,
- AutoOpen: minden file (text állomány is) megnyitásakor,
- AutoClose: minden file bezárásakor,
- AutoExit: Word-ből való kilépéskor.

Mind a mai napig a Word makróvírusok többsége ezt a módszert alkalmazza: általában az **AutoOpen makró segítségével, már a fertőzött (ál)dokumentum megnyitásakor képesek aktivizálódni. AutoNew, AutoClose, AutoExit makrók használata kevésbé jellemző, az AutoExec makrót** csak a **Normal.dot-ból és a Word „Startup"** könyvtárában található sablon file-okból való automatikus aktivizálódásra használják a makróvírusok, ugyanis az AutoExec makrót hiába tartalmazza más egyéb sablon file, ha az nem töltődik be Word indításkor, akkor a makró a későbbi betöltés folyamán sem fog végrehajtódni.

Az automakrók használata azért is olyan népszerű, mert ezek minden nyelvi változatban megegyeznek, így ezek használatával egy Word makróvírus más nyelvű verziókon is tud szaporodni, mint amiben ki lett fejlesztve.

2. Word parancs átdefiniálással

A Word belső parancsai **olyan utasítások, amelyek a felhasználó tevékenységétől függően „felszólítják" a Word-öt, hogy végezzen el valamit:** pl. File mentéskor egy FileSave belső parancs hajtódik végre, aminek hatására az aktuális dokumentumot vagy sablont a Word lementi.

Ezek parancsok **átdefiniálhatóak**, ami azt jelenti, hogy ha létezik egy belső paranccsal azonos nevű makró, akkor az adott parancs végrehajtása esetén a makró fut le, az eredeti parancs végrehajtását már a makróból kell intézni, ha erre egyáltalán szükség van: pl. ha létrehozunk egy FileSave makrót, akkor állomány lementésnél csak ez a makró kapja meg a vezérlést.

A Word minden menüpontja, eszközsorból, az egyes eszközök vagy különböző billentyűzet-kombinációk szintén valamilyen parancshoz vannak rendelve. Így a megfelelő menüpont kiválasztásával, egy eszközgomb vagy egy billentyűzet-kombináció lenyomásával szintén egy parancs vagy egy azzal megegyező nevű makró végrehajtását érhetjük el. Az előbbi példánál maradva, ha létezik egy FileSave makró, akkor a Fájl menüből a Mentés-t kiválasztva, vagy az eszközsorból egy floppylemezt ábrázoló gombot vagy a Ctrl-S billentyűzet-kombinációt lenyomva ez a makró fog vezérlést kapni.

Ha egy elég sűrűn használt belső paranccsal megegyező nevű makrót tartalmaz a betöltött fertőzött (ál)dokumentum, akkor jó esély van rá, hogy a vírus aktivizálja magát. A belső parancsok átdefiniálását a Word makróvírusok nemcsak aktivizálódásra, hanem további dokumentumok megfertőzésére is használhatják.

A Word belső parancsai nyelvi változatonként eltérhetnek egymástól, ám meg lehet oldani, hogy – hasonlóan a **CAP makróvírus**hoz – egy makró

lekérdezze a Word nyelvi hovatartozását és ennek alapján megkeresse a helyi WordBasic parancsokat.

3. Billentyűzet átdefiniálással

A Word lehetőséget ad arra, hogy különböző billentyűzetkombinációkat tetszőleges parancsokhoz vagy tetszőleges makrókhoz rendeljünk hozzá. Ekkor a megfelelő billentyűk lenyomása után a hozzárendelt parancs vagy makró hajtódik végre.

Ennek a módszernek a segítségével – a fertőzött (ál)dokumentum betöltése után – kétféleképpen is aktivizálhatja magát egy makróvírus:

- a fertőzött sablon a Word-ben már létező billentyűzetkombinációhoz rendeli a vírus aktivizáló makróját és ezért a legközelebbi billentyűzeten keresztül kiadott parancsvégrehajtáskor nem az eredeti parancs hajtódik végre, hanem a makróvírus kódja indul: pl. ha az aktivizáló makró Ctrl-S-hez van rendelve, akkor billentyűzeten keresztül történő file mentéskor a vírus fog elindulni és nem a lementés,
- egyéb, gépelés közben sűrűn használatos billentyűhöz rendeli az aktivizáló makrót, pl. szóközhöz.

4. Menüpont átdefiniálással, beszúrással

A Word-ben mindenegyes menüpont valamilyen parancshoz vagy makróhoz van rendelve. Az egyes menüpontok könnyen „átszabhatóak" igényeinknek megfelelően, így megoldható, hogy új menüpontot szúrjunk be vagy egy már meglévő menüpontot is más parancshoz, makróhoz rendelhetünk. Ennek segítségével kétféleképp tud aktivizálódni egy makróvírus:

- a fertőzött sablon a Word-ben már létező, sűrűn használt menüponthoz (pl. Fájl|Mentés) rendeli hozzá az aktivizáló makrót és a legközelebbi menüpont kiválasztáskor ez a makró fog lefutni,
- egy új menüpontot szúr be a már meglévők közé, számítva arra, hogy a kíváncsi felhasználó esetleg azt kiválasztja: pl. **Fájl|Olvass el** menüpont eléggé figyelem felkeltő lehet...

5. Eszköztár átdefiniálással, beszúrással

Hasonlóan a menüpontokhoz az eszköztár ugyanúgy átalakítható és ezáltal is kétféleképp tud aktivizálódni egy makróvírus:

- a fertőzött sablon a Word-ben már létező eszköztárhoz (pl. Fájl|Mentés „lemez" gombjához) rendeli hozzá az aktivizáló makrót és az adott eszköztár gomb lenyomásakor ez a makró fog elindulni,
- egy új eszköztár gombot szúr be a már meglévők közé, számítva arra, hogy a kíváncsi felhasználó esetleg azt kiválasztja: pl. egy felkiáltó jel a lementés és a nyomtatás között elég feltűnő.

6. Makrógomb beszúrással

Makrógomb segítségével makróparancsot tudunk beszúrni egy dokumentumba: pl. magyarázó szöveget tudunk fűzni a dokumentum egy bizonyos pontjához úgy, hogy az adott helyre egy olyan makró gombját szúrjuk be, amit megnyomva a makró egy segítő párbeszédpanelt jelenít meg. Sajnos ez a lehetőség gonosz szándékkal kihasználható: a párbeszédpanelt megjelenítő makró helyett a vírust aktivizáló makró is elindítható. Ez egyúttal hátránya is a módszernek, hiszen a felhasználónak feltűnhet a dokumentum bármely részére a semmiből odakerült makrógomb.

Ezt az aktivizálódási módszert alkalmazó Word makróvírus eddig még nem jelent meg.

7. Űrlap mező beszúrással

A Word segítségével létrehozható űrlapokat vagy a képernyőn, vagy kinyomtatva tudunk kitölteni. A képernyőn megjelenő űrlap esetében a felhasználó az űrlapmezőkbe adatokat vihet be. Ezekhez a mezőkhöz olyan makrókat lehet rendelni, amelyek a megfelelő helyzetben (pl. mezőbe belépve, annak elhagyásakor, lezárásakor) automatikusan lefutnak. Ilyen mezőt a megfertőzött dokumentum tetszőleges helyére beszúrhat a vírus, de hasonlóan az előző ponthoz a felhasználónak itt is elég furcsának tűnhet egy semmiből előbukkanó űrlapmező. Ha azonban a vírus a dokumentum elejére szúr be egy ilyen mezőt, akkor semmiféle módszerrel nem akadályozható meg a makró lefutása: dokumentum megnyitáskor a Word a dokumentum elejére viszi a kurzort, így belép a mezőbe...

Ezzel a módszerrel működő vírus már létezik, melyet direkt egy makróvírus védelmi program kijátszására készítettek el, mivel az ezt a módszert még nem ismerte.

Az aktivizálódási mechanizmusokból természetesen egyszerre több is felhasználható, ekkor nagyobb a valószínűsége, hogy **vírus vezérlést** kap. Találkoztam már olyan makróvírussal, amely nem automakrós technikával

próbálkozott, mégis tartalmazott üres automakrókat. Ennek az volt a célja, hogy a Word-ben már rezidens, automakrókat is tartalmazó makróvédelmi makrókat felülírja.

Szaporodás

Egy Word makróvírus miután aktivizálta magát a fenti módszerek valamelyikével, „**rezidens**" vagy „**nem rezidens**" lesz. Ez utóbbi a Word makróvírusok körében azt jelenti, hogy keres egy vagy több (akár az összes) megfertőzhető dokumentumot, azokat megnyitásuk után (ál)dokumentum vagy csatolt sablonfile stílusban megfertőzi, majd bezárja. A vírus szempontjából probléma egyrészt több dokumentum fertőzése esetén a szükséges időmennyiség, másrészt az állományok megnyitása – bezárása feltűnő, éppen ezért ez a legritkábban alkalmazott fertőzési módszer.

Egy „rezidens" Word makróvírus aktivizálódásakor a fertőzött sablon makróterületéről saját makróit vagy a Word globális makróterületére – amit a Normal.dot-ban tárol, vagy a Word-ben beállítható „Startup" könyvtárban egy tetszőleges nevű sablon file-ban tárolt globális makróterületre másol át, feltéve, ha ezek a területek még nem fertőzöttek. Ha a normál sablonon változás történt, pl. makró rögzítés, akkor a Word ezt automatikusan lementi, feltéve, hogy ki van kapcsolva a Normal.dot lementés megerősítés (amit egy makróvírus meg tud és meg is szokott tenni), különben figyelmezteti a felhasználót az esetleges változásokra. A „Startup" könyvtáras „rezidens" technika alkalmazásakor előfordulhat, hogy az adott könyvtárban egyáltalán nem létezik sablon file, ilyenkor a vírus maga gondoskodik egy tetszőleges nevű, de .dot kiterjesztésű fertőzött sablonfile létrehozásáról, ha már létezik benne sablon file, és a vírus azt fertőzte meg, akkor a vírusnak kell gondoskodnia a változtatások lementéséről.

Ezeknek a sablonállományoknak a megfertőzése azért fontos a vírus részéről, mert ezáltal biztosítva van, hogy a Word indításakor a vírus makrói automatikusan
betöltődjenek és valamilyen módszerrel vezérlést szerezzenek. Egy alapvető jellemzőben található eltérés a normál sablon és az „Startup" könyvtárban található sablonok között: ha nincs automakró tiltás érvényben, akkor a Normal.dot-ban található tetszőleges automakró használatával a vírus aktivizálni tudja magát, ellenben a „Startup" könyvtárban található sablonfileokban lévő automakrók közül csak az AutoExec és az AutoExit hajtódik végre!

A *Normal.dot megfertőzésével* kapcsolatban fontos lenne megjegyeznem, hogy a Word minden egyes verziójánál más névvel lehet hivatkozni a globális makróterületre. Ez a probléma az első makróvírusok megjelenése-

kor annyiban jelentkezett, hogy egyes makróvírusok nyelvspecifikusak voltak, tehát csak azon a nyelvű Word-ön tudtak szaporodni, amelyben ki lettek fejlesztve: ha aktivizálódása során a vírus rosszul hivatkozott a globális makróterületre, akkor a felhasználó csak annyit látott, hogy a Word egy WordBasic hibára hivatkozott (ami eléggé feltűnő volt). Ezen problémák egy részét a leleményes vírusfejlesztők úgy oldották meg, hogy a vírus minden makrójából annyi és olyan példány szerepelt a víruskódban, amilyen Word fertőzésekre szánták: pl. angol és német Word-höz tartozó makrókat is tartalmaz az erre a két nyelvi változatra fejlesztett makróvírus. A következő fejlődési lépés az volt, hogy a globális makróterület nevét és a Word nyelvi hovatartozását is lekérdezte a makróvírus: ezek már tetszőleges nyelvi változaton képesek szaporodásra.

A vezérlés megszerzésétől kezdődően (tehát már amikor egy automatán betöltődő sablonra átmásolta magát) fertőzni fog – a fertőzést vezérlő makrótól függően – minden egyes megnyitott, szerkesztett, bezárt stb. dokumentumot, amiből a felhasználó nem vesz észre semmit. Egy dokumentum megfertőzése kétféleképpen történhet:

- **(ál)dokumentumos módszernél** a vírus a célpont dokumentumot átalakítja sablonná (**mindössze 1 utasítás!**), a sablon makróterületére másolja a vírusmakrókat, beállítva azokat úgy, hogy az (ál)dokumentum legközelebbi megnyitásakor abból aktivizálódni legyen képes, ha egy nem fertőzött rendszerrel találja magát szemben,
- **dokumentumhoz csatolt sablon file-os módszer**nél a dokumentumba csak egy csatolási megjegyzést tesz, majd attól függően hogy **CEB** vagy **FAT** stílusban fertőz, készít a dokumentumhoz egy sablonfile-t is, ami tartalmazza a vírus makróit.

A fertőzött dokumentumot pedig elegendő egyetlen egyszer megnyitni egy vírusmentes gépen, ha képes lesz vezérlésátvételre, akkor...

Kiegészítésként szeretném felhívni a figyelmet arra, hogy van még egy megoldás – amit eddig még egyetlen Word makróvírus sem használt ki, amivel sablonfile (és makró) betöltés Word indításkor automatikussá tehető: ha Winword.exe után paraméterként egy létező dokumentumot vagy sablont adunk meg, akkor azt a Word automatikusan betölti. További variációs lehetőség a /m paraméter használata, ekkor a paraméterben megadott tetszőleges helyen elhelyezett sablon file-t nemcsak betölti, hanem a beállított makrót automatikusan végre is hajtja a Word. Mindkét megoldás kihasználható gonosz szándékkal, hiszen parancssort képes módosítani egy Word

makróvírus, ráadásul ezzel a módszerrel lehetőség nyílik arra, hogy akár az automatikusan betöltődő sablonokból vagy egyéb sablonból makró induljon el – autoopen makró alkalmazása nélkül!

4.1.3. A Word makróvírusok objektív rutinjainak lehetőségei

A PowerPoint kivételével az összes Microsoft irodai alkalmazás (Word, Excel, Access) file-, lemez- és könyvtárkezelése, valamint programvezérlése fejlettnek tekinthető: kizárólag makróutasítások segítségével képesek vagyunk szekvenciális file megnyitására, lezárására, file-ból való olvasásra és írásra, file-ban való seek-elésre, file másolására, átmozgatására és átnevezésére, file attribútum lekérdezésére és módosítására, joker karakterek (*,?) felhasználásával file-ok közötti keresgélésre, aktuális könyvtár lekérdezésére, könyvtárak közötti mozgásra, könyvtár létrehozására és törlésére, hálózati meghajtó elérésére, nem kötelezően Windows programok indítására és leállítására, azokra való átváltásra, Windows háttérben futó alkalmazás ablak méreteit megváltoztatására (akár minimalizálni vagy maximalizálni), de akár a Task List-ról olyan alkalmazások eltűntetésére, amelyek éppen futnak (félig-meddig titkosítható egy program futása).

Ráadásul ez a sok lehetőség az egyes alkalmazásokon belül már elérhető volt az Office 97-ben egyesített makrónyelv (VBA) megjelenése előtt: Word-ben a WordBasic illetve az Access-ben az AccessBasic makrónyelv ugyanígy bírt ezekkel a tulajdonságokkal, nem is beszélve az Excel-ben kezdetektől fogva használt VBA-ról.

Mindezekből látható, hogy gyakorlatilag nemcsak Word, hanem Excel vagy egy Access makróvírus **komoly pusztítást vihet véghez:** törölhet és/vagy felülírhat fontosabb rendszerállományokat, könyvtárakat vagy bármely egyéb állományokat, de akár le is formázhatja a winchestert, anélkül, hogy ezt a felhasználó meg tudná akadályozni (**Format parancs elindítása /U /AUTOTEST paraméterekkel), tetszőleges bináris – akár hardver pusztító -rutinokat is végrehajtathat a géppel (a makróvírus által létrehozott szöveges állomány felhasználása Debug bemenetként, mint az ascii és a hibrid batch vírusoknál**), ami esetleg egyéb vírus is lehet... Tehát gyakorlatilag egy makróvírus ugyanolyan mértékű pusztítást tud csinálni, mint bármely egyéb bináris társa.

Ugyanakkor a bináris kódok és a külső programok használata valamennyire rendszerspecifikussá tesz egy makróvírust, hiszen pl. egy PC-s Bios rutin meghívása Macintosh gépeken nem lehetséges, vagyis a Microsoft irodai programjai közül a Macintosh verziót használók ezáltal jobban vannak védve a makróvírusok károkozásaival szemben.

A szándékos károkozás nemcsak a rendszer fontosabb elemei, hanem az adott alkalmazás által létrehozott állományok felé is irányulhat: elég hatásos tud lenni, ha a makróvírus állomány lementéskor azokat jelszóval védi le, aminek eredményeképpen a felhasználó – a jelszó hiányában – nem tud majd hozzáférni anyagához. Erre a problémára azonban létezik megoldás: az Internet-en számtalan helyen található Word vagy Excel jelszóvédelem-feltörő program, igaz, az Internet nem mindenki számára elérhető.

Kanyarodjunk vissza egy picit a Word makróvírusok objektív rutinjának vizsgálatához: kisebb fajta pusztításnak számít – de ettől még nem megbocsátható bűn, ha a Word makróvírus bizonyos szavakat kicserél más szavakra vagy tetszőleges helyre különböző szavakat szúr be a dokumentumban, aminek javításához egy terjedelmesebb dokumentumnál hosszabb ideig eltartó kézi munka szükséges.

Mint a legelején említettem, makrórögzítés esetén a Word az adott verzió makrónyelvén írt utasítások sorozataként tárolja el a makrót, így makrórögzítés segítségével is lehet pusztító rutinokat írni: pl. egy komplett dokumentum kijelölése után az egyes karakterek színének a háttér színére változtatásával – és ezen folyamat rögzítésével – egy közvetetten pusztító makrót sikerült létrehozni, mert annak lefuttatásakor a felhasználó azt hiszi, hogy üres dokumentummal van dolga és esetlegesen törli vagy felülírja azt.

Nem romboló szándékú makróvírus – amit most újra Excel és Access makróvírusra is értek – megjeleníthet bármilyen üzenetet vagy grafikus ábrát a képernyőn, poénosabb képernyővédőt indíthat el vagy átállíthatja a Windows különböző beállításait (pl. a színeit a WW.Color vírushoz hasonlóan), Word-nél a nyomtatandó szövegbe tetszőleges helyre beleszúrhat valami megjegyzést (mint pl. a WW.Nuclear).

Végül egy nagyon fontos megjegyzés: Word-ben és Excel-ben az objektív rutin rész a vírus aktivizálódása nélkül is lefuthat abban az esetben, ha az objektív rész egy külön makróban van, ekkor parancs vagy billentyűzet átdefiniálási, menüpont vagy eszköztár átdefiniálási, beszúrási, makrógomb – nyomógomb vagy űrlap mező beszúrási módszerrel külön elindítható ez a makró. Ez akkor is megtörténhet, ha pl. nyelvi probléma miatt nem tud szaporodni a makróvírus, tehát nem aktivizálható vírus is képes lehet rombolásra! Ezt alkalmazzák a trójai makrók is: az aktivizálódáskor csak pusztító makró indul el.

Az objektív rész aktív vírus nélküli lehetséges lefutására az egyik legjobb példa a WW.Nuclear vírus: ha a vírus nem aktív, de a fertőzött (ál)dokumentumból nyomtatni szeretnénk, akkor a nyomtatás legvégén a francia atomrobbantások elleni propaganda szöveget olvashatunk! Gondol-

junk bele, ha a gépünkkel egy fontos szerződést fax-on (nyomtatáson) keresztül szeretnénk elküldeni, akkor a címzett legnagyobb csodálkozására a túloldalon megjelenne a propaganda szöveg...

4.1.4. Lopakodó Word makróvírusok

A **lopakodás első lépéseként** a normál sablont is fertőző Word makróvírusok a Word-ben a **Normal.dot** lementés megerősítést kikapcsolták. Erre azért volt szükség, mert a Word alapértelmezésben megerősítést kér a Normal.dot lementésekor, ha azon bármi változás történt (vírusmakrók átmásolása globális makróterületre).

Az **első lépéshez** sorolnám a vírusfejlesztők két másik ötletét, amelyeket mind a mai napig nem jellemzően alkalmaznak: 1

- a vírus makróit **félrevezető nevekkel** látják el – pl. Windows Help, így a makrólistában a kevésbé ravasz felhasználónak nem tűnik fel az idegen makrónév,
- a vírus bekapcsolja a **gyors mentést**, ezáltal a Word csak a dokumentumokon végrehajtott változtatásokat menti. Ez a fajta mentés gyorsabb, mint a teljes mentés, viszont a Word annyira „teleszemeteli" az így lementett állományokat, hogy a sablonokban a gyengébb antivírus programok nehezebben vagy egyáltalán nem képesek makrók keresésére vagy irtására.

Második lépésként a fejlesztők vírusaik makróit un. execute-only attribútummal látták el: azok a makrók, amelyek ezzel a jellemzővel bírnak, védve vannak mindennemű megtekintéstől vagy módosítástól, kizárólag futtathatóak, törölhetőek, átnevezhetőek vagy átmásolhatóak az egyik sablonról a másikra (tehát a vírus továbbra is szaporodóképes lesz) és a Word titkosított formában tárolja ezeket a makrókat a sablon file-okban (ez a titkosítás visszafejthető). Az execute-only attribútumos megoldást azért kezdték el alkalmazni, mert egyrészt így egyszerűbb eszközökkel senki nem lát bele, hogy az adott makró mit cselekszik, másrészt a *vírusfejlesztők nem igazán szeretik, ha az általuk fejlesztett vírusnak* „**véletlenül" más „copyright"-os változata is megjelenik.** Ez a lehetőség azonban kizárólag a Word 6.0-ban és a Word 95-ben használható, mivel Word 97-ben csak egész projektet lehet védeni, ami miatt egy makróvírus fertőzésképtelen is lehet... 2

A **harmadik lépés** megtétele szintén egyértelmű: azon menüpontok feletti vezérlés átvétel, amelyeken keresztül a makrók elérhetőek illetve törölhetőek még akkor is, ha az egyes makrók execute-only tulajdonsággal bír- 3

nak. A legelső ilyen vírusok az **Office 97** előtti Word-nél az **Eszközök|Makró** és/vagy a **File|Sablonok|Szervező|Makró** menüpontot, **Word 97**-ben az **Eszközök|Makró** és/vagy a **Nézet|Eszköztárak|Visual Basic** menüpontot tüntették el (egyszerű menüpont törléssel), későbbi változataikban a makró elérő menüpontokat nem törölték a többi menüpont közül, hanem hozzárendeltek egy vírusmakrót, ami a menüpont kiválasztásakor:

- semmi eredményt nem adott vissza: a menüpont látható volt a többi között, csak nem szolgált semmire, vagy
- hibajelzést adott vissza: mintha – pl. egy felhasználói belepiszkálás eredményképpen – nem lenne elérhető a menüpont, vagy
- láthatóvá tett minden makrót úgy, hogy csak a vírusmakrókat nem lehetett átnevezni vagy törölni, vagy
- láthatóvá tett minden makrót úgy, hogy csak a vírusmakrók nem szerepeltek a listában, mintha nem lenne fertőzött a rendszer.

Ezzel a lopakodási technikával azt sikerült kiszűrni, hogy a felhasználónak feltűnjön a fentiekben említett menüpontok eltűnése, még akkor is, ha azokat nem vagy csak nagyon ritkán használja. Természetesen nem kell azt gondolni, hogy minden ma megjelenő lopakodó makróvírus alkalmaz ilyen menüpont elérés átverő technikát.

4

Nem olyan régen a lopakodó Word makróvírusoknak egy egészen furcsa fajtája jelent meg: makróit a dokumentum gyorsszöveg részében tároló makróvírusok. A **Word**-ben használható gyorsszöveg (autotext) a később újra felhasználni kívánt szövegelemek (pl. gyakran használt címek, szerződésekben rendszeresen használt mondat) vagy ábrák (pl. embléma) tárolására szolgáló hely. Ezen a területen a vírus főbb makrói, mint gyorsszöveg bejegyzés vannak tárolva, a makróterületen legalább egy vírus aktivizáló makró található – különben a gyorsszövegben elrejtett utasítások semmiképpen sem kapnának vezérlést –, aminek az a feladata, hogy vezérlés megszerzés esetén a gyorsszövegből a makrókat átmásolja a makróterületre. A makróterületre másolt makrók intézik, hogy a célpont dokumentumok **gyorsszöveg részébe belekerüljenek** a megfelelő makrók illetve a makróterületre az aktivizáló makró. Mivel az aktivizáló makró nemcsak (ál)dokumentumba, hanem csatolt sablon file-ba is belekerülhet, ez további variációs lehetőséget ad a „**gyorsszöveges**" vírusmakrók elhelyezésére, hiszen azok csatolt sablonos módszer esetén az eredeti dokumentumba vagy a csatolt sablonba is belekerülhetnek (tehát a vírus akár eldöntheti, hogy főbb részeit hol tárolja).

n+1

Véleményem szerint ehhez **hasonló lopakodási technika** megvalósítható lenne a dokumentum szöveges (amit a felhasználó közvetlen szerkeszt)

részével is: a vírus makróinak utasításai szövegként lennének tárolva (pl. a dokumentum végén), ugyanolyan színnel, mint a háttér szín, vagyis szinte „**láthatatlan**" lenne, az aktivizáló makró a szöveg megfelelő részét másolná a makróterületre.

Az autotext részben szaporodó makróvírusok két „**előnyt**" egyesítenek magukban: egyrészt a *Word gyorsszöveges lehetőségeit kevésbé kihasználó felhasználó nem veszi észre* a vírus jelenlétét, másrészt ezen módszer kifejlesztéséig a heurisztikus makróvíruskereső programok többsége csak a sablon file makróterületét vizsgálta...

4.1.5. Polimorf Word makróvírusok

A **Word a sablon file**-okban minden makróutasítást tokenizáltan tárol: minden utasításhoz egy kód van rendelve, sablonmentés esetén csak a makrók neve és az egyes makrókon belüli utasításokhoz tartozó kódok lesznek beleírva az állományba. Az első makróvírusok elleni programok ezt a tudást úgy próbálták kihasználni, hogy az addig megjelent makróvírusok makróneveit illetve a makróikon belül található utasítás (kód) sorozatokat keresték a megfertőzhető vagy megfertőzött állományokban. Az execute-only makrók primitíven kódolt tokenizált formában vannak tárolva, a visszakódoláshoz szükséges kulcsot maga az állomány tartalmazza, így ezeknek a visszafejtése sem okozhatott problémát az első víruskeresőknek.

Erre válaszul a vírusírók **kifejlesztették polimorf Word makróvírusaikat**, amelyekben – a mutációt intéző kódtól függően – már nehezen vagy egyáltalán nem lehet makrónevek és makróutasítások után kutatni. Ellentétben viszont bináris társaikkal itt nincs külön decryptor és kódolt rész, másrészt a polimorf bináris vírusok jellemzően csak a decryptort mutálják, míg egy polimorf Word makróvírus az összes makróját változtathatja.

Egy **polimorf Word makróvírus**on belül a **mutációs rutin** az alábbi változtatásokra képes:

- **véletlenszerűen generált karaktersorozatok**, számok beszúrása a makrókódba véletlen helyekre, megfelelően kikommentezve, hogy hibával ne álljon le egy makró futása
- véletlen számú üres komment beszúrása a makrókódba véletlen helyekre,a kikommentezésre szolgáló „**REM**" és a ' (szimpla idézőjel) folyamatos cserélgetése,
- véletlen számú **üres sor beszúrása** a makrókódba véletlen helyekre,
- egyéb **semleges utasítások beszúrása** a makrókódba véletlen helyekre,

Σ

- makrókódban használt **változók nevének megváltoztatása**,
- feltételes elágazások megváltoztatása,
- egymással szorosan össze nem függő parancsok cserélgetése,
- utasítás(ok) lecserélése másik, de eredményében ugyanolyan utasítás(ok)ra,
- a permutációs vírusokhoz hasonlóan **makrón belüli szubrutinok felcserélése**,
- a víruson belüli **makrók sorrendjének változtatása**,
- makrók átnevezése olyan módon, hogy azok egymást továbbra is meg tudják hívni.

Sokan hiányolhatják az egyes mutációs lépések részletesebb kivesézését, példák írását – mint azt tettem a bináris polimorf vírusoknál – de a polimorf Word makróvírusok szinte ugyanazokat a változásokat képesek eszközölni önmagukon, mint bináris társaik, tehát nem nehéz elképzelni egy ilyen vírus önálló „életét", ezenkívül egy magyar nyelvű Word help-ből a bináris példák alapján könnyen kitalálható Word makróbeli megfelelőjük.

4.1.6. Word makróvírusok korlátai Word 97 környezetben

Bár az alábbiakat a Word makrónyelvi jellemzőinél kellett volna leírnom, a Word makróvírusok működésének ismerete nélkül nem lett volna megérthető.

Mint ott említettem, a **Word 97 a WordBasic**-ben készített makrókat automatikusan **átkonvertálja VBA**-ra. Nem teljesen publikus, de a *Word 97 nemcsak átkonvertál, hanem vírust is ellenőriz: a legjellemzőbb WordBasic-es makróvírusok (pl. Concept, Wazzu) szekvenciáit keresi a konvertálandó makrókban, találat esetén a vírushoz tartozó makrókat le sem fordítja, de erről semmiféle értesítést nem ad a felhasználónak*! Miután tudva levő, hogy egy ilyen védelem – legjobb esetben – a Word 97 megjelenésekori makróvírusokat ismerheti csak, rögtön felvetődik az emberben a kérdés, hogy akkor a Microsoft miért nem teszi lehetővé a Word 97 víruskereső részének folyamatos frissítését? A válasz roppant egyszerű: ilyen frissítésekre a Microsoft (sajnos) nem hajlandó, pedig nem lenne bonyolult, hiszen az ellenőrzést tartalmazó Wwintl32.dll állomány legfrissebb verzióit kellene minden lehetséges úton terjeszteni.

A lopakodó vírusok execute-only makrós tulajdonságánál említettem, hogy Word 97-ben kizárólag egész projektet lehet védeni. További jellemzője a Word 97-nek, hogy védett projektből nem lehet nem védettbe makrót

másolni: ez azt jelenti, hogy ha a **Normal.dot nem védett (általában nem az)**, akkor az execute-only technikát és a Normal.dot-t támadó makróvírusok Word 97 alatt egyszerűen nem lesznek fertőzőképesek!

Az **Office 97 Service Release 1 (SR1)** javítása óriási áttörést hozott a makróvírus fronton: a javítás után VBA-ban nem lehetséges a WordBasic sablonból sablonba makrót átmásoló parancsok (Macrocopy és OrganizerCopy) segítségével globális sablonból (ál)dokumentumba makrókat másolni és ez a korlátozás nem kikapcsolható.

Ez azt jelenti, hogy ha a makróvírus **képes is megfertőzni a globális sablont,** onnan további szaporodásra nem lesz képes. Sablonból globális sablonba való másolás azért sem tiltott, mert ezt használják telepítéskor a hasznos makrók (mert vannak ilyenek is).

Az örömbe némi üröm is vegyül: megjelentek már azok a Word makróvírusok is, amelyek az SR1 javítását képesek kijátszani úgy, hogy a vírus modulját szöveges file-ba exportálják, majd a megfertőzendő dokumentumba ezt a file-t importálják. Ilyen vírus pl. a Strage Days.

4.2. EXCEL MAKRÓVÍRUSOK

4.2.1. Az Excel makrónyelvi jellemzői

Ugyanúgy mint a Word-ben, az Excel-ben is a makrók műveletek egymásutánja, csak itt már az 5.0-ás verziótól kezdődően *VBA nyelven lehet rögzíteni vagy szerkeszteni azokat, melyek eszköztárhoz, gyorsbillentyűkhöz vagy menüponthoz rendelhetőek.* Természetesen a hozzárendelés itt sem kötelező, ekkor egy külön menüpontból – Excel 5.0-ban Eszközök|Makró – tudjuk futtatni a számunkra fontos makrót.

Az Excel-ben is léteznek sablonok, melyeket azonos felépítésű munkafüzetek létrehozásakor mintaként használhatunk: tartalmazhatnak szöveget, grafikát, formátumokat, oldalbeállításokat, képleteket és makrókat. Az Excel indulásakor az összes .xls állományt betölti – mint automatikus sablon – az indító (Startup) és a másodlagos indító könyvtárból. Alapértelmezés szerint az indító könyvtár az Excel könyvtárában található \Xlstart alkönyvtár, a másodlagos indító könyvtárat a felhasználó állíthatja be. Természetesen az elsődleges „Startup" könyvtár is megváltoztatható.

Az Excel-nek azonban nem létezik olyan szintű automatikus sablonja, mint a Word-nek a normál sablon, mert

- a felhasználó által rögzített vagy létrehozott makrókat az Excel alapbeállítás szerint az aktuális munkafüzet Visual Basic (VB) modul lapjára rögzíti,
- a Word saját normál sablonját – ha az nem létezik – első futtatása után automatikusan létrehozza, ellenben az Excel ezt – az Xlstart alkönyvtárban Personal.xls filenéven – csak akkor teszi, ha a felhasználó az un. egyéni makró-munkafüzetre rögzíti az általa létrehozott makrókat.

Vagyis elméletileg az **Excel globális makróterület tároló állományának a Personal.xls file-t tekinthetjük,** hiszen azt létrehozása után az Excel minden indításakor automatikusan betölti (igaz ezt más állományokkal is megteszi).

Az Excel lementéskor a sablon állományoknak .xlt, a munkafüzet fileoknak .xls kiterjesztést ad és hasonlóan a Word-höz kiterjesztéstől függetlenül felismeri, hogy milyen állománnyal van dolga. Az (ál)dokumentumos fertőzést alkalmazó Word makróvírusokkal ellentétben az Excel makróvírusoknak ezt a lehetőséget nem kell kihasználniuk, mert – hasonlóan a sablonokhoz – egy munkafüzet is tartalmazhat makrókat. A két fajta állomány közötti különbség azok betöltésekor fedezhető fel: egy sablon megnyitásakor az Excel létrehozza a sablon másolatát és azon lehet dolgozni – így az eredeti sablon változatlan formában később is rendelkezésre áll, egy munkafüzetet pedig közvetlen megnyit. Ez az apró különbség azonban a makrók betöltődését és esetleges végrehajtását nem befolyásolja.

Az Excel-ben található egy mélyebb szintű makróbeépítési lehetőség is, amellyel az automatikus sablonokhoz hasonlóan megoldható, hogy makrók Excel elindítása után automatikusan rendelkezésre álljanak: az Eszközök|Makróbeépítő menüponton keresztül telepíthetőek vagy távolíthatóak el az .xla kiterjesztésű un. beépülők (add in application), melyek új parancsokkal és függvényekkel egészítik ki az Excel-t. Ezekre a beépülőkre jellemző, hogy egészen eltávolításukig rendelkezésre állnak az Excel-ben, s bár a bennük található makrók futtathatóak, azok neve nem látható az Excel makró elérő menüpontjaiból.

Szintén nem elhanyagolható szempont, hogy az Excel-nek 4 verziója is használatos manapság, az Excel 4.0, Excel 5.0, Excel 95 és az Excel 97. Az egyes verziók közötti makrónyelvi eltérések az alábbiak:

- Az Excel 4.0 és az Excel 5.0 között lett lecserélve az **Excel saját makró nyelve VBA**-ra, valamint itt változott a 4.0 verzióban hasz-

nálható globális makrólap egyéni makró munkafüzetre, tárolás szempontjából **Global.xlm**-ről **Personal.xls**-re, csak felhasználhatóságuk lényege nem változott.

- Az Excel 5.0 kezdődően az **összes verzió makrónyelve a VBA lett,** makrónyelvezetük a felülről való kompatibilitás jegyében lett tovább fejlesztve: egy alacsonyabb verziójú Excel-ben megírt makró biztosan működik egy magasabb verziójúban, de ez fordítva már nem biztos, hogy igaz.
- Az Excel 5.0 és annál nagyobb verzióinak állományfelépítése eltér az Excel 4.0-étól, ezért 4.0 formátumban való file lementéskor az Excel 5.0 vagy annál nagyobb verziói **csonkolják a file-t (nem mentik le a makrókat).**

Egyrészt az makrónyelv cseréje miatt (1. pont), másrészt állomány felépítések eltérése miatt (3. pont) a legalább Excel 5.0 alatt futó Excel makróvírusok Excel 4.0-ban és Macintosh Excel-ben sem fognak futni.

Az Excel nyelvi változatai között is található különbség, az egyes változatokon fejlesztett makrók fordítástól függően képesek más nyelvi változatokon futni: pl. első Excel makróvírusként a „Laroux" francia Excel-ben nem futott.

4.2.2. Az Excel makróvírusok működése

Betöltés

Mivel az Excel-ben minden állomány tartalmazhat makrót, ezért egy makróvírusnak teljesen mindegy, hogy **munkafüzetet, sablont vagy éppen egy beépülőt fertőz meg, állomány átkonvertálásra nincs szüksége**. Ettől függetlenül előfordulhat, hogy a felhasználó – előzőleg munkafüzetként használt – fertőzött állományt átkonvertál más formátumba, de ez a Excel makróvírusok működését többnyire nem befolyásolja.

Vezérlés átadás

Hasonlóan, mint a Word makróvírusoknál, induljunk ki abból, hogy betöltöttünk egy Excel makróvírus által fertőzött állományt és **a vírusnak valahogy meg kell szereznie a vezérlést**. Ez a következőképpen történhet meg:

1. Automakrók használatával

Itt is van lehetőség automakrók használatára, amelyek valamilyen tevékenység végrehajtásakor automatikusan lefutnak, ha az automata sablonok bármelyike vagy az éppen betölteni kívánt Excel állomány tartalmazza az alábbi makrók bármelyikét. Az

- Auto_Open: Excel file megnyitásakor,
- Auto_Close: Excel file bezárásakor,
- Auto_Activate: adott Excel file aktívvá válásakor,
- Auto_Deactivate: adott Excel file inaktívvá válásakor,

Az eddig megjelent Excel makróvírusok többsége az Auto_Open és a Auto_Close adta lehetőségeket használták ki.

2. Gyorsbillentyű átdefiniálással

Ennek az aktivizálódási módnak a lényege megegyezik a Word-nél írtakkal, de:

- az Excel 5.0 nem engedte meg a *már meglévő billentyűzetkombinációk hozzárendelésének felüldefiniálását* – pl. Ctrl-S-hez nem lehetett makrót rendelni, mert ez már használva volt file lementéshez, az Excel 97-ben már igen,
- a billentyűzetkombinációban a **Ctrl+karakter** vagy a **Ctrl+Shift+karak-ter kombináció használható csak, ahol a karakter kizárólag ékezet nélküli betű lehet,** vagyis a Word makróvírusoknál alkalmazott gépelés közben sűrűn használt billentyűhöz való hozzárendelés nem működik.

Ha ezt a technikát alkalmazó makróvírus az első pontot figyelmen kívül hagyja, akkor hibaüzenetet fog visszaadni az Excel, de – hogy örömünk ne legyen teljes – ez a hiba kiküszöbölhető: ha jobban megfigyeljük, akkor a már létező billentyűzetkombináció hozzárendelések kizárólag nagy betűvel vannak írva. Ez azt jelenti, hogy ha egy makróhoz az Excel-ben már használt hozzárendelés kis betűs változata van rendelve és billentyűzeten a kis betűk az érvényesek (CapsLock kikapcsolva és Shift sincs nyomva), akkor az Excel-beli billentyűzet kombináció lenyomása esetén a vírus vezérlést kap.

3. Menüpont átdefiniálással, beszúrással

Bár az Excel dokumentációk és help-ek határozottan állítják, hogy makrókat Excel-ben kizárólag az Eszközök menün belül egy új menüponthoz lehet csak rendelni, gyakorlati tapasztalataim szerint tetszőleges menüponthoz rendelhető egy már létrehozott makró (a VB modulon belül Eszközök|Menüszerkesztő menüpontban), így gyakorlatilag ez a módszer teljesen megegyezik a Word-nél említettel.

4. Eszköztár átdefiniálással, beszúrással

Ez az aktivizálódási lehetőség megegyezik a Word makróvírusok ugyanezen módszerével.

5. Nyomógomb beszúrással

Az Excel-ben használható nyomógomb ugyanazt a szerepet tölti be, mint Word-nél a makrógomb és sajnos ugyanúgy kihasználható hátsó szándékkal.

6. Esemény vezérelt eljárások felhasználásával

Az Excel bizonyos események bekövetkeztekor az adott eseményhez rendelt eljárásnak adja a vezérlést. Ezek az eljárások és események az alábbiak lehetnek:

- OnTime: adott dátumon és/vagy időpontban,
- OnSheetActivate: Excel állományban adott lap kiválasztásakor,
- OnWindow: Excel-en belüli ablak kiválasztásakor,
- OnKey: meghatározott billentyű lenyomásakor,
- OnCalculate: adott oldalon újraszámolás esetén,
- OnEntry: adott oldalon adatbevitel esetén,
- OnData: más alkalmazástól való adatérkezéskor,
- OnRepeat: Szerkesztés|Ismét menüpont kiválasztásakor,
- OnUndo: Szerkesztés|Visszavonás menüpont kiválasztásakor.

Ha egy makróvírus kódja tartalmaz egy ilyen eljárást, akkor abból az aktivizáló kód is indítható.

Az aktivizálódási mechanizmusokból itt is felhasználható egyszerre több, ekkor a vezérlés megszerzésére a vírusnak nagyobb az esélye.

Szaporodás

Egy Excel makróvírus sem tud másképp viselkedni mint egy Word makróvírus, vagy „rezidens" lesz vagy nem. A nem rezidens viselkedés is ugyanazt jelenti Excel vírusoknál: keres egy vagy több (akár az összes) megfertőzhető célpontot, azokat megnyitásuk után megfertőzi, majd bezárja. A módszer jellemzői ugyanazok: időigényes és feltűnő a fertőzés.

Gyakorlatilag a „rezidens" Excel makróvírusok ugyanúgy működnek aktivizálódásukkor, mint Word makróvírus társaik: a fertőzött állományból vagy az egyéni makró-munkafüzetre (**Personal.xls**) vagy egy – akár elsődleges, akár másodlagos indító könyvtárban található – automatikus sablon VB modul lapjára másolják magukat, feltéve, ha ezek a területek még nem fertőzöttek. Ha az egyéni makró-munkafüzeten változás történt – pl. makró rögzítés, akkor a Excel ezt automatikusan egy Personal.xls állományba menti le, feltéve, hogy ki van kapcsolva a lementés megerősítés (amit egy makróvírus meg tud és meg is szokott tenni), különben figyelmezteti a felhasználót az esetleges változásokra. Egyéb indító könyvtáras „rezidens" technika alkalmazásakor ha nem létezik sablonfile a könyvtárban, akkor a vírusnak kell gondoskodnia egy tetszőleges nevű, de .xls kiterjesztésű fertőzött sablonfile létrehozásáról, ha már létezik benne sablon file és a vírus azt fertőzte meg, akkor a vírusnak kell gondoskodnia a változtatások lementéséről. Ezen sablonállományok megfertőzésének egyértelműen az a célja, hogy Excel indításkor a vírus makrói automatikusan betöltődjenek és valamilyen módszerrel vezérlést szerezzenek.

Az Excel makrónyelvi jellemzésénél említett un. beépülők felhasználása gonosz szándékkal is megvalósulhat: a vírus vagy egy már létező beépülőt fertőz meg, vagy létrehoz egyet és beállítja, hogy azt az Excel indításakor automatikusan betöltse – ezáltal a vírusnak még automatikus sablont sem kell fertőznie.

A vezérlés megszerzésétől kezdődően (tehát már amikor egy automatán betöltődő sablonra átmásolta magát) fertőz – a fertőzést vezérlő makrótól függően – minden egyes megnyitott, szerkesztett, bezárt stb. *Excel állományt, amiből a felhasználó nem vesz észre semmit. A fertőzött dokumentumot pedig elegendő egyetlen egyszer megnyitni egy vírusmentes gépen, ha képes lesz vezérlés átvételre, akkor...*

Hasonlóan a Word-höz az Excel-nél is van arra lehetőség, hogy egy létező Excel állományt, mint paramétert használjuk fel az Excel indításakor – az Excel automatikusan befogja tölteni, ellenben itt nem lehet automatikussá tenni az állományban található makró futtatását parancssorból.

4.2.3. Az Excel makróvírusok objektív rutinjainak lehetőségei

A Word makróvírusok objektív rutinjainak lehetőségeinél részletesen elemeztem, hogy milyen pusztításokat tudnak véghez vinni az egyes Microsoft irodai programok makróvírusai, valamint a Word-nél említett közvetett pusztító (karakterek színének megváltoztatása háttérszínre) makró rögzítése, és a kisebb pusztítások körébe sorolt szó és érték cserélgetés itt is lehetséges.

4.2.4. Lopakodó Excel makróvírusok

Az első lopakodó Excel makróvírusok – Word-ös társaikhoz hasonlóan – ***kikapcsolták* az egyéni makró-munkafüzet mentésének** megerősítését. 1

Második megoldásként az Excel-ben alapértelmezés szerint elérhető „elrejtés" lehetőséget használták ki: munkafüzeteket és a munkafüzet lapokat (VB modul lapokat is) lehet elrejteni a nem kívánt módosítások elkerülése érdekében. Az elrejtés által a makrók továbbra is elérhetőek, de a füllapok között nem lehet látni a makróvírus VB modul lapját. Ennél mélyebb szintű elrejtés is lehetséges: *a VB-ben létezik olyan utasítás, amellyel úgy rejthető el lap, hogy azt a felfedés paranccsal sem lehet láthatóvá tenni*! 2

Harmadik lopakodási lehetőség teljesen megegyezik a Word-nél említett harmadik lépéssel: makrókat elérő és felfedés/elrejtés menüpontok feletti vezérlés átvétel. 3

A **negyedik**, eddig még nem alkalmazott lopakodási lehetőség az – automata betöltéskor már említett – **un. beépülők használata**. Ha ugyanis a vírus nem szabványos munkafüzetként vagy sablonként fertőzi meg a célpontokat, hanem beépülőként, akkor a vírus makrói továbbra is futásképesek, de azok nevei nem láthatóak az Excel makró elérő menüpontjaiból. 4

4.2.5. Polimorf Excel makróvírusok

Bár az Excel állományok felépítése gyökeresen eltér a Word által használttól, ugyanúgy minden makróutasítást tokenizáltan tárol. Vagyis ezekben az állományokban is lehetett az addig megjelent makróvírusok makrónevei illetve a makróikon belül található utasítás (kód) sorozatok után keresni, egészen a polimorf típusok megjelenéséig.

A polimorf Excel makróvírusok tulajdonságai és a makrókódjukban bekövetkezhető változások megegyeznek a Word-nél írtakkal.

4.3. ACCESS MAKRÓVÍRUSOK

Ugyanazt jelentik Access-ben a makrók, mint Word-ben vagy Excel-ben, csak itt kizárólag szerkesztés útján, Access 97 előtti verzióikban AccessBasic, Access 97-ben VBA nyelven lehet létrehozni azokat.

Access makróvírus vizsgálati szempontból nem is igazán fontos, hogy az egyes makrónyelvek illetve nyelvi változatok között milyen különbség van, hiszen Access használata nem jellemző az általa létrehozott .mdb állományok cseréje – helyi hálózaton, Internet-en e-mail-ben – még inkább nem (kevés a potenciális célpont), ezenkívül **az Access-nek rengeteg olyan programozási korlátja van, ami miatt szinte nem érdemes Access makróvírust fejleszteni:**

- *makrókat kizárólag előre meghatározott billentyűzetkombinációkhoz lehet rendelni, menüpontokhoz nem,*
- *összesen 2 féle automakróval rendelkezik, ezek közül az Autoexec adatbázis megnyitásakor kap vezérlést, az Autokeys alapértelmezés szerint billentyűzet átdefiniálásra szolgál – szintén adatbázis megnyitásakor, de hátsó szándékkal is felhasználható,*
- *nem rendelkezik globális vagy automatikus sablonokkal, ahol makrók vannak tárolva és indításkor betöltve.*

Mivel egy előre meghatározott billentyűzetkombináció felhasználásnak egy adatbázis megnyitás után kicsi a valószínűsége, ezért nem is igazán kell figyelembe venni az 1. pontban írt aktivizálódási módot. Ezen ok és a 2. pont miatt **egy Access makróvírus kizárólag fertőzött állomány megnyitásakor tud aktivizálódni valamelyik automakró** (többnyire az Autoexec) segítségével, a legutolsó pont miatt **kizárólag „nem rezidens" technikával fertőzhet**. Ez azt jelenti, hogy ha egy Access makróvírussal fertőzött adatbázist megnyitunk, akkor közvetlenül a megnyitás után keres egy vagy több (akár az összes) megfertőzhető célpontot, azokat megnyitja, megfertőzi majd bezárja. Vagyis ugyanolyan **időigényes** és feltűnő ez a fertőzési módszer, mint a „nem rezidens" Word és Excel vírusoknál. Pontosabban időigényesebb: Access adatbázisokat egy átlag felhasználó sokkal ritkábban használ, így akár egy lehetséges célpont keresése az egész lemez átnézését igényli, ami túlságosan feltűnő lehet – egy fertőzött állomány megnyitása több percig is eltarthat!

Bár az Access makróvírusok szaporodási formái enyhén szólva is korlátozottak, pusztítási lehetőségük jóval szélesebb körű: a Word makróvírusok objektív rutinjainak elemzésénél részletesen elolvasható, hogy egy Access makróvírus mire is képes...

Megjelentek „lopakodó" Access makróvírusok is (!!), melyek az Access makró elérő menüpontjait törlik. Ne felejtsük el, hogy mivel menüponthoz nem rendelhető makró, ennél fejlettebb lopakodásra nem lesznek képesek.

Mindezek után azt hiszem, elég érthető, hogy az eddig megjelent összes Access makróvírus szám nem éri el a 20-at!

4.4. AMI PRO MAKRÓVÍRUSOK

Az **Ami Pro**-ban található makrók célja ugyanaz, mint bármely előző pontban említett programnál, de itt a dokumentumokhoz tartozó makrók a dokumentum file névvel megegyező, de .smm kiterjesztésű állományban található. Amikor egy dokumentumot a felhasználó betölt, akkor a szövegszerkesztő a dokumentumhoz tartozó .smm kiterjesztésű állományt (ha létezik) betölti. Ezáltal jön létre a kapcsolat a dokumentum és a hozzá tartozó makró állomány között.

Ami Pro alatt is léteznek automakrók, valamint lehetőség van menüpont átdefiniálásra és beszúrásra. Tehát jóval több aktivizálódási lehetősége van egy Ami Pro, mint egy Access makróvírusnak, az alábbi okok miatt azonban mégis kisebb számban jelentek meg:

- az **Ami Pro** nem a legnépszerűbb szövegszerkesztők közé tartozik, kevesebb a potenciális áldozat, mint Word esetében,
- külön van tárolva a dokumentum és a hozzá tartozó makró file – hasonlóan a Word makróvírusok csatolt sablon fertőzési módszeréhez – éppen ezért csak akkor tudnak szaporodni, ha nemcsak dokumentumot, hanem a hozzá tartozó makró file-t is másolják egyik gépről a másikra,
- a fertőzés nagyon könnyen eltávolítható: csak törölni kell az egyes dokumentumokhoz tartozó makró állományokat,
- az **Ami Pro** dokumentumok és makró file-ok egyszerű ascii állományok, így a szövegszerkesztő használata nélkül is könnyen megbizonyosodhatunk, hogy a makró file vírusmakrókat vagy egyéb makrókat tartalmaz.

Az eddig egyetlen makróvírust, a Green Stripe-ot a Mark Ludwig féle vírusíró magazinban publikálták először. Működésére jellemző, hogy automakrós aktivizálódása után a szövegszerkesztő alapértelmezett dokumentumkönyvtárban az összes dokumentumot azonnal megfertőzi, a megnyitott dokumentumokban a „Its"-eket lecseréli „It's"-re.

Mint látható, ezek a makróvírusok működési mechanizmusukban nagyon hasonlítanak bináris társaikra, csak amíg a bináris vírusok „valóságos" gépi környezetben, addig ezek az adott alkalmazás által kreált környezetben élnek és virulnak.

!

Merre fejlődnek a makróvírusok? Sokan azt jósolják, hogy az Excel makróvírusok száma gyorsan fog emelkedni, mások azoknak a vírusoknak az elterjedésére tippelnek, amelyek több alkalmazás alatt (pl. Word-ben és Excel-ben), „keresztben" (cross-platform) képesek fertőzni. Valószínűleg ezutóbbi fog bekövetkezni, mert máris megjelentek azok a makróvírusok – pl. a Strange Day, amelyek az Office 97-ben egységesített irodai alkalmazások makrónyelvének felhasználásával szaporodnak.

Számíthatunk más alkalmazások makrónyelvén kifejlesztett vírusok megjelenésére is, hiszen ha az adott program makrónyelvében találhatóak olyan parancsok, lehetőségek – pl. automakrók, menü-, billentyűzet átdefiniálás –, amelyek a segítségével egy makró vezérlést szerezhet, valamint a program által létrehozott állományokban makró is tárolható, akkor némi fantáziával már kifejleszthető rá makróvírus. A továbbiakban az adott alkalmazás makrónyelvi lehetőségeitől függ, hogy a vírus hogyan tud pusztítani, lopakodni és/vagy mutálni saját kódját.

Végül, egy **„aranyos" sztori:** egy makróvírusnak majdnem sikerült megjárnia az űrt! 1997 októberében azok a számítógépek, amelyekkel e-mail-t küldtek Houston, Moszkva és a Mir űrállomás között, makróvírussal fertőződtek meg, a fertőzési góc kiinduló pontja egy e-mail-hez kapcsolt fertőzött (ál)dokumentum volt. A vírust egy „fertőzött e-mail"-ben – a Mir űrállomásra való elküldése előtt nem sokkal – sikerült felfedezni. Bár a **NASA** szakemberei elégedetten nyilatkoztak a vírus elcsípésével kapcsolatban, elárulták, hogy nem ez lett volna az első vírus, ami megfertőzi a **Mir**-en található laptop gépeket. Kíváncsi lennék, melyik volt az első számítógépes vírus, amely megjárta az űrt?!

5. VÍRUSVÉDELMI MÓDSZEREK

Az operációs rendszer lehetővé teszi a számítógép hatékony és sokoldalú kihasználását: vezérli a programokat, elosztja az erőforrásokat, kényelmessé teszi az ember-gép kapcsolatot, egyszerűbbé teszi a programírást, mindezt a lehető legnagyobb biztonsággal és ellenőrzéssel. Sajnos a PC-k jellemző operációs rendszerei – **DOS, Windows 3.1, Windows for Workgroups 3.11, Windows 95, Windows 98** – közül egyik sem rendelkezik a legutoljára említett tulajdonságokkal (biztonság és ellenőrzés), amelyeknek a hiányát használják ki a vírusok.

Bár ezeknek a közkedvelt operációs rendszereknek valamilyen védelemmel való felruházására a gyártó már tett kísérleteket – antivírus programot mellékelt hozzájuk, az igazi megoldás az lenne, ha a rendszer adminisztrálná, hogy milyen program milyen erőforráshoz férhet hozzá és így az erőforrás hozzáférés ellenőrzésével a rendszer mindig biztosíthatná saját maga épségét. Erre a legjobb példa az alapvetően **PC-re fejlesztett Linux**, melyet olyan **komoly védelemmel láttak el**, hogy nem érdemes rá vírust fejleszteni, mert az nem tudná igazán átvenni a vezérlést a rendszer feje felett. Mivel ilyen védelemmel nem rendelkeznek a fentebb említett rendszerek, ezért nekünk kell valamilyen vírusvédelemmel felruházni ezeket.

A hálózati operációs rendszereknél egy kicsit más a helyzet, mivel azok alapvetően tartalmaznak valamilyen beépített védelmet, mely a felhasználók azonosítására, programok, file-ok és könyvtárak különböző szintű elérésének biztosítására szolgál. Egy megfelelően beállított hálózati rendszer minimális védelmet már ezzel biztosíthat, de ez korántsem kielégítő, ezért itt is szükség van vírusok elleni védelmek alkalmazására.

Egy programról – legyen az bináris vagy makró – nehéz eldönteni, hogy az vírus-e vagy sem, erre különböző "érdekes" műveleteit, viselkedését, illetve egyes utasításait elemezve lehet csak válaszolni. Az előző fejezetek alapján láttuk, hogy a számítógépes vírusok milyen pusztításra képesek, ezért *szükség volt víruskereső programok kifejlesztésére, a vírusok fejlődésének gyorsasága miatt többféle védelmi módszer alkalmazására.* Bár általános algoritmus – ami 100% biztonsággal el tudja dönteni egy kódról, hogy az víruskód vagy sem – nem létezik, a lehető legtöbb védelmi módszer alkalmazása egy antivírus programban növeli annak értékét.

Manapság a legtöbben úgy **védekez**nek a vírusok ellen, hogy rendszeresen vagy alkalomszerűen begyűjtik a **legújabb antivírus programokat és ezeket használják**. Igaz ezek rendszeres használata csökkenti a rendszer

megfertőződésének veszélyét, ez korántsem elég, mert mindig **újabb és újabb vírusok jelennek meg**, melyek közül jó néhány direkt arra van felkészítve, hogy **antivírus programokon egyszerűen "keresztül gázoljon".**

Most lássuk, milyen vírusvédelmi módszereket és technikákat lehet alkalmazni. Néhányat közülük (csali programok, hardveres védelem) már csak érdekességképpen fogok megemlíteni.

5.1. SZIGNATÚRA, SZEKVENCIA VAGY BYTE-MINTA KERESÉS

A **számítógépvírus is program**, ami azt jelenti, hogy **adatokból és utasításokból áll**, melyek többnyire más programokban nem találhatóak meg: ha egy vírusra jellemző adatokat és utasítássorozatokat egy programban megtalálunk, akkor az a program minden valószínűség szerint fertőzött. Az ötlet megvan, csak meg kell valósítani.

A számítógépen minden adat és utasítás hexadecimális számként, byteként vagy byte-sorozatonként van tárolva, így a vírusban az egymást követő utasítások is. Azonban nem a vírus összes egymás utáni utasítására és adatára szeretnénk rákeresni a megtámadható célpontokban, hanem csak egy jellemző utasítássorozat és/vagy adat részre, melyet **szignatúrának, szekvenciának** nevezünk. Ez a szignatúra ugyanúgy **hexadecimális számok**ból, byte-okból áll, éppen ezért ritkán byte-mintának is szokták nevezni.

Pl. az `FC 8B F2 81 C6 0A 00` szignatúra a következő utasításoknak felel meg:

Hexakód	Utasítás
FC	CLD
8BF2	MOV SI,DX
81C60A00	ADD SI,000A

A fenti példa egy 7 byte-os szignatúra, mely a **Vienna vírusra** jellemző, tehát ez alapján már lehetne keresni a vírus által megfertőzhető .com állományokban.

Az antivírus program fejlesztők az ilyen szignatúrákat egy vírus visszafejtése után a kódból veszik, majd a szignatúrát beleillesztik a víruskereső programjuk adatbázisába, azzal a kiegészítéssel, hogy az adott szignatúrát milyen megfertőzhető célpontokban és annak melyik részén keresse a prog-

ram. Erre *a kiegészítésre azért van szükség, hogy egy komplett rendszer ellenőrzésnél a keresési idő minél jobban lecsökkenjen, hiszen pl. boot-szektorban file-vírust nem érdemes keresni.*

Ahhoz, hogy egy víruskereső minél több vírust legyen képes megtalálni szekvencia keresés alapján, rengeteg vírust kell visszafejteni, abból mintát venni és a mintát beilleszteni a programba. Sajnos naponta készülnek újabbnál újabb vírusok és ilyen gyorsan nem lehet mindig aktualizálni a program szignatúragyűjteményét. *Manapság is rengeteg olyan vírus van, amely még nincs felfedezve és ezekből még nem történt mintavétel.*

A **szignatúra** keresésnek létezik egy kicsit **továbbfejlesztett változata a joker karakteres** (? és *) **keresés**. Ezeket a karaktereket akkor használják, ha a vírusban van valami nem állandó rész, vagy pedig egy szignatúrával egy vírus több variánsát is meg akarják találni. Erre a továbbfejlesztésre az első polimorf vírusok megjelenése után volt szükség: ezek a vírusok még nem voltak képesek határtalan módon módosítani önmagukat, csak néhány átírást tudtak magukon elvégezni, így ezzel a módszerrel kiszűrhetőek voltak.

Kérdőjel alkalmazása a mintában azt jelenti, hogy ott tetszőleges félbyte vagy byte állhat, a víruskeresőnek azt nem kell figyelembe vennie. Ha egy vírus több variánsa csak egy-két byte-tal tér el az eredetitől, akkor ilyen módszerrel és egy mintával megtalálható.

Pl. az `FC 5? 8B F2 81 C6 0A 00` szignatúra az **alábbi mintáknak** felelhet meg:

FC 56 8B F2 81 C6 0A 00
FC 57 8B F2 81 C6 0A 00.

Ezeknél a példáknál az `56` egy PUSH SI-t, az `57` egy PUSH DI-t takar. A kérdőjel természetesen a byte első részében is állhat – pl.`?0`, ekkor a byte első felét nem kell figyelembe vennie a keresőnek. Egész byte-os alkalmazásnál:

Pl. az `FC?? 8B F2 81 C6 0A 00` **szignatúra** az alábbi mintákkal egyezhet meg:

FC 90 8B F2 81 C6 0A 00
FC 5E 8B F2 81 C6 0A 00
FC 5F 8B F2 81 C6 0A 00.

Itt a `90` egy NOP utasítást takar, az `5E` egy POP SI-t, az `5F` egy POP DI-t. Több kérdőjel egymás melletti alkalmazása is lehetséges:

Pl. az `FC???? 8B F2 81 C6 0A 00` **szignatúra** az alábbi minta lehet:

FC 90 90 8B F2 81 C6 0A 00
FC 56 90 8B F2 81 C6 0A 00
FC 90 57 8B F2 81 C6 0A 00 .

A * alkalmazása egy szignatúrában azt jelenti, hogy azon byte-ok között, ahol a * helyezkedik el, tetszőleges számú byte szerepelhet.

Pl. az `FC * 8B F2 81 C6 0A 00` **szignatúra** az alábbi mintákkal lehet egyenlő:

FC 90 8B F2 81 C6 0A 00
FC 56 90 8B F2 81 C6 0A 00
FC 90 1E 8B F2 81 C6 0A 00
FC 90 57 5F 8B F2 81 C6 0A 00.

További variációs lehetőség a joker karakterek egymás utáni használata, kombinálása.

A **byte-minta** szerinti keresés **specifikus védelem, ismert file-, boot/partíciós- és makróvírusok felismerésére szolgál.** Makróvírusok esetében, ha az adott alkalmazás tokenizáltan tárolja a makrókat, akkor a *token byte-jait és az esetleges makróneveket lehet keresni a megfertőzhető állományokban.* Ezt a módszert minden közismert antivírus program – *pl. Scan, F-prot, Chkvir, Virware, Uve, Virkill* – használja. Segítségükkel az általuk ismert vírusokat könnyen felismerhetjük, a programok nagy része a felismert vírusok többségét irtani is képes. Újabb vírusokat egy régebbi verziójú program már nem detektál, éppen ezért mindig be kell szerezni a legfrissebb programverziót, ami nem nehéz, mert elég sűrűn vannak aktualizálva.

Érdemes többféle víruskeresőt is használnunk, bár az egyre növekvő winchester kapacitások és vírusszám miatt az ellenőrzés sokáig eltarthat. **Használatuk rendszerint nagyon egyszerű**, nem igényelnek túl nagy rendszerismeretet. A módszer további hátrányai:

- **False positive**: rosszul megválasztott szekvencia esetén vakriadózhat a keresőprogram – ha a minta olyan utasítássorozat (rutin) byte-jait tartalmazza, ami más, nem vírusprogramban is előfordulhat, akkor vírussal nem fertőzött célpontot is fertőzöttnek titulálhat.

- False **negative**: a szekvencia kiválasztásától függetlenül a védelem olyan megfertőzhető objektumot tart tisztának, amely valójában már nem az. Ez akkor fordulhat elő, ha ez a védelem még nem ismeri az adott vírust.
- **Régebbi víruskeresőkben a szignatúrák nem voltak titkosítva** vagy kódolva, így az antivírus programok könnyen jelezhették egymást fertőzöttnek, mivel általában azonos szignatúra szerint szoktak keresni. Éppen ezért a későbbiekben a vírusokból 2 különböző helyről választottak szignatúrát, valamint a program a szekvenciákat kódolva tárolta.
- **Ez a módszer nem képes a polimorf vírusok megtalálására**, mert az ilyen vírusokból állandó alakváltoztató képességük miatt nem vehető minta, ami alapján lehetne keresni azokat. Ezen vírusok detektálásához speciálisan megírt algoritmusok kellenek.

5.2. ISMERT VÍRUSOKAT ELTÁVOLÍTÓ VÉDELEM

Ez a védelem az előzőtől annyiban különbözik, hogy **az általa ismert vírusokat nemcsak megkeresi, hanem el is távolítja.** Ennek a védelemnek "tudnia" kell, hogy melyik vírus milyen módosításokat végez megtámadható objektumokon, különben nem tudná visszaállítani a fertőzött rendszer fertőzés előtti állapotát. Bár nem a módszer hátránya, de vannak olyan esetek, amikor egy vírus eltávolítása szinte lehetetlen, mert az olyan súlyosan roncsolta a rendszer egyes területeit, hogy csak backup visszatöltés vagy egy komplett újrainstallálás segít.

A módszer **legkényesebb** része a vírusok pontos azonosítása, mert lehet, hogy egy vírus és variánsa között van azonosság, de a kiirtásuk a különbözőségüknek "köszönhetően" már másképp történhet. Éppen ezért a pontos felismerés érdekében jóval hosszabb vagy dupla szignatúrát (egy vírusból 2 különböző minta) kell választani az egyes vírusokból.

A pontos azonosításra az ellenőrzött rendszer érdekében is szükség van: ha ez a védelem egy nem fertőzött objektumból el akarná távolítani a "magtalált" vírust, akkor azt tönkretenné.

Ezt a módszert szintén minden közismert antivírus program használja.

5.3. HEURISZTIKUS KERESÉS

Ezen speciális keresési eljárás kifejlesztésének az oka egyrészt a lopakodó és a polimorf vírusok megjelenése, másrészt fel nem fedezett vírusok felismerésének a célja volt.

Egy heurisztikus kereső nem szignatúrákat keres, hanem viselkedése és műveletei, illetve azok száma alapján próbálja eldönteni a lehetséges célpontról annak fertőzöttségét. Egy ilyen védelemmel ellátott program a bináris programokhoz egy beépített processzoremulátorral rendelkezik, amellyel az ellenőrizni kívánt program utasításainak a végrehajtását szimulálja és analizálja. *Mivel nem az igazi processzor hajtja végre az utasításokat, ezáltal a programban esetlegesen bennlévő vírus nem tud elindulni, így sok vírustrükk kivédhető.*

Az általános heurisztikus kereső a szimulált végrehajtás folyamán különböző vírusra utaló műveletekre figyel: pl. végrehajtható állományokba való beleírás, nem publikált interruptok használata, speciális módon való rezidenssé válás, debug ellenes trükk használata.

A specifikus heurisztikus kereső a szimulált végrehajtás folyamán az azonos csoportba tartozó – **pl. MOV reg.,reg. vagy MOV reg.,cím** – utasításokat darabra megszámolja, ha ez a szám valamilyen vírusra jellemző, akkor figyelmeztet.

A **makróvírusok heurisztikus keresésénél** nincs processzoremulátor használat: a heurisztika a megfelelő állományokban jellemző makróvírus utasításokat keres – pl. *makró másolás, létrehozás, feltöltés, törlés, bizonyos makrók keresése, makrók megszakításának tiltása, sablonná konvertálás, normál sablon lementés megerősítés kikapcsolás, globális sablonban található makrók számának és az egyes makrók nevének lekérdezése, automakrók használatának újra engedélyezése, állomány jelszóval való levédése, makró indítása megadott időpontban, paraméter begyűjtés Win.ini állományból, menüpont törlés, szekvenciális állomány közvetlen írása és olvasása, file attribútumának megváltoztatása, valamilyen aktivizálódási mód utasításai, könyvtár vagy file törlés.*

Persze itt is előfordulhatnak **vakriadók**: pl. nemcsak vírusok használnak nem publikált interruptokat vagy debug ellenes trükköt, hanem a másolásvédelemmel ellátott programok többsége is, valamint nagyon sok olyan hasznos makró van, ami a globális makróterületre telepszik be stb.

A mai antivírus programok nagy részében már található **heurisztikus kereső**, az mindig a szekvencia keresés illetve vírusirtás után fogja analizálni az ellenőrizendő programot.

5.4. ELLENŐRZŐ ÖSSZEGES VÉDELEM

Az ellenőrző összeges védelem a lemezen tárolt programok, adatállományok, dokumentumok, valamint a boot-szektor és partíciós tábla ellenőrző összegeivel operál.

Mint tudjuk, egy számítógépen minden program és adat számként van tárolva. Ez a tulajdonság vírusvédelmileg kihasználható: ha a megfertőzhető **célpontok byte**-jait egy bonyolult matematikai algoritmusba – **CRC**-be (**C**yclic **R**edundancy **C**heck) – helyettesítjük, majd a kapott értéket eltároljuk, akkor lehetőségünk nyílik arra, hogy egy későbbi időpontban újraszámolt értékekkel összehasonlítva a régieket, megmondjuk, hogy melyik célpont fertőződhetett meg az eltelt idő alatt. Ugyanis jó algoritmus esetén kicsi a valószínűsége, hogy ugyanazt az értéket kapjuk meg újraszámolva az adott objektumra, ha azon változás történt, ahhoz pedig, hogy egy vírus kiszámolja a **CRC**-t és módosítsa a lementett értékeket, szüksége lenne a matematikai algoritmusra, ami védelmi rendszerként eltérő lehet. Elméleti szinten persze elképzelhető olyan vírus, amely egy-két ellenőrző összeges védelem **CRC** algoritmusát ismeri és használja.

Ha az újraszámolt **CRC** nem egyezik meg, akkor sem biztos, hogy vírusról van szó: vagy megsérült az eredeti **CRC** érték vagy pedig a **CRC** ellenőrzött objektum módosította önmagát...

Mint minden módszernek ennek is vannak hátrányai:

- A FAT és **CEB** vírusok nem közvetlenül fertőzik a programokat, ezért egy fertőzött program újraszámolt **CRC**-je megegyezik a fertőzetlen állapotban számolt **CRC**-jével. Elviekben ez megoldható, hiszen a vírusvédelem minden program kezdő szektorát ellenőrizhetné, ha túl sok azonos helyen kezdődne, akkor szinte biztos, hogy **FAT** vírusról van szó, vagy érdemes lenne figyelembe venni, hogy hány azonos nevű, de más kiterjesztésű program létezik. Ez a fajta ellenőrzés azonban nem az ellenőrző összeges védelem körébe tartozik.

- Léteznek olyan programok, amelyek kifejezetten érzékenyek arra, ha megváltoztatjuk őket – pl. ha az ellenőrzött program végén van tárolva a **CRC** érték: jobb esetben nem hajlandó működni, roszszabb esetben rombolhat a rendszerben.

- A **CRC** értékeket tárolni kell, a lehető leggyakrabban erre egy külön állomány szolgál. Ha egy vírus ezt az állományt képes megtalálni és törölni, akkor az ellenőrző összeges védelem legközelebbi

futtatásakor úgy érzékeli, mintha még nem készített volna **CRC** adatállományt, éppen ezért nem is tudja kiszűrni az esetleges változásokat. Ez ellen lehet védekezni: **CRC** értékeket tároló adatbázist floppyra kell kimenteni, ami lassú, vagy az adatbázisból több példányt tárolni különböző könyvtárakban a winchesteren.

A lopakodó vírusok jellemzőjükből fakadóan ha aktívak a memóriában, akkor mindent az eredeti állapotában mutatnak, vagyis hiába az ellenőrzés, ha a fertőzött rész **CRC**-je megegyezik a régi eltárolt értékkel, éppen ezért ennek a védelemnek ügyelnie kell arra, hogy lehetőleg semmiféle vírus ne csücsüljön benn a memóriában vagy anti-lopakodó technikát kell használnia ahhoz, hogy egy esetlegesen aktív lopakodó vírus át ne tudja verni.

Nem minden antivírus program alkalmaz ilyen védelmet és rengeteg csak ezzel a védelmi módszerrel működő egyéb program létezik.

5.5. ÁLTALÁNOS RENDSZERFELÜGYELŐ VÉDELEM

Az általános rendszerfelügyelő – angolul monitor – programok, ha megfelelően vannak egy rendszerbe illesztve, akkor minden egyes rendszerindításkor rezidenssé válnak és egészen a rendszer kikapcsolásáig vagy memóriából való **uninstallálásukig** aktívak: ezek a vírusok elleni harc első frontvonalai, mivel megelőzhetik a vírusok rendszerbe való bejutását.

Az ilyen típusú program:

- ellenőrzi a futtatott programok működését,
- figyeli a rendszerint vírusok által hívott interruptokat,
- illegális memória művelet,
- illegális program művelet
- és lemezkezelésnél figyelmezteti a felhasználót és engedélyt kér az általa illegálisnak tartott művelet folytatására. Rendszerint **szignatúra** és **heurisztikus keresővel vannak kombinálva,** ezáltal képesek megakadályozni fertőzött programok elindítását, rendszerbe való bemásolását és fertőzött lemezek használatát.

Makróvírusok elleni harchoz is fejlesztett ilyen típusú programokat: az elsők még csak makró szinten védték az egyes alkalmazásokat, de ez nem adott elegendő védelmet, *folytatásnak már mélyebb szintre beülő makróvírus védelmi rendszereket fejlesztettek.* Ezek ugyanúgy működnek, mint bináris társaik és *hasonlóan kombinálva vannak szekvencia és heurisztikus keresőkkel.*

Nem minden esetben lehet tisztázni, hogy melyek az illegális műveletek és melyek nem: vannak olyan gyengébben sikeredett monitor programok, amelyek már egy program **rezidenssé válásakor** figyelmeztetnek, vagy **floppy formázásakor „kiabálnak" a boot-szektor változása miatt,** vagy **ha sablon állományból való makró másoláskor abban ismeretlen vírust vélnek felfedezni.**

Léteznek vírusok, melyek **monitor programot érzékelve el sem indulnak vagy legalábbis vírusra jellemző dolgot nem csinálnak, de vannak olyanok is, amelyek a legjobb felügyelő program mellett is képesek aktivizálódásra.**

5.6. EGYÉB SZOFTVERES VÉDELMI MÓDSZEREK

5.6.1. Védőoltás a megfertőzhető célpontoknak

A reprodukciós rutin részletezésénél említettem, hogy a vírusok nagy része a megfertőzött objektumot megjelöli, hogy többszörös fertőzést elkerülje. Ebből **rögtön adódik egy roppant egyszerű,** ám kevésbé tökéletes megoldás: **ha a vírus helyett mi jelöljük meg a célpontot, mintha az fertőzött lenne, akkor a vírus nem fogja megfertőzni. Vagyis az objektumot immunissá tettük, beoltottuk a vírus ellen.**

A megjelölés vírustípusonként különbözik, file-vírusoknál alkalmazható immunizálási módszerek:

- program létrehozási idejének (dátum és/vagy időpont) átállítása, ha a vírus pl. a másodperc adott értéke alapján tudja eldönteni a fertőzöttséget,
- hossz megváltoztatása, ha a vírus a program hossza alapján dönt a fertőzésről, pl. a hossz osztható-e 16-al,
- .exe fejlécben található ellenőrző összeg átírása,
- azonosító hozzáírása programhoz, pl. a Jerusalem vírus a megfertőzött filevégekhez az "MsDOS" azonosítót fűzi,
- környezeti változó beállítása, pl. a Syslock vírus azokon a gépeken nem szaporodik, ahol be van állítva egy SYSLOCK környezeti változó (Set SYSLOCK=@).

Néhány boot-vírus esetén alkalmazható módszer, ha a boot szektor néhány byte-ját átírjuk – a boot-program ettől függetlenül működőképes marad, de a vírus fertőzöttnek fogja hinni a lemezt.

Boot- és filevírusok elleni védekezés egy régebben elég gyakran alkalmazott módja volt az immunrutin hozzákapcsolás: program esetén ez azt jelentette, hogy az appendelő vírusokhoz hasonlóan olyan rutint kapcsoltunk hozzá – pl. egy antivírus programmal, megfelelő paraméterrel való indítása után – ami minden egyes program futtatáskor **CRC** értékszámolással ellenőrizte a "gazda" program épségét, fertőzöttség esetén vagy "csak" figyelmeztette a felhasználót vagy el is távolította a vírust. Természetesen a bootszektor illetve a partíciós táblára is lecserélhető immunizáló rutinra. Ennek a legfejlettebb immunizációs technikának főellensége a felülíró típusú vírus, valamint a másolásvédett programok, amelyek kifejezetten érzékenyek programkódjuk változására.

A makróvírusok felépítésükből adódóan másképp ellenőrzik a fertőzöttséget, így az esetleges immunizálás is másképp történhet:

- üres makró(k) beszúrása megfelelő névvel: a makróvírusok többsége név szerint ellenőrzi, hogy milyen makró(ka)t tartalmaz a megtámadott célpont, van közöttük olyan, amely csak 1 makróját (pl. FileSave, Autoopen!!), van olyan, amely az összes makróját keresi. Azok a makróvírusok, amelyek csak 1-2 jellemző vírusmakrót (pl. AutoOpen, AutoExec) keresnek a célpontban, könnyen immunizálhatják az állományokat egymás ellen!
- megfelelő dokumentum változók beállítása, ha a makróvírus adott értékekre figyel.
- "üres" file-ok létrehozása, ha a makróvírus fertőzés folyamán megadott könyvtárban valamilyen file meglétét ellenőrzi, pl. a **DOS** könyvtárban az Ega5.cpi-t (amit a vírus hoz létre, ha az adott gépen már fertőzött),
- Win.ini bejegyzés bővítés (utolsó kettő csak komplett rendszer immunizálásra használható!).

Az immunizációs technika kevésbé elterjedésének egyik oka, hogy egy-egy vírus egészen másképpen jelölheti a fertőzést, még egy módszeren belül is – pl. X vírus csak akkor nem fertőzi meg a file-t, ha a létrehozási dátumban a másodperc 31-re végződik, míg Y vírus csak a 32-re végződőket nem bántja és lehet, hogy a vírus variánsa csak ennyiben tér el, másrészt számtalan fertőzöttséget nem ellenőrző vírus létezik, harmadrészt egy ellenőrző összeges program használata utáni bárminemű immunizálás (módosítás) fertőzésnek tűnhetne.

5.6.2. Írásvédelem használat

5.6.2.1. Szoftveres írásvédelem

A program által biztosított írásvédelem is létezik: read-only attribútum használata file-okra **DOS** alatt, illetve írásvédelem program által történő biztosítása tetszőleges meghajtókra. Ilyen szoftveres védelmi program pl. a HdSafe, a HdProtect stb. A read-only attribútumot a vírusok könnyen kijátszák: a megtámadott programról ezt az attribútumot leszedi, megfertőzi, majd újra visszaállítja az eredeti attribútumot. Program által biztosított írásvédelem esetén az egyszerűbb programok nem tudnak írni a lemezre, de mint minden program, ez is kijátszható, ugyanakkor a gép kikapcsolásakor a látszólagos írásvédelem is elvész.

5.6.2.2. Hardveres írásvédelem

Winchesterre egyedül a régebben itt-ott felbukkant vírusvédelmi kártyák voltak csak képesek fizikai írásvédelmet biztosítani, ma már nem jellemző ezek használata.

Többen is reklamáltak nálam az 1. kiadásban írt floppy hardveres írásvédelmének átverési lehetőségéről írtak miatt. Szeretnék idézni egy levélből: "Floppy esetében a leragasztás tökéletes védelmet tud nyújtani: a floppy kapcsolási rajzán nyomon követhető, hogy a fotódióda jele és az íráskérelem jele egy ÉS kapun keresztül fut az íróáramkörbe, így ha írásvédett a lemez, akkor egyszerűen nincs íróáram, vagyis a felmágnesezéshez szükséges áram sincs meg. Ez már a legelső PC-k floppy-ján is így volt megoldva és ez az egyetlen dolog, amihez gyakorlatilag azóta nem nyúltak."

Jobban utána néztem a dolgoknak és kiderült, hogy a régebbi 160/180/320/360KB-os 5.25" duplamagas floppymeghajtókban egy nagy, egykarú emelő végén levő ék kattant be, ha nem volt leragasztva az írásvédő kivágás, ha gyengén volt leragasztva, akkor az emelő erős rugója miatt letépte az írásvédő ragacsot és lehetett rá írni. Egyesek szerint ebből, valamint a 3.5" meghajtóknál az írás védelmi pöcök megfordított használati sorrendje (az 5.25"-höz képest akkor van védve, ha át lehet látni a lyukon) miatt és az esetleg bepiszkolódott, beragadt érzékelőjű floppy-k miatt keletkezhettek az írásvédett floppyra író vírusokról szóló "rémtörténetek".

Egy valamit még mindig nem értek: a Novell hogyan tudta azt megoldani, hogy a Netware 2.x változatainak telepítésekor visszaír az írásvédett lemezre?!

5.6.3. Csali programok

Régebbi antivírus programok vírusellenőrzéskor csali programokat helyeztek el az ellenőrzött rendszerben, melyeket lefuttattak, majd a programok futtatás utáni állapotát összehasonlították az eredetivel: ha változás történt, akkor annak oka csakis file-vírus lehetett. Napjainkban ilyen csali programokat file-vírusokkal való kísérletezgetés közben használnak.

Ilyen csali programok egyszerű futtatása azonban nem mindig elegendő, mert nem minden file-vírus támadja meg azonnal az elindított vagy létrehozott programokat, nem is beszélve a lopakodó vírusokról, amelyek megpróbálnak mindent az eredetiben mutatni.

5.7. HARDVERES VÉDELEM

Valamikor a 90-es évek elején egyes antivírus cégek vírusvédelmi kártyák fejlesztésében látták a jövőt: a védelem egy olyan kártyán keresztül valósult meg, mint amilyen a számítógépben található más kártyák – pl. monitorvezérlő kártya, a kártya ROM-jában vagy RAM-jában lévő antivírus program már akkor aktivizálódhatott, amikor a boot-folyamat elindult, tehát blokkolni tudta a boot-vírusokat. Legjobban talán a monitor programokhoz hasonlítható, de azzal szemben volt egy óriási előnyük: ha benn volt a kártya a gépben, akkor biztos hogy végrehajtódott a kártya programja és az nem volt változtatható.

Természetesen ezeket a kártyákat is át lehetett verni – a kártya programját ott kellett meghívni, ahol a különböző szintű védelmi ellenőrzéseket kikapcsolta – mégsem erről voltak híresek.

Hogy miért tűntek el? Nem tudni, talán azért, mert a "piac így döntött", másrészt a kártyák vírus ismeretének bővítése nem teljesen jól volt megoldva, harmadrészt különböző szoftverek futását a legtöbb kártya akadályozta.

A vírusvédelmi kártyák után nem sokkal egyre több olyan alaplap jelent meg, melyek lehetővé tették a winchester boot-szektorának és partíciós táblájának védelmét: módosítási kísérlet esetén figyelmezteti a felhasználót és annak döntése alapján engedélyezi a módosítást. Ez a hardveres vírusvédelem mind a mai napig jellemző és hasznos, bár használata Windows 95 telepítés esetén kevésbé ajánlatos...

A fentiekben felsorolt védelmi módszerek önmagukban nem adnak maximális védelmet, a lehető legnagyobb biztonságot kombinált megoldásokkal érhetjük el a vírusokkal szemben.

Mégis az lenne a legjobb, ha minden lehetséges vírusvédelmi megoldás már alapvetően bele lenne építve az operációs rendszerekbe. Ehhez azonban vagy a PC-ken használt rendszereknek a filozófiáját kellene megváltoztatni vagy egy teljesen új operációs rendszert kellene kifejleszteni, amire mindenki áttérne. Új rendszerre való áttérésnek a legkisebb az esélye, marad tehát a különböző antivírus programok továbbfejlesztése, azok alkalmazása valamint a következő fejezetben tárgyalt vírusvédelmi szabályok, tanácsok betartása.

6. VÍRUSVÉDELMI TANÁCSOK

Abszolút biztonságos számítógépes rendszer nem létezik: zárt rendszer esetén is elég egy gonosz szándékú munkatárs, aki a rendelkezésre álló fejlesztői eszközök felhasználásával létrehoz egy vírust, egy férget vagy egy trójai programot. De ne abból induljunk ki, hogy munkatársaink gonoszak, a legnagyobb problémát a

Vírusfertőzések kiváltó okai

- felhasználók hiányos ismeretei (nem értik és nem használják az alapvető biztonsági eszközöket, fertőzött szoftvereket másolhatnak a rendszerbe és nem képesek észrevenni a vírusra jellemző számítógép viselkedéseket),
- hiányzó és/vagy nem megfelelő biztonsági ellenőrzések (egy PC-hez általánosan hiányoznak a megfelelő szoftveres és hardveres biztonsági megoldások),
- hiábavalóan létező biztonsági ellenőrzések (könnyen kitalálható jelszavak, hozzáférési jog technika kihasználatlansága, adott felhasználónak felesleges és veszélyes jogok biztosítása),
- jogosulatlan gép felhasználások (rendszerbe való betörés, felhasználók jog túllépése),
- hálózatok veszélyeztetettsége (csak alapvető biztonsági rendszer használata, anonymus felhasználónak jogok biztosítása) jelentik.

Ezekből adódnak a komplett rendszer vírusfertőzések, rendszer összeomlások. A vírustámadások, vírusfertőzések veszélye, ezáltal a rendszer veszélyeztetettsége és az esetlegesen keletkező károk mértéke jelentősen csökkenthető bizonyos munkahelyi és egyéb biztonságtechnikai előírások, óvintézkedések betartásával. Ezek az előírások és tanácsok 4 fázisra oszthatóak: !

- megelőzés,
- detektálás,
- azonosítás,
- helyreállítás.

6.1. VÍRUSFERTŐZÉS MEGELŐZÉS

Nagyobb cégeknél és szervezeteknél a megelőzés első lépéseként érdemes létrehozni egy szakértői csoportot, akik megteszik a megfelelő vírusvédelmi lépéseket és rutinosak vírusos problémák felmerülésekor. Ha ilyen típusú emberek nem találhatóak a vállalaton belül, akkor érdemes egy adat- és vírusvédelemre szakosodott céggel szerződést kötni a rendszeres vírusellenőrzés érdekében.

6.1.1. Víruskapuk és bezárásuk

Egy számítógépes rendszerbe egy vírus vagy maga a vírusíró által – aki ellen tehetetlenek vagyunk, vagy pedig olyan ember "segítségével" jut be, aki mit sem sejt a vírust hordozó lemezről, programról, vagy "makrós" állományról.

Kezdjük a legelején: egy PC bekapcsolásakor az egyes eszközök lekérdezése, tesztje és felprogramozása vírusvédelmi szempontból lényegtelen, annál fontosabb viszont az operációs rendszer betöltésének kezdete. Ha nincs floppy meghajtó vagy winchester, akkor boot Eprom-os géppel van dolgunk. Itt van az első víruskapu: vajon tiszta lehet az Eprom? Magyarországon már előfordult olyan eset, hogy a boot Eprom program beégető cég vírusfertőzést kapott és a vírussal fertőzött programot égette be!

1

Ha nem boot Eprom-os a gép, akkor a Bios beállításoktól függően vagy floppyról, vagy winchesterről, vagy – az újabb gépeken – CD-ről próbálja betölteni az operációs rendszert. Floppy esetén a boot szektor helyén lévő programot, winchester esetén a partíciós tábla programját tölti be a gép és átadja a ezeknek a programoknak a vezérlést. Második víruskapu: ha nem egy eredeti boot szektort vagy partíciós tábla programot tölt be a gép, hanem egy vírust, akkor az máris vezérlést kapott! Winchesteren az eredeti partíciós tábla program választja ki az aktív partíciót, majd onnan betölti a boot szektort. Ez is egy víruskapu, de ez még a másodikhoz tartozik.

2

Ennél a második kapunál adódhatnak az első igazi problémák: ha a rendszer alapbeállítás szerint a floppy-n próbálkozik először az operációs rendszer betöltésével, akkor egy – a meghajtóban felejtett – akár nem rendszerlemezről is egy boot-vírus könnyen aktivizálódhat: az eredeti boot program helyett a vírus töltődik be és kap vezérlést, végrehajtódnak a vírusba beleírt utasítások, majd ezután betölti az eredeti boot programot és átadja a vezérlést, a boot program pedig hibaüzenettel figyelmeztet az operációs rendszer hiányára. Ha a felhasználó észrevéve hibát, másodszorra már a winchesterről indítja a rendszert, akkor a már aktív vírus rámászhat a win-

chesterre is. Megjelentek azonban olyan vírusok is (pl. Michelangelo), amelyek rögtön megfertőzik a winchestert, így nincs szükség a második rendszerindítási próbálkozásra!

Boot-vírus csak így képes megfertőzni a merevlemezt. Ez ellen a hiba lehetőség ellen nagyon könnyen lehet védekezni: !

- A mai gépek Bios-ában beállítható a rendszer boot-olási sorrend: ezt kell átállítani úgy, hogy a boot-olási kísérlet először az első (vagy a második) merevlemezről majd a floppyról történjen. A legújabb Bios-ok már azt a lehetőséget is támogatják, hogy kizárólag merevlemezről történjen a boot-olás.
- Ha nincs B: meghajtónk, akkor a floppyvezérlő kábelt kell átkötnünk A:-ról B:-re. Hátránya, hogy a kábel eredeti állapotba való visszakötéséig floppy-ról nem lehet rendszert indítani.
- A legdrasztikusabb módszer, ha a floppy meghajtókat kiszereljük a gépekből, ám a szándékos fertőzéstől ez sem véd meg: valaki új vírust fejleszthet a rendszerben, vagy pedig egy egyszerű Copy con: parancs után sorba begépeli egy kisebb vírus hexadecimális kódjait, majd lefuttatja a létrehozott programot. Az általam ismert legkisebb vírus 44 byte-os, 44 db számot megjegyezni nem nehéz, ráadásul a felhasználóra nem is terelődhet a gyanú, hiszen nincs floppymeghajtója. Floppy kiszerelést hálózatoknál "érdemes" megcsinálni.

Menjünk tovább a boot-olással: MS-Dos esetén az Io.sys és az Msdos.sys, PC-Dos esetén az Ibmbio.com és az Ibmdos.com, Windows 95 esetén az Io.sys után a Config.sys-beli meghajtó programok, majd a Command.com után az Autoexec.bat-ból meghívott programok kapnak vezérlést. Harmadik víruskapu: a rendszerprogramok, a Config.sys-ből és az Autoexec.bat-ból betöltött programok nem fertőzöttek? Esetleg nem olyan programot indítunk el minden egyes rendszerindításkor, ami csak adott feltételre (pl. dátumra) vár? 3

A boot-olás befejeződött, de indítva vannak egyes programok. Negyedik víruskapu: programok futtatáskor biztosak lehetünk azok vírusmentességéről? Másolásvédett terméket használunk-e? Ha igen, akkor hivatalosan megvásároltuk, vagy egy másolásvédelemtől megszabadított verzióval dolgozunk? Nem lehet, hogy a programon belül még egy másolásvédelmi funkció van?! Biztos forrásból származnak programjaink? 4

Rendszeresen jelennek meg trójai programok, gyakran a kalóz szoftver is egy módosított program, ami vírust vagy trójai funkciót tartalmazhat.

Az "igazi" trójai programok kiszűrésére néhány ötlet:

- nincs dokumentáció a programhoz, vagy ha van, akkor "jellemző" szövegeket tartalmaz pl. death, megafuck stb.,
- nincs cég vagy copyright információ kijelzés a programban vagy a mellékelt dokumentációban, vagy ha van, akkor egészen furcsa és "feltűnő" nevű a cég pl. Hungarian Trojan Agency, vagy nem létező a cég,
- ellenőrizni kell, hogy tényleg a cég terméke a kapott program (ezt különböző számítástechnikai újságok figyelésével könnyen megtehetjük),
- összehasonlítás az előző verzióval méret és létrehozási dátum szerint,
- bináris állományok átlapozása, szöveg után kutatva (feltéve, ha nem kódolva tartalmazza azokat): egy elég furcsa szöveg árulkodó lehet,
- esetleg létező forrásfile átnézése, ellenőrzése.

5

Munkánk során más felhasználókkal – gépekkel floppyn, helyi hálózaton, vagy Internet-en keresztül kommunikálunk. Ötödik víruskapu: Vírusmentes bináris és makrókat tartalmazható állományokat csere-berélünk, küldünk levélben, töltünk le? Milyen biztonsági előírások vannak érvényben a partner cégnél, akitől dokumentumok érkeznek e-mail-en?

Mivel felgyorsult világunkban a hatalmas adatforgalom jellemző, ezért:

- minden letöltött vagy elküldeni szándékozott állományt vizsgáltassunk át a lehető legfrissebb antivírus programokkal,
- lehetőleg minimalizálni kell a megfertőzhető állományok cseréjét,
- értékes lemezeinket soha ne kölcsönözzük, csak azok másolatait,
- idegen gépen csak írásvédett floppyt használjunk,
- idegen gépnél használt nem írásvédett floppyt vírusellenőrizzük,
- helyi rendszerünket úgy alakítsuk ki, hogy az ne programok "raktára" legyen, vagy ha ilyen céllal fog működni, akkor magas fokú biztonsági előírások legyenek érvényben,
- legyen egy közepes kapacitású gép, amelyen a kívülről érkezett anyagok – vírusellenőrzés után – kipróbálhatóak. Ha kísérletezgetéseink során sem tapasztalható trójai vagy vírus aktivitás, korántsem lehetünk biztosak a fertőzöttség mentességben: előfordulhat, hogy a vírus adott gépen nem képes szaporodni vagy a trójai sok futtatás után aktivizálja magát.

Bár a legnagyobb biztonságot jelentő megoldások korlátozzák a rendszer működését, ezzel csökkenthető a kívülről jövő fertőzés valószínűsége.

6.1.2. Vírusvédelmi oktatás

Az egyes víruskapuk bezárása és nagyobb védelmi szint elérése nem képzelhető el a felhasználók vírusvédelmi oktatása nélkül.

A felhasználóknak el kell magyarázni:

- a trójai programok, férgek, vírusok működési elvét, rombolási lehetőségeit,
- az ellenük való védekezés lehetséges módozatait,
- az általános biztonsági eljárások és programok használatát,
- mentési és másolási eljárások és programok használatát,
- hogyan lehet észrevenni a rendszerhez nem jogosult személyek tényked ését,
- a különböző technikai ellenőrzések használatát saját adatai, programjai érdekében,
- hogyan figyelje és vegye észre a rendszer és a szoftverek abnormális viselkedését, illetve ilyen esetekben mit kell és lehet cselekedni, honnan és kitől lehet további információkat szerezni.

Σ

A felhasználók oktatása pénz- és időigényes, a rendszer ezáltal viszont védettebb lesz, a védelem pedig olcsóbb, mintha hétről-hétre az adatok helyreállításával, a rendszer rendbetételével kell bajlódni. Az olyan cég vagy szervezet, ahol az oktatás alapvető, bármiféle probléma esetén időt és pénzt takaríthat meg.

6.1.3. Mentés, másolás, biztonsági példány készítés

A mentést legelőször egy rendszerlemez készítéssel kell kezdeni. Ezt elegendő egyetlen egyszer megtenni, attól fogva bármi probléma esetén a rendelkezésünkre áll. Ennek a lemeznek a rendszer állományokon kívül a legfontosabb – rendszerhez tartozó és egyéb – segédprogramokat valamint antivírus programokat kell tartalmaznia. Lehet, hogy az összes szükséges program egy floppyra el sem fér, mégis érdemes több floppyn tárolni ezeket. Azzal még jobban bebiztosíthatjuk magunkat, ha ezekből a lemezből kettőt készítünk. Mondanom sem kell, ezek a lemezek legyenek vírusmentesek és írásvédettek.

1

A további mentéseknek a rendszer egyéb elemei felé kell irányulniuk: gyári installációs lemezek és CD-k, adatok, egyéb programok. Nagyobb mennyiségű adat és program lementéskor hasznos dolog CD-t használni: tároló kapacitása nagy, huzamosabb ideig képes adatokat, programokat tárol-

2

ni. Egyetlen hátránya van: ha véletlenül fertőzötten mentettünk ki néhány dolgunkat CD-re, akkor a vírust arról a CD-ről többé nem lehet törölni. Ilyenkor a CD-t vírusmentesítve kell másolni.

Gyári installációs lemezeinkről, CD-inkről biztonsági másolatot kell készíteni és a floppy-kat az eredeti példányokkal együtt írásvédetté kell tenni. Az írásvédelemre azért van szükség, mert egy ismeretlen vírussal fertőzött gépre történő installáláskor a gyári lemezek is megfertőződhetnek, ezután másik gépen való használatuk egyértelműen annak megfertőzését fogja jelenteni.

A nem gyári programokról is legalább egy, de lehetőleg két másolatot kell tárolni. Programokat azért sem érdemes minden egyes backup-oláskor elmenteni, mivel azok nem változnak, ellentétben például egy adatbázissal. Ügyelni kell arra, hogy a mentett programok ne legyenek fertőzöttek: vírus takarítás után, ha visszatöltjük a fertőzött másolatokat, akkor az olyan, mintha ki sem takarítottuk volna a vírust a gépből.

3 Létfontosságú adatbázisainkat, táblázatainkat, dokumentumainkat, (globális) sablonjainkat sűrűn kell menteni, így szinte mindig aktuális állapotot tudunk visszaállítani vírusfertőzés elhárítása után. Pénzt spórolhatunk meg, ha régebben elmentett adatainkat töröljük, pl. ha hetente 3-szor van mentés, akkor az 1 hónappal ezelőtti adatokat már felül lehet írni, hiszen azok már elvesztették aktualitásukat.

Ha mentés folyamán bármilyen tömörítőt használunk, akkor fennáll egy esetleges rejtett fertőzés lehetősége is. Ez ellen a legjobb víruskereső programok már tudnak védekezni: tömörített állományokat is képesek átvizsgálni.

6.1.4. Hálózati hozzáférés beállítás

A hálózati operációs rendszerek rendelkeznek egy beépített védelemmel, amely a felhasználók azonosítására, programok, file-ok és könyvtárak különböző szintű elérésének biztosítására szolgál. Egy hálózati rendszer megfelelő beállításokkal akadályozhatja a vírusok szaporodását, így kisebb kárt is tudnak okozni, mint egy önálló PC-n.

1 A felhasználó azonosítását a lehető legegyszerűbben bejelentkezési név és jelszó használatával lehet megoldani. A jelszavak időközönkénti kötelező cserélése a rendszer biztonságát növeli, különösen akkor, ha kizárólag új jelszót lehet használni. Fontos, hogy minden egyes emberhez megfelelően legyenek hozzárendelve az elérési és hozzáférési jogok. Ha valaki távozik a cégtől vagy szervezettől, akkor a hálózatban rá vonatkozó jogokat törölni kell. Sosem lehet tudni, hogy távozása előtt kinek árulta el jelszavát...

A rendszergazda kivételével ne legyen más olyan ember, aki mindenhez hozzáférhet, minél kevesebben használhatják ezt a jogkört, annál jobban csökkenthető a vírustámadás veszélye. Lehetőleg csak ő telepítsen programot a hálózatra és mindig ellenőrizze a hálózat figyelmeztetéseit, ha valaki jogtalanul akar erőforrásokhoz hozzáférni. 2

A guest vagy az anonymus felhasználók részére korlátozni kell azoknak a parancsoknak a körét, amit használhatnak. Nem szabad ezeknek a felhasználóknak bizonyos szoftvereket, parancsokat elindítaniuk, pl. amelyek valamilyen módon növelhetik egy ember privilégiumát a rendszeren belül. Már csak azért sem, mert ha egy ilyen programot megtör, akkor ezáltal több jogot szerezhet a rendszerben, mint kellene. Jó, ha fontosabb anyagokhoz guest-ek egyáltalán nem férhetnek hozzá, mivel így esetleg megakadályozhatunk egy anonymus fertőzést. 3

A szándékos károkozástól és fertőzéstől menthetjük meg az antivírus programokat, ha megfelelően védett – pl. csak olvasható – helyre installáljuk azokat.

Pszichológiailag bizonyított, hogy az olyan rendszerbe, amely nagyon "reklámozza" önmagát pl. bejelentkezéskor, szívesebben "törnek be", az ilyen rendszerek jobban vonzzák a betörő hajlamú embereket. A mai hálózatok nem feltörhetetlenek és akik elegendő rendszerismeret birtokában vannak, azok képesek is rá.

6.1.5. Makróvírus megelőzés

Makróvírusoknál az eddigieken kívül egyéb megelőzési lehetőségeink is adódnak. Legelőször – a rezidens makróvírusok jellemzőjét figyelembe véve – érdemes lementeni és read-only attribútummal ellátni Word-nél a normál sablont – Normal.dot, Excel-nél az általunk használt automatikus sablonokat és az egyéni makró-munkafüzetet – Personal.xls. Ezzel annyiban akadályozhatjuk a makróvírust szaporodásában, hogy az nem lesz automatikusan betöltve az adott alkalmazás indításakor – hiszen a lementés nem történik meg. Ugyanakkor, ha egy fertőzött állományt megnyitunk, akkor a vírus az alkalmazásból való kilépésig aktív lesz, valamint a fejlettebb makróvírusok le tudják szedni az írásvédettség attribútumát egy apró trükkel: a vírus az Autoexec.bat-ba beszúr egy sort, ami törli az automatikusan betöltődő állományok read-only attribútumát – pl. attrib -r C:\Winword\Template\Normal.dot, így a legközelebbi rendszer indulást követően a makróvírus már fertőzőképesebb. Erre a megoldásra azért van szükség, mert a Word-ben vagy Excel-ben megnyitott állományokkal azok

bezárásáig semmiféle művelet nem végezhető, attribútumuk sem változtatható.

Ha binárisan – pl. Dos-ból – megnézünk egy makrókat tartalmazó állományt, akkor az abban szereplő makrók neveit megtalálhatjuk benne (lehet, hogy utasításokat is, de ez alkalmazásfüggő). Ha ezek a makrónevek egy vírusra jellemzőek – pl. Payload – akkor a file-lal óvatosan kell bánni. Igaz, a makróvírusok utáni keresésnek nem ez a legjobb módja, ugyanis számtalan olyan program van, amely megmutatja számunkra az állományban található makróneveket, sőt a szlovák fejlesztésű Hmvs program még vissza is fejti azokat.

! Valamennyi Microsoft irodai alkalmazásban lehetséges az automakrók tiltása a Shift gomb lenyomásával – a program indításkor vagy egy állomány megnyitásakor. A Word ezen a téren többet kínál: a "DisableAutoMacros" parancstól kezdve makrónyelvi értelmezője nem fogja végrehajtani az automakrókat. Ezt parancsot többféleképpen is végrehajtathatjuk:

- globális makróterületen AutoExec (vagy AutoOpen) makró létrehozásával, mely a tiltó parancsot tartalmazza,
- /m kapcsoló és parancssorba beírás segítségével – pl. Winword /mDisableAutoMacros,
- /m kapcsoló és parancssorba makrónév beírás segítségével – pl. Winword /mAutomakrotiltas, ahol az "Automakrotiltas" az a makrónév, melyben a parancs szerepel.

A Word féle megelőzési módnak a hátránya, hogy az automakrók futtatása újra engedélyezhető, amit a legtöbb makróvírus meg is tesz.

Ha egy dokumentumnak csak a tartalmára vagyunk kíváncsiak, akkor ezt érdemes olyan programmal megtenni, melyben a makrók futása lehetetlen: pl. a Windows 95-ös Wordpad segítségével Word 6.0 és Word 95-tel készített dokumentumokba belenézhetünk, de azokból a makrók nem tudnak aktivizálódni, mert a Wordpad-nek nincs makrónyelvi értelmezője. Sajnos ez a program a Word 97 dokumentum formátumát nem ismeri, viszont a Microsoft által kiadott Wordview már igen, bár ennek használata egy kicsit rizikósabb, mivel bizonyos makrókat képes végrehajtani!

Szintén egy megelőzési mód lehet, ha a Word-ben elkészített anyagot olyan formátumban tároljuk el, ami felépítéséből adódóan nem tartalmazhat makrókat – pl. Word-nél ilyen Rich Text Formátum (RTF). A módszer hátránya, hogy ha egy Word makróvírus már aktív, akkor kizárólag sablonként hajlandó menteni, másrészt lehet, hogy a más formátumba való mentés az anyag "kinézeti" sérülését vonja maga után.

Az irodai alkalmazások többségénél lehetőségünk van lementett dokumentumainkat, táblázatainkat jelszóval levédeni. Több felhasználós környezetben így könnyen korlátozható az állományhoz való hozzáférés, valamint az egyes felhasználók által az állományon végezhető változtatások is korlátozhatóak. Mindez vírus megelőzés szempontjából előnyös és hátrányos is lehet: előnyös, ha a mindenki által használt állományok nem megfertőzhetőek, hátrányos, ha egy fertőzött állományból a vírus nehezen távolítható el a jelszóvédelem miatt.

Az Office 97-től kezdődően lehetőségünk van arra, hogy a betölteni kívánt anyagot az Office megfelelő programja megvizsgálja, tartalmaz-e makrókat. Ha igen, akkor figyelmezteti a felhasználót, majd felajánlja a lehetőséget, hogy makrók nélkül vagy makrókkal együtt töltse be. Bár ez egy elég jó megelőzési mód, mégis vannak hátrányai:

- az Office akkor is figyelmeztet, ha a file-ban menü- és/vagy parancsgomb beállítások találhatóak (sokan ezt elég idegesítőnek tartják ezért ezt az ellenőrzést rögtön ki is kapcsolják),
- hasznos makrókat tartalmazó állományoknál is figyelmeztet,
- hasznos makrókat tartalmazó, de fertőzött dokumentum betöltését a felhasználó engedélyezheti,
- ezt az ellenőrzést "okosabb" makróvírus képes kikapcsolni.

Szintén az Office 97 mellett szólnak a 4.1.6. pontban említett jellemzők: a Word 97 bizonyos makróvírusokat nem fordít át WordBasic-ről VBA-ra, az execute-only makrókat alkalmazó vírusok sem mindig szaporodóképesek és az SR1-el patkolt verziókban globális sablonról (ál)dokumentumba nem lehet makrókat másolni.

Ám a Word 97 előtti verziók használata is jelenthet némi előnyt, hiszen a mai Word makróvírusok legtöbbje már VBA-ban készül, vagyis régebbi változatokon nem működnek.

6.1.6. Antivírus program használat

Lehetőségeink szerint szerezzük be és használjuk rendszeresen a legújabb vírusölőket. Ezek segítségével elérhetjük, hogy minél több és újabb – ismert vagy ismeretlen – vírust azonosítsunk.

Újságokban és antivírus cégek honlapján ritkán lehet találni víruskeresési szignatúrákat. Jó, ha ezeket összegyűjtjük, számos víruskereső képes általunk megadott szekvenciákat is keresni file-okban. Ezeket a szignatúra file-okat érdemes és hasznos továbbadni másoknak is.

Jó, ha használunk rezidens vírusfigyelő programot, fontos, hogy ez már rendszerindításkor elinduljon – igaz, egy ilyen rezidens vírusfigyelő minél több vírust ismer, annál nagyobb memóriára van szüksége és egy kissé lassítja is a gépet.

CRC ellenőrző rendszerünk telepítése előtt biztosnak kell lennünk abban, hogy nem fertőzött rendszerbe telepítjük a programot, ha mégis megtörténik a baj, akkor a detektálásnál nem fog semmiféle változást észlelni, hiszen a vírusok nagy része még egyszer nem fogja megfertőzni a lehetséges, de már fertőzött célpontokat.

Ha komplett rendszerünk minden része már rendelkezik CRC-vel, akkor az újabb állományok rendszerbe való integrálásakor azok CRC értékét is ki kell számolni és megfelelően biztonságos helyre elmenteni. Ezen CRC értékeknek a hasznát fertőzés detektáláskor látjuk majd.

6.1.7. A Comspec – Ramdrive trükk

Néhány évvel ezelőtt nem ekkora és nem ilyen gyors winchesterek voltak, mint amelyek mostanában vásárolhatóak, éppen ezért az ember minden trükköt hajlandó volt bedobni a PC gyorsításának érdekében.

Az egyik ilyen gyorsítási lehetőség a Comspec átirányítása egy ramdrive-ra: mivel a Command.com nem teljesen töltődik be a memóriába, ezért – a Comspec környezeti változó alapján – néha magában keresgél fontos információk után. Ha létrehozunk a memóriában egy ramdrive-ot (Config.sys-ben: device[high] = [megh:][\elér.út\]ramdrive 064 512 010 [/a]) amire felmásoljuk a Command.com-t, akkor a Comspec ramdrive-ra való átállítása után (set COMSPEC=D:\COMMAND.COM, ahol D: a ramdrive) gépünket gyorsabbá tettük, mivel a ramdrive (valójában memória) elérési ideje gyorsabb, mint a winchester.

Ezzel egy kicsit védetté is tettük a Command.com-ot: sok vírus e környezeti változó alapján fertőzi ezt a rendszer shell-t, ez esetben viszont csak a ramdrive-on lévő példány fog megfertőződni, ami egy újabb rendszerindításkor elvész.

6.2. VÍRUSFERTŐZÉS DETEKTÁLÁS

Ha gépünket – a rendszeres vírusellenőrzés illetve a megelőző intézkedések ellenére – megfertőzi egy vírus, akkor annak egy idő után jelei vannak. Ennél a pontnál még nem célunk, hogy a vírust név szerint azonosítsuk, itt ki-

zárólag azt kell megállapítani, hogy a különböző furcsa eseményeket vírus okozza vagy sem.

Mikor gyanakodhatunk vírus jelenlétére? Általában furcsa dolgok sorozata után, bár nem minden esetben lehet vírus a baj okozója (egy-egy példával illusztráltam):

- a programok hossza, létrehozási dátuma megváltozik: ha az állományokhoz hozzáfűző CRC programot használunk, akkor ezek változhatnak, Σ
- megváltoznak a könyvtárbejegyzések, rejtélyes állományok jelennek meg: programhiba miatti leállás után maradhatnak byteszeméttel teli állományok, pl. Windows fagyás esetén,
- lassabban futnak a programok, esetleg lefagynak: tévedni emberi dolog, a programokat emberek írják...,
- szokatlan ábrák jelennek meg a képernyőn pl. táncoló labdák, röpködő lepkék, a képernyőn szabályos időközönként átfutó alakzatok, fura szövegek: egy nagyon "kedves" munkatársunk vírusdemóval akar megtréfálni
- memória méret változás, hibás lemezfelületek számának növekedése: több rezidens programot töltöttünk be, mint amennyit eddig szoktunk használni, vagy egy lemezkezelő program nem működik helyesen,
- végrehajtható állományok eltűnnek: valaki játékprogramot másolt fel gépünkre és nem volt elég hely,
- az eddig jól működő programunk sűrűn nyúl lemezhez: nincs futtatva a cache program,
- egy víruskereső kizárólag a memóriában vírust talált: vaklárma vagy egy programunk ugyanúgy kezel egy adott eszközt, mint a "megtalált" vírus,
- lemez vagy file-tartalom elveszik: a cache program nem tudott írni lemezre újraindítás előtt,
- egyéb hibák jelentkeznek a gépen: egy hardver eszköz elromlott.

Makróvírusoknál a jelenlét egészen más dolgokban jelentkezik:

- az alkalmazás nem hajlandó más formátumban lementeni a szerkesztett anyagot, csak olyanban, ami makrót is tartalmazhat, Σ
- rosszul és/vagy egyáltalán nem lehet lementeni,
- lassabban menti le az anyagokat,
- új automatikus sablonok jelennek meg,
- új menüpontok, eszköztárak, ismeretlen makró-nyomógombok, űrlapmezők jelennek meg,

- már meglévő menüpontok, eszköztárak tűnnek el úgy, hogy közben egyik kollégánk sem piszkált bele gépünkbe,
- a lementett anyagok egy része jelszavas lesz,
- ismeretlen szavak, értékek kerülnek dokumentumba, táblázatba – pl. nyomtatáskor,
- a Windows beállításai hirtelen megváltoznak.

A vírusdetektálás több vírusvédelmi programnál a fertőzés megelőzéshez szorosan kapcsolódik: pl. az ellenőrző összeges védelemnél az előző lépésben kell kiszámolni a CRC értékeket, ami alapján a változást detektálni lehet. Ilyen védelmi programok használatánál fokozottan kell ügyelni arra, hogy tiszta rendszerlemezről induljon a gép, különben nem tudja helyesen összehasonlítani az újonnan kiszámolt CRC értékeket a régebben kiszámoltakkal.

Ám a CRC ellenőrzés korántsem elegendő: rendszeres időközönként és mindig a legújabb verziót kell használni víruskereső és a heurisztikus kereső programjainkból, melyekkel az összes lehetséges meghajtót ellenőrizni kell, hogy ne maradjon rejtett fertőzés.

6.3. VÍRUSFERTŐZÉS AZONOSÍTÁS

Ha meggyőződtünk arról, hogy nem vakriadóról van szó, akkor az egyik legfontosabb kérdés, hogy mivel és honnan hurcoltuk be a vírust, a másik, hogy pontosan milyen vírussal van dolgunk, hiszen minden vírust típusától függően másképp kell kezelni és irtani, valamint fel kell deríteni, hogy azonosításáig milyen pusztítást végezhetett. Sajnos előfordulhat olyan eset, amikor az azonosítás már nem lehetséges, mert a vírus rombolási fázisába lépett.

A fertőzés azonosító kategóriába a specifikus víruskereső programok tartoznak, melyek közül érdemes egyszerre többet is használni, így nagyobb az esély a pontos azonosításra. Bár Magyarország nem tartozik a vírusíró nagyhatalmak közé, fontos, hogy nemzeti specialitásokat ismerő programot is használjunk.

Víruskereső programjaink indítását érdemes egy batch programra bízni, ez megment minket a víruskereső programok egyenkénti indításától és a sok gépeléstől.

Ha specifikus víruskereső programunkat rendszeresen futtatjuk, akkor az azonosítás egyszerre történhet a detektálással. Fontos, hogy az azonosító program vírusmentes környezetben fusson!

6.4. VÍRUSFERTŐZÉS UTÁNI HELYREÁLLÍTÁS

A helyreállítás a legnehezebb feladat, de ha szigorúan betartunk minden ajánlott szabályt és tanácsot, akkor elérhető a minimális adatvesztéssel járó helyreállítás is!

A helyreállítás folyamán 3 kategóriába kell sorolni az egyes rendszerelemeket:

- azok az elemek, amelyeket a vírus megfertőzött, ekkor a fertőzött objektumok helyreállításával vagy mentésből való visszatöltésével lehet próbálkozni,
- azok az elemek, amelyeket a vírus tönkretett vagy módosított, ekkor kizárólag a módosított elemek visszatöltésével kell foglalkozni,
- azok az elemek, amelyekkel a vírus nem csinált semmit.

A fertőzött objektumok helyreállítása négyféleképpen lehetséges:

- vírusölővel, ami a fertőzés előtti állapotukba állítja vissza a fertőzött célpontokat,
- a vírusnak egy hibáját kihasználva, a vírus önmagát távolítja el a rendszerből (!),
- sajátos kézi módszerekkel,
- a biztonsági másolatból történő visszatöltéssel.

Σ

Minden egyes perc nagyon fontos, minél több időt bajlódunk a vírus eltávolításával, annál jobban elszaporodhat. Cselekedjünk gyorsan, de meggondoltan.

Ha file- vagy boot/partíciós vírussal van dolgunk, akkor első lépésként indítsuk gépünket egy tiszta rendszerlemezről: a floppy behelyezése után nem elegendő csupán a Ctrl-Alt-Del-t használni, mert sok vírus ezt a fajta rendszerindítást ki tudja védeni: vagy lekapcsolni vagy reset-elni kell a gépet. Rendszerindítás után nyugodtan dolgozhatunk a fertőzött lemezen is: átválthatunk akár a boot-vírussal fertőzött winchesterre is, onnan a vírus nem tud aktivizálódni, mivel ilyenkor a boot program nem töltődik be és nem hajtódik végre. Makróvírusoknál ilyen rendszerindítás eleve nem szükséges, elegendő, ha a makrónyelvi lehetőséggel ellátott alkalmazásból kilépünk.

1

Rendszer újraindítás illetve az alkalmazásból való kilépés után le kell menteni azokat az állományokat, amelyek az utolsó mentés óta nem lettek backup-olva. Ha programvírussal van dolgunk, akkor a programok közül semmit nem szabad lementeni, még akkor sem, ha adott programról nincs másolatunk: egy programot lehet pótolni, de az adatokat nem, ráadásul lehet, hogy eltennénk későbbre a fertőzést.

2

Mentés után az egész rendszer újrainstallálása, visszatöltése illetve antivírus program használata között a fertőzött rendszerrészek kényességétől és a backup létezésétől függően dönthetünk.

3a Mentés visszatöltéshez vagy újrainstalláláshoz akkor kell folyamodni, ha a vírus olyan módon és/vagy mértékben módosított vagy rombolt, hogy a visszaállítás még a vírus eltávolítása után is szinte lehetetlen, vagy gyorsabban végzünk az újratelepítéssel, mintha az keresnénk, hogy mi nem sérült. Ez a megoldás nagyon hosszú és borzalmas munka, akár órákig is eltarthat. Visszatöltés esetén egy valamire nagyon figyelni kell: a visszatöltött anyagok között ne legyen vírusfertőzött rész.

3b A megfelelő vírusirtó program kiválasztásakor figyelembe kell venni, hogy nem mindegyik képes teljesen visszaállítani a célpontot fertőzés előtti állapotába. Ez programoknál abban nyilvánulhat meg, hogy az eredeti hossz valamennyi byte-tal növekedni fog és lehet, hogy a hossznövekedésből fakadóan nem fognak működni – különösen a másolásvédett – programok. A makróvírusok Dos-beli irtása igaz, hogy gyors, de kevésbé megbízható, mert az egyes speciális Windows rutinkönyvtárak használatának helyettesítése több programozást igényel, így nem biztos, hogy a program tökéletesen távolítja el a makróvírust (pl. nem kerülnek vissza az eredeti beállítások, ez egyes menüpontok). Ilyen esetekben legjobb a mentés visszatöltés.

Ám a makróvírusok irtása történhet Windows-ból és magából a makrónyelvi értelmezővel ellátott alkalmazásból is: a Windows-os irtás előnye, hogy a Windows rutinkönyvtárait lehet használni, ezáltal megbízhatóbb, viszont lassabb. Az adott alkalmazásba ültetett vírusvédelmi program gyakorlatilag tökéletesen tudja irtani az egyes vírusokat és csak akkor fut, amikor szükség van rá, hátránya viszont, hogy nem teljesen védett az alkalmazás esetleg nem publikált lehetőségeit maximálisan kihasználó makróvírusok ellen, hacsak abszolút mélyen be nem ágyazódott.

Kézi módszerek

Ha azonban nem rendelkezünk a vírust irtó programmal, akkor kellő szakértelem és rendszerismeret esetén írunk egyet (rosszul megírt vírusölő program több kárt okozhat, mint maga a vírus) vagy sajátos kézi módszerekkel próbáljuk kipucolni a rendszerből.

Nézzük, milyen kézi módszerek lehetségesek:

1. Boot/partíciós vírusok kezelése

- A Dos-ban is megtalálható Sys parancs segítségével nemcsak az egyes rendszer állományokat tehetjük fel egy lemezre, hanem új

boot szektor programot is generál a cél lemezre: pl. winchester boot szektor újragenerálás – "Sys C:", floppy boot szektor – "Sys A:". A floppyra rákerülő felesleges rendszer file-okat ezután már lehet törölni.

- Szintén a Dos-ban megtalálható Fdisk paranccsal új partíciós tábla programot generálhatunk, ha használjuk a kevésbé ismert /mbr kapcsolóját. A partíciós tábla vírusok csak a partíciós tábla programot cserélik le, a partíciós tábla adatait (melyik partíció hol kezdődik, melyik az aktív) nem bántják, ezért egy "Fdisk /mbr" parancs kiadásával felülírhatjuk a vírust.
- Harmadik kézi módszerünk is a Dos egyik parancsának a használatán alapul: formázzuk le a fertőzött lemezt, mivel az újraépíti a boot szektort. A Dos 5.0-tól kezdődően használjuk a /U kapcsolót – pl. "Format A: /u", különben a vírus felélesztéséhez szükséges információ a lemezen marad: a Dos Unformat parancsával akaratlanul is, de újraéleszthetjük a fertőző gócot. A formázás a floppyról szinte biztosan leveszi a vírust, winchesternél ez nem mindig sikerül.
- Amennyiben tudjuk, hogy a vírus hol tárolja az eredeti boot-szektort vagy partíciós táblát, akkor nyert ügyünk van: elegendő egy disk editor program, aminek a segítségével az eredeti boot programot beolvassuk a memóriába, majd pedig az eredeti helyére visszatesszük a program segítségével, így a vírust felülírtuk. Persze ez sok száz floppy lemeznél elég kényelmetlen és egy idő után unalmas.
- Egyes segédprogramok – pl. Norton DiskDoctor – képesek arra, hogy az eredeti boot programot újra felépítsék, vagy pedig a lemezen megkeressék az elveszettnek hitt partíciós táblát.
- Jó esetben a partíciós tábla és a boot-szektor ki van mentve egy fileba egy disk editor program segítségével, fertőzés esetén – a tiszta rendszerlemezről való indítás után – az eredeti helyükre kell visszamásolni ezeket.
- Az előző ötlet alapján meg lehetne azt is csinálni, hogy egy tiszta floppy lemez boot szektorát beolvassuk a memóriába, majd ezzel felülírjuk a fertőzöttet (winchesterre ez már kevésbé kivitelezhető, mert kicsi a valószínűsége, hogy ugyanakkora méretű és partíció kiosztásút találjunk). Fontos, hogy csak ugyanolyan típusú lemezre lehet másolni a boot-szektort, mint amilyenről leszedtük, pl. 360 Kbyte-os lemezre csak 360 Kbyte-os lemez boot-szektorát másol-

hatjuk, 1.2 Mbyte-osra 1.2 Mbyte-ost stb. Lássuk most ennek a gyakorlati kivitelezését, a Debug program segítségével:

C:\>Debug	elindítjuk a Debug-ot és egy tiszta floppy-t teszünk a meghajtóba
-	ez a Debug promptja
-L CS:0000 0 0 1	jó boot program betöltése a memóriába
-R CX	
:000	ha ez az érték nem 200, akkor írjuk át arra
-R DX	
:000	ha ez az érték nem 0, akkor írjuk át arra

Most kell kivenni a tiszta lemezt és betenni az ugyanolyan fertőzöttet

-W CS:0000 0 0 1	jó boot-program kiírása a lemezre
-Q	kilépés

Vannak olyan vírusok – pl. a One-Half, amelyek kódolják a winchester bizonyos területeit úgy, hogy a visszakódoláshoz szükséges adatok a partíciós program helyén találhatóak. Éppen ezért minden egyes kézi módszerrel óvatosan kell bánni, mert ha egyszerűen csak felülírunk, akkor az elkódolt területek tartalmát már nem lehet visszaállítani. Ilyenkor a megfelelő vírusölő és visszakódoló program megjelenéséig inkább ne vagy nagyon keveset használjuk a gépet.

Ugyanez a szabály, ha olyan vírust azonosítottunk gépünkön, amelyhez nem létezik vírusirtó program és kézi módszerrel sem lehet kitakarítani.

2. Programvírusok kezelése

Programvírusokra sajnos már nincs ennyi kézi módszer, ezeket nehezebben lehet kezelni.

- A felülíró típusúaknál nem lehet irtásról beszélni, a fertőzött programokat törölni kell.
- A CEB vírusoknak az irtása a vírusokat tartalmazó file-ok törölésével jár.
- A lopakodó vírusok jellemzésénél már említettem az átnevezéssel járó vírus beugratási lehetőséget: ha minden futtatható programot

átnevezünk nem futtathatóvá, akkor a vírus eltűnik a programokból, egy reset után a memóriából is. Ezek után nem marad más hátra, mint a visszanevezés.

3. Makróvírusok kezelése

Ha makróvírust kézzel szeretnénk kiirtani, akkor legelőször az összes automatikus sablont nevezzük át más kiterjesztésűre, Word-nél nézzük meg, hogy parancssorban megadott makrót nem indít-e (ezt a paramétert ideiglenesen iktassuk ki), ezzel máris elértük, hogy az alkalmazás nagy valószínűséggel aktív makróvírus nélkül fog elindulni.

Általában minden alkalmazásnak van egy vagy kettő olyan menüpontja, amelyben a makrók elérhetőek. Ezeknél a menüpontoknál lehetséges törölni az egyes vírusmakrókat, feltéve ha tudjuk, hogy a látható makró listából melyik tartozik a vírushoz és melyik nem. Sajnos a lopakodó makróvírusok éppen ezeket a menüpontokat rendszeresen célba is veszik, viszont Word-nél a Fájl|Sablonok|Szervező|Makró menüpont eltűntetése kevésbé jellemző. Éppen ezért ezen a menüponton keresztül érdemes próbálkozni: az egyik ablakban megnyitjuk a fertőtleníteni kívánt (ál)dokumentumot, majd töröljük a megfelelő makrókat. Az átnevezett automatikus sablonokat ugyanúgy kell ellenőrizni, mint az összes többi egyéb állományt.

(Ál)dokumentumból való kézi makróvírus törlés után már csak egy dolgunk van hátra, hogy a sablont visszaminősítsük dokumentummá. Ezt kizárólag úgy lehet megtenni, hogy nyitunk egy új állományt a megtisztított sablon alapján, majd mentjük dokumentum formátumban. Legkönnyebb dolgunk akkor van, ha csatolási bejegyzéssel szaporodó Word makróvírussal találkozunk, ekkor egyszerűen törölni kell a csatolt sablont.

Excel esetében mindig fel kell fedni az összes elrejtett munkafüzetet vagy sablont az Eszközök menün belül, csak ezután törölhetőek a vírusmakrók. Excel-nél érdemes megvizsgálni az idegen (és a saját) makróbeépülőket is, ugyanis a felhasználók többsége csak az Excel féle makróbeépülőket alkalmazza.

A kézi makróvírus irtás az automatikus sablonok visszanevezésével fejeződik be.

A vírus irtása után – legyen az kézi vagy program általi – minden lemezünket újra át kell vizsgálni, hogy ne maradjon rejtett (pl. asztalfiókban felejtett floppy) fertőzés.

Végső esetben – ha a vírust nem tudjuk eltávolítani – értesítsük vírusszakértőnket, vagy egy víruselhárítással foglalkozó céget. Ők azok, akik a

legjobban értik dolgukat és nagyon hálásak tudnak lenni, ha olyan vírussal "lepjük meg" őket, amelyet vírusölőjük még nem képes detektálni vagy irtani.

Milyen károk keletkezhetnek egy fertőzés folyamán? Csak egy apró lista: antivírus szakemberek kiszállási díja, óradíja, gépek egyenkénti ellenőrzése (gépidő kiesés), az összes adathordozó ellenőrzése (gépidő kiesés), adatok helyreállítása (ha van mentés), programok helyreállítása, esetlegesen megrongált hardver kicserélése... stb.

Egy komolyabb fertőzés tehát nagyon sok kárt okozhat, még egy kis cégnél is, éppen ezért érdemes inkább a megelőzésre és az antivírus programok használatára koncentrálni.

7. ANTIVÍRUS PROGRAMOK

Mióta megjelent az **első riport a számítógépes vírusokról** különböző számítástechnikai újságokban, nagyon gyorsan számtalan cég beszállt a harcba, **sorban bocsátották ki vírusvédelmi programjaikat.** Ezeknek a cégeknek rövid idő alatt kellett:

- megtervezni egy védelmi rendszert,
- megírni a programkódot,
- a felhasználóknak dokumentációt írni,
- alfa és béta tesztet csinálni, majd
- piacra dobni a kész terméket.

Ezt a sok dolgot rövid idő alatt megcsinálni monumentális feladat.

Aztán elkezdődött a harc a felhasználók kegyeiért és mindenki **más-más trükkhöz** folyamodott: az egyik ilyen hatásos vevőcsalogató reklám arról szólt, hogy az adott vírusirtó program **hány vírust képes megkeresni és/vagy irtani**. Repkedtek a különböző számok, valójában kevésbé szaporodott ennyire az egyes keresők tudása: a szakemberek több ezer – most már tizenötezer – vírusról beszéltek, valójában ezeknek a vírusoknak csak a töredéke volt "**eredeti**", az összes többi az eredetiek kisebb-nagyobb mértékben módosított változata, variánsa volt. Ezekhez a variánsokhoz pedig lehet, hogy **csak a szekvenciát kellett egy picit módosítani.** Akik nem bírták a versenyt, megszűntek, felvásárolták őket vagy egybeolvadtak más, addig konkurens céggel. Sajnos kevés cég és program maradt azon a piacon, ami most már nemcsak a vírusok elleni hősies küzdelemről híres.

Mint azt a vírusfertőzés megelőzésnél ajánlottam, **egyszerre több víruskereső programot is érdemes használni**, melynek oka, hogy a legtöbb vírusölőre **96–97%**-os védelmet ígérnek íróik, de mondjuk csak **tizenötezer féle vírus esetén a fennmaradó 3–4% 450–600 vírust jelent**, ami elég jelentős szám. Ha azonban **több, különböző vírus ismeretű antivírus termékeket használunk, akkor ez a % lecsökkenhet akár 1%-ra is!** Ugye mennyivel jobban hangzik?!

A következőkben néhány általam **kedvelt és használt antivírus programot szeretnék bemutatni**, a teljesség igénye nélkül: csak a főbb jellemzőiket említettem meg, sikeres használatukhoz egyaránt szükséges a programokhoz mellékelt dokumentáció valamint a későbbiek folyamán a gyakorlati tapasztalat. Egy valamiben megegyeznek: **valamennyinek létezik szabadon terjeszthető változata.**

Ám korántsem kell azt hinni, hogy csak ezeket a programokat érdemes használni: számítástechnikai újságokban rendszeresen *megjelenő mélyebb szintű antivírus program leírások és összehasonlító tesztek alapján talán könnyebb dönteni.*

7.1. McAfee VirusScan

Valaha az egyik legelterjedtebb és legkedveltebb antivírus programcsomag volt itthon, mára egy kissé háttérbe szorult. Az első **DOS**-os verziók óta megjelent **Windows 3.1, Windows 95, Windows NT, OS/2, MacOS, Linux, Solaris, FreeBSD** és **SunOS** alá is.

Külön termék a **WebScan** – ami **Web** böngésző és levelező programok által behurcolt vírusok felfedezésére alkalmazható – és a **ScreenScan**, ami **Windows 95** alatt a képernyővédő beindulásakor a **Scan**-t elindítja (kímélve a felhasználó idegeit, aki minden bekapcsoláskor nem a víruskeresést akarja nézni). A **DOS**-os verziónak egy védett üzemmódos változata is létezik, ez kevésbé sűrűn panaszkodik futtatásakor a kevés memóriára.

A program **szabadon terjeszthető változatát "evalution copy"-nak** hívják, magánszemélyek 30 napig, cégek és szervezetek egy kipróbálás vagy egy bemutató erejéig használhatják ingyenesen. Az "evalution copy" elnevezéshez azért ragaszkodnak, mert a cég el akarja kerülni a shareware szó alapján mindenkinek eszébe jutó lebutított szoftverek fogalmát.

Használatuk épp olyan egyszerű, mint régen, és a nemzetközi forgalomban lévő vírusokat továbbra is kitűnően megtalálja. Elég sok vírust képes eltávolítani, *gyakorlati tapasztalataim alapján egyedül makróvírus eltávolító része hagy maga után némi kívánnivalót: ha makróvírust távolít el a Normal.dot-ból, akkor nem állítja vissza az eredeti menüpontokat – pl. Cap vírus esetén az Eszközök\Makró menüpontot.*

Rezidens védelme sok memóriát igényel és lassítja a gépet, ha minden lehetséges ellenőrzést bekapcsolunk. Alapértelmezés szerint a futtatható programokon kívül Winword dokumentumokat és sablonokat (*.do?) valamint az Excel állományait (*.xl?) figyeli.

Windows 95-ös keresőjének a felülete szinte teljesen megegyezik a Windows 95 beépített filekereső program felületével, ám itt a keresési eredménylistában a fertőzött állományok és a vírus neve látható.

A program minden változata külön adatbázisokban tárolja a kereséshez és az irtáshoz szükséges információkat, melyek Internet-en és különböző BBS-eken keresztül letölthetőek. Érdekes módon a programra továbbra is jellemző, hogy egyes vírusokat nemzetközileg elfogadottól eltérően nevez.

Mindezeket leszámítva egy nagy vírusismeretű, de más antivírus programokhoz képest lassú és memóriaigényes programról van szó. *Lassúsága talán hatalmas vírustudásának (is) köszönhető.*

További információ: http://support.nai.com
http://www.2fkft.hu

7.2. THUNDERBYTE ANTIVIRUS

A Thunderbyte Antivirus programcsomag szintén nem újonc a vírusfronton. Felépítése meglehetősen egyedi: több egymással együttműködő, egymást kiegészítő programból áll. A programokat assembly-ben fejlesztették, ezért méretük nem túl nagy és rendkívül gyors, saját file-kezelési eljárásának köszönhetően képes észrevenni a **lopakodó makróvírusokat. DOS, Windows 3.1, Windows 95** és **Windows NT** verziói léteznek.

Rezidens vírusvédelme szabályozza a file-okhoz, a memóriához és a lemezekhez való hozzáférést: beállítástól függően az ellenőrző program először ismert vírusok után kutat a futtatott programban. Megadható, hogy mely programok lehetnek rezidensek, melyek férhetnek hozzá közvetlenül a lemezhez, melyek változtathatják meg más programok attribútumát stb. *Itt-ott néha idegesítő is lehet a használata: pl. a lemezvédelem annyira jó, hogy minden lemezhez való hozzáférést észrevesz és így állandóan zaklatja a felhasználót kérdéseivel.*

Parancssorból vagy menüből indítható keresője nagyszerű heurisztikájáról és nemes egyszerűségéről híres. A menükön belül elérhető vírusokról információt csak regisztrált felhasználóknak ad, ezt a cégtől kapott regisztrációs file létezésének ellenőrzésével tudja meg.

A **Windows 95**-ös változat nem alkalmazza a **DOS**-os verzió saját filekezelési eljárását, a rezidens védelmi rész pedig kevésbé különül el a keresőtől és kevésbé hatékony víruscsapda.

Hasonlóan az előző programhoz ennek is létezik egy **vírus definíciós adatbázisa, mely rendkívül könnyen frissíthető ismertebb Internet-es antivírus helyekről.**

További információ: http://www.norman.nl
http://www.thunderbyte.com

7.3. F-SECURE

A harmadik "legöregebb" harcos a vírusfronton. Rendelkezik **DOS**, Windows 3.1, Windows 95, Windows NT és OS/2 változattal. A **DOS**-os verzió használata jobban ajánlott, mert csak ez képes heurisztikus keresésre, igaz, annak érzékenysége nem paraméterezhető.

A fő verzióváltás 2.x-ről 3.x-re még 1997-ben történt. A két változat közötti különbség, hogy a 3.x verzióktól kezdődően megszűnt a Virstop rezidens vírusvédelem, a program keresési szekvenciái nem bővíthetőek, viszont képes tömörített állományokban is vírus után keresni a tömörítő program használata nélkül, támogatja a Microsoft SMS-t valamint az SNMP-t.

Maga az F-Secure is képes makróvírusok keresésére, mégis a 2.24 verziótól kezdődően erre egy külön programot hoztak létre F-Macro – Windows alatt F-Macrow névvel. Az F-Macro kizárólag **DOS** alatt vagy **DOS** ablakban, az F-Macrow Windows 3.1, Windows 95 és Windows NT alatt makróvírusok keresésére és irtására használható.

A program egyik erénye a hálózati üzemmód: hálózaton keresztül a felhasználók F-Secure upgrade-je automatikus lehet, fertőzés esetén a rendszergazdának üzenetet küld vagy rögtön el is távolítja a vírust stb.

Dos-os változatának menüből való futtatása kényelmesebb, de parancssorból is indítható, ekkor rögtön meg kell neki adni, hogy mit akarunk ellenőrizni és hogyan: milyen file-okban keressen vírust, mely könyvtárakban milyen file-okat, vagy melyik meghajtót ellenőrizze teljes egészében, készítsen-e report file-t, a megtalált vírusokkal mit tegyen, tömörített állományokat és a memóriában önkicsomagoló futatható programokat ellenőrizze-e stb. Az ellenőrzés beütemeztethető konkrét időpontokra vagy holt időre (amikor nincs semmiféle tevékenység).

További információ: http://www.datafellows.com
http://www.2fkft.hu

7.4. HMVS

Ezt a kitűnő antivírus programot Szlovákiában fejlesztették ki kizárólag makróvírusok keresésére és irtására, **DOS**-ban, de működik Windows 3.1, Windows 95 és Windows NT alatt is. A 2.6-os verziót 1998 novemberében cserélték le a 3.0 verzióra, amely már nemcsak Word és Excel, hanem Access makróvírusokat is képes lekezelni.

Véleményem szerint egy abszolút profi makróvírus ellenes program, mely Word 6.0 és Excel 5.0-tól kezdődően Office 97-ig bezárólag, neuron hálózatos mesterséges intelligencia segítségével képes minden ismert és is-

meretlen makróvírus felfedezésére. Neuron hálózatos technikája miatt futásához szükség van koprocesszorra is.

Képes makrók visszafejtésére is – megfelelő paraméterezéssel – akár 12 nyelven! Támogatja a hosszú fileneveket (Windows 95) és rendkívül gyors. Alapértelmezés szerint vírusirtáskor a fertőzött állományokat más kiterjesztéssel elmenti.

A program letöltési helye: ftp.bke.hu /pub/mirrors/sac
ftp.elf.stuba.sk /pub/sac
ftp.netlab.sk /pub/sac
http://sac-ftp.gratex.sk

7.5. VIRUSBUSTER

A VirusBuster programcsomag a Hungarian VirusBuster Team (HVT) terméke, létezik **DOS**, Windows 3.1, Windows 95, Windows NT és Novell Netware változata. Honlapjukról a **DOS**-os változat kipróbálásra letölthető (valamint a vírus adatbázis frissítés).

A **DOS**-os programcsomag tartalmaz egy programot a már megtörtént fertőzés felismerésére és megszüntetésére (Chkvir), egy rezidens részt a fertőzések megelőzésére, meggátolására (Virsec) és ezen kívül néhány segédprogramot: pl. Onehalf, Getto2000, Kaczor, Honecker vírusokat irtó programocskákat, valamint Word és Excel makróvírusok elleni makrómodulokat is.

A Chkvir vírusirtó a víruskeresések összes módszerét használja, külön adatfileból vett információk alapján képes vírusokat eltávolítani. Mutációk és ismeretlen vírusok felfedezésére is alkalmas: a program tartalmaz egy x86 processzor emulátort, aminek a segítségével védetten tudja futtatni a vizsgálandó programokat, miközben figyeli a vírusgyanús funkciókat.

A Virsec vírusvédelem egy különálló számítógép vagy akár egy egész hálózat vírusvédelmét el tudja látni, képes egyes részeit az EMS-be feltelepíteni. A Virsec-kel kijelölhetjük, hogy mely file-okat mennyire akarunk védeni, a beállításokat képes elmenteni állományba, hogy a beállított adatokat később is fel tudja használni.

Bár menüből kényelmesebb, parancssorból is indítható. Ekkor meg kell adni az ellenőrizendő meghajtót és alkönyvtárakat és az egyéb paramétereket. A program menüje Turbo Vision-ben íródott, egérrel is lehet használni. Indításakor memóriaellenőrzés után próbálja beolvasni az aktuális meghajtó könyvtár struktúráját.

Lemezolvasási módjait is beállíthatjuk: vagy **DOS** lemezműveletekkel, vagy BIOS-on vagy device driver-en keresztül olvasva keresi a vírusokat. Ennek segítségével a lopakodó vírusokat is könnyen lebuktathatja.

Képes tömörített állományokba is belekukkantani, a tönkretett file-okat szükség esetén törölheti, az immunizált programokról az immunrutint képes eltávolítani. Word és Excel futtatása nélkül képes makróvírust irtani.

Vírusismerete kisebb, mint más nagy külföldi programoké, de a magyar specialitásokat kitűnően ismeri.

További információ: http://www.vbuster.hu

7.6. VIRWARE

Ez a debreceni fejlesztésű program – mely **DOS**-os és Windows-os felülettel is rendelkezik – különösen az itthon gyakran felbukkanó vírusokra specializálódott, jó heurisztikával rendelkezik. A vírusok irtásához szükséges információkat egy külön állományban tárolja.

A fejlesztő külön programot írt a makróvírusok keresésére és irtására, de ez hozzátartozik a programcsomaghoz. Hatalmas előnye, hogy kis méretű, így könnyen felfér egy vírusmentesítő floppy-ra. Az ellenőrzés ütemezhető napi egyszeri futtatására, vírus találat esetén üzenetet képes küldeni a Netware konzolra. Kiegészítő védelemként kitűnően használható.

További információ: http://www.edu.dote.hu/~virware
http://www.elender.hu/~virware

7.7. IRTO

Ez egy kizárólag **DOS**-os felületű, de Windows alatt is használható, egyetlen futtatható állományból álló vírusirtó, mely nem próbálja felvenni a versenyt egyéb programokkal, de az itthon előfordult és komoly gondokat okozó vírusokat szinte egytől egyig képes irtani. Tömörített állományokba csak a tömörítő program meghívásával tud belenézni, néhány makróvírust is ismer.

További információ: http://winnie.obuda.kando.hu/~nfl/irto.html

7.8. ULTIMATE VIRUS ELIMINATOR

Ez szintén egy hazai vírusirtó program, mely igaz, hogy **DOS**-os felületű, de tapasztalataim szerint kitűnően működik Windows 3.1 és Windows 95 alatt is. Hasonlóan a Tbav-hoz ez a program is assembly-ben készült, hasonlóan kicsi és gyors, sőt heurisztikus keresője talán még jobb is. A program rezidens védelemmel nem rendelkezik.

A shareware változatot magánemberek szabadon használhatják, minden egyéb esetben a kereskedelmi verzió használata érvényes. A szabadon terjeszthető változat nem tartalmaz dokumentációt, viszont teljesen magyar nyelvű minden menüpont és segítséget is lehet kérni. A programon belül lehetőségünk van csak keresésre illetve keresésre és irtásra is, az általa keresett vírusokról egy külön menüponton belül információt kérdezhetünk le.

A vírusirtáshoz és kereséshez szükséges információkat 7 db.dlm kiterjesztésű, un. Dynamic Loadable Module-ban tárolja. Ha ismeretlen makróvírust talál és annak kiirtását kérjük, akkor a fertőzött állományban található összes makrót törli, ami nem mindig előnyös.

További információ: http://astrobase.bajaobs.hu/~m_arts
http://www.hcbyte.hu/~m_arts

7.9. VSUM

Ha a felhasználó vírust talál, akkor legtöbbjük szeretne róla minden információt megszerezni, pl. melyek a jellemző tulajdonságai, hogyan működik, mikor aktivizálódik, mit fertőz meg, mivel lehet ellene védekezni stb.

A Vsum egy hypertext formátumú szöveges adatbázis, mely különböző vírusok jellemzőit tartalmazza űrlap szerint: a vírus fertőzési és rombolási tulajdonságai, a felfedezéshez vírusazonosító sztringek, név és ország szerint vagy akár hosszadatok alapján pillanatok alatt visszakereshető a kérdéses vírus leírása. Ha valaki szeretne többet megtudni az adott vírusról, akkor nem kell semmi mást megtennie, mint tovább olvasnia a vírusleírást. Az egyes leírások kinyomtathatók.

Az adatbázis óriási hátrányának tartom, hogy egyetlen makróvírusról sem ad információt, kizárólag file- és boot vírusok adatait tartalmazza.

Még 1992 elején volt próbálkozás arra, hogy ezt a programot és adatbázist meghonosítsák nálunk, egy újság mellékleteként meg is jelent, de tudtommal anyagi okok és kellő érdeklődés hiányában az adatbázis szövegeinek fordítása abbamaradt.

További információ: http://www.vsum.com

A fentiekben felsorolt programokon kívül további vírusellenes programok tölthetőek le a HMVS-nél említett Slovak Antivirus Center (SAC) ftp site-járól: ftp.elf.stuba.sk/pub/sac. Ennek a site-nak a tükrözése a Budapesti Közgazdasági Egyetem ftp szerverén is megtalálható: ftp.bke.hu/pub/mirrors/sac. Aki esetleg nem férne Internet-hez, annak tudom javasolni a Chip Magazin CD mellékletét, amelyen minden hónapban megtalálhatóak a SAC új programjai és egyéb leírások.

Az Új Alaplap című újság CD mellékletén is lehet antivírus programokat találni.

További ajánlott antivírus program beszerzési hely a hamburgi egyetem ftp szervere: ftp.informatik.uni-hamburg.de, ahol a Virus Test Center (VTC) egyéb anyagai – tesztek, leírások – is megtalálhatóak.

8. VÍRUSGYŰJTEMÉNYEK LÉTREHOZÁSA ÉS KARBANTARTÁSA

Egy jól karbantartott vírusgyűjtemény nagyon fontos segédeszköze az **antivírus-kutatóknak**. A gyűjteményt lehet használni antivírus szoftverek tesztelésére, az eddig megjelent több ezer vírusról való ismeretek rendszerezésére és nem utolsó sorban más antivírus-kutatókkal való cserére. Sajnos, **ha avatatlan vagy gonosz kezekbe kerül** egy ilyen gyűjtemény, akkor az ugyanúgy elősegítheti újabb, az **eddigieknél gonoszabb és okosabb vírusok születését** is, de jó értelemben is segíthet: a vírusból ellesett programozási trükköt más, **hasznos programban** fel lehet használni. Egy ilyen gyűjtemény létrehozása és **rendszeres karbantartása** nem egyszerű feladat, mivel manapság nagyon sok vírus van és vírusok számának növekedése elég gyors.

Az **antivírus-kutatókon** kívül az **informatikával megszállottabban** foglalkozó fiatalok is gyűjtenek számítógépvírusokat, csereakciókat szerveznek, ahol az is számít, hogy ki mennyire új vírussal rendelkezik: csak egy kicsit változtatnak egy-egy víruson és máris **új példányok vannak a gyűjteményükben**. Hiába: új kor – új szokások.

A számítógépes alvilág által terjesztett gyűjtemények sok vírust, vírusszimulátort, trójai programot, vírusgenerátort, szöveges állományt és teljesen jó, fertőzetlen programot is tartalmazhat. Régebben ezeket a gyűjteményeket vírus csere-bere **BBS**-ekről, manapság az egyes vírusíró csapatok honlapjáról lehet letölteni.

8.1. KICSOMAGOLÁS, SELEJTEZÉS, SZÉTVÁLOGATÁS

Elmúltak azok az idők, amikor az összes létező, mintegy tucat vírus egy floppy-ra ráfért. Manapság egy **jó gyűjtemény több ezer file-ból áll, ami Mbyte**-okat foglalhat el a lemezen, éppen ezért – a hely megtakarítás érdekében – közismert és közkedvelt tömörítő programokat szokás használni.

Ha **új vírusgyűjteményt** kapunk, akkor kicsomagoláskor ügyelni kell arra, hogy

- megfelelő mennyiségű szabad lemezterületünk legyen,
- a kitömöríthető állomány esetleg könyvtárstruktúrákat is tartalmazhat, ami a kitömörítő program nem megadott kapcsolója miatt el is veszhet,

- az archive állomány – biztonsági okokból – **jelszóval titkosított** lehet (antivírus kutatók kizárólag jelszóval védett állományokat cserélnek egymással),
- a **tömörített file-ban néhány állomány ugyanolyan néven szerepelhet,** mégis más a tartalma (kétszer egymás utáni kicsomagolással és átnevezéssel kivédhető a probléma).

Kicsomagolás után következhet azon állományoknak – pl. vírus magazinoknak – a törlése, amelyek már nekünk is megvannak, melyet a **módosított állományok kiszűrése** követ. Jellemzően előforduló módosítások:

- a file elejét néhány szöveges karakterrel felülírták,
- a file elejéből hiányzik valamennyi byte,
- program belépési pontja a program méretén túlmutat – pl. rosszul eltávolított vírus esetén,
- a file appendelő vírust tartalmaz, de hiányzik a vírusra mutató ugróutasítás – ebből még kinyerhető a vírus,
- egyéb – pl. nyomkövetéshez használt INT 3 – utasítás lett beleírva a víruskódba – ha ez éppen a vírus azonosító szignatúra rész, akkor kijavítható és szaporítható lesz a vírus.

Olyan módosított állománnyal is lehet találkozni, amiből a benne található vírus továbbra is futásképes vagy kinyerhető:

- a file végéhez néhány sztring van hozzácsapva – pl. **régebben a Tntvirus antivírus** program által hozzáragasztott "**MsDOS**",
- a file végéhez származási helyét (**BBS**, **Internet oldal**) népszerűsítő szöveg van illesztve,
- a file végéhez – antivírus program által képzett – **ellenőrző összeg van hozzáfűzve** (ez könnyen eltávolítható antivírus programokkal),
- a vírus olyan futtatható programot fertőzött meg, amit később **átalakítottak önmagát futás közben memóriában kicsomagoló programmá** (egy kísérleti futtatással vagy SAC shareware gyűjteményében is megtalálható, tömörített és/vagy kódolt futtatható programokat kibontó Unp-vel is eltávolítható),
- többszörösen vírusfertőzött file (debugger vagy bináris file editorral szétválaszthatóak a vírusok).

Ezek után következhet a vírusok és az egyéb állományok – trójai programok, vírus-szimulátorok, vírus magazinok, utility programok, vírusgenerátorok – szétválogatása. Vírus-szimulátorok jellemzően azért szoktak gyűjteményekbe belekerülni, mert egyes víruskeresők ezekben is éles vírust vél-

nek felfedezni, a legtöbb gyűjtő pedig hisz az ilyen rosszul detektáló programoknak. A **szétválogatás** azért is fontos, mert a kollekciót antivírus programok tesztelésére, detektálási arány meghatározására is használhatjuk és tisztességtelen lenne, ha ez utóbbi egy nem vírusos file meg nem találása miatt csökkenne.

Egy gyűjteményben találhatunk olyan file-okat is, amelyek a kiterjesztésüknél fogva végrehajthatóak lennének, de mégsem azok, mert "csak" egy boot-vírust tartalmaznak: valaki olyan disassemblerrel vagy debugerrel próbálta visszafejteni, ami csak végrehajtható kiterjesztésű állományokkal hajlandó dolgozni. Ezeket az állományokat korántsem szabad összekeverni a dropper programokkal! Jobb esetben a kiterjesztés takarja, hogy milyen floppy-ról lett belemásolva file-ba a boot-vírus – pl. .360 = 360 Kbyte, .144 = 1.44 Mbyte.

8.2. SZAPORÍTÁS

Selejtezés után következhet azoknak a vírusoknak a kiszűrése, **amelyekkel már rendelkezünk:**

- víruskereső programokkal ellenőrzést végzünk az állományokon,
- a generált riport file-ok alapján könnyen kiszűrhetjük, hogy mely vírusokkal rendelkezünk már.

Ha nagyon sok szabad lemezterületünk van, akkor jó, ha több olyan fileunk is van, amelyet ugyanaz a vírus fertőzött meg, hivatásos antivírus cégeknél egy-egy vírusból annyi példány van, ahány célpontot képes megfertőzni.

A **gyűjteményben már szereplő vírusos file-ok törlése** után a maradék file-ok vizsgálata következik. A vírussal való **fertőzöttségre** a lehető legjobb bizonyíték az, ha a file-ban van olyan kódrészlet, amely indítása után képes **önmagát reprodukálni**, ha mégsincs, az még nem azt jelenti, hogy a file nem tartalmaz vírust: lehetséges, hogy a vírus csak olyan környezetben képes "élni", amelyben készítették – pl. a **Ping-Pong vírus** csak 8086 processzoros gépeken fut. Ám korántsem biztos, hogy az a program, amely elindítása után más programhoz ragasztja magát, az valódi vírus: pl. a **Druid vírus** csak a germjéből képes elindulni, a megfertőzött állományokból a továbbiakban már nem!

Szaporításnál ügyelni kell arra, hogy egy objektumot egyszerre több vírus is megfertőzhetett, viszont előnye, egyrészt az esetleges burkolat (pl. népszerűsítő szövegek) elvesztés – mivel az nem tartozik a vírustesthez,

másrészt a kisebb megfertőzött célpontok tárolásához kevesebb lemezterület szükséges.

File-vírusoknál tiszta környezetben kell próbálkozni valamilyen rövid csali program – melyet előzőleg több könyvtárba is felmásoltunk – **szándékos megfertőzésével**. Ha a fertőzés sikerült, akkor össze kell hasonlítani azokkal a példányokkal, amelyekkel egy családba tartozik, ha már rendelkezünk vele, akkor törölni kell. Ez az összehasonlítás nehézségekbe ütközhet oligomorf és teljesen polimorf vírusok esetén. Ha a fertőzés nem sikerült, akkor más konfigurációjú gépnél illetve más operációs rendszereken is próbálkozni kell. Ha ez sem megy, akkor következik a kód közvetlen visszafejtése és analizálása.

Boot-vírusoknál a két hasonló, esetleg egy családba tartozó boot-vírust "installálni" kell két ugyanolyan típusú formázott floppyra, majd ezután Diskcomp paranccsal összehasonlítani. A *boot-vírus file-ba mentése helyett érdemesebb a* ***dropper programot*** *megtartani*, mivel az kényelmesebb a gyűjtő szempontjából: teljesen önállóan az eredeti boot-szektort és a vírust is a megfelelő helyre teszi.

8.3. OSZTÁLYOZÁS, RENDSZEREZÉS

Egy **vírusgyűjteményt** kizárólag valamilyen hierarchia szerint szabad létrehozni és rendszerezni, ez legjobban egy olyan könyvtárstruktúrában valósulhat meg, ahol a főágak az egyes víruscsaládok, a mellékágak a csoportok és variánsok, végül az adott vírust tartalmazó alkönyvtárak következnek. A gyűjteményen belül külön könyvtárban kell tárolni azokat az állományokat, melyekben a víruskeresők vírust jeleztek, de azok mégsem szaporodtak. Ezzel a hierarchiával már a családfából látható, hogy mekkora vírusgyűjteménnyel rendelkezünk.

A valóban új vírusokat tartalmazó file-okat be kell illeszteni a régi vírusgyűjteményünk rendszerébe. Ha a víruscsaládhoz és a csoporthoz való tartozást az antivírus programok riport file-jai alapján nem tudjuk eldönteni, akkor szükség lehet a vírus visszafejtésére is. **Antivírus cégek** nagyon gyakran valamilyen programozási fogás alapján tudják csak besorolni az új vírust egy családba, ugyanis minden vírusírónak van egy jellemző stílusa, programozási fogásai. **Analízishez és visszafejtéshez** sokféle segédprogramot kell felhasználni, hiszen pl. egy-egy **disassemblernek vagy debugernek másmás az erőssége, visszafejtési hatásfoka**. A vírusok visszafejtése nem egyszerű dolog, de nagyon izgalmas munka.

Ha mégsem található semmi hasonlóság egy vírus és a régi családok között, akkor egy új család első tagját kaptuk meg, de lehet, hogy egyben az utolsót: számtalanszor megtörtént, hogy valaki próbaképpen írt egy vírust, de nem folytatta a fejlesztését és mások sem tartották arra érdemesnek, hogy átírogassák itt-ott, vagyis variánsai sem lettek. Rendszerint nem túl sok új vírus nyerhető egy hatalmas és rendetlen kollekcióból.

8.4. TESZTELÉS

A vírusgyűjtemény ezek után még mindig nem használható fel teljesen antivírus programok tesztelésére, hiszen igaz, hogy egy file-vírus keresési teszt az adott **antivírus program lefuttatását jelenti a gyűjtemény könyvtárstruktúráján, de mi van a polimorf és a boot-vírusokkal**?! *Egy polimorf vírussal több ezer (!) file-t kell szándékosan megfertőzni – ami elég időigényes, hogy lássuk a vírusellenes program találati arányát, a boot/partíciós vírusokat azért kell installálni lemezekre, hiszen azokat nem file-okban keresik a programok.*

A **tesztelési célú szaporítás** nemcsak idő-, hanem helyigényes is, ugyanakkor rendkívül hasznos lehet a **vírusellenes programfejlesztőknek**: megtalálhatják szinte az összes programhibát, a vakriadós eseteket ki tudják szűrni, a vírus megnevezési problémákat megoldhatják (a nemzetközileg elfogadottól eltérő néven jelzi ki a program) stb.

8.5. VÍRUS CSERE-BERE

Vírusok csere-berélésére régebben **BBS**-ek, **manapság** inkább az **Internet** a jellemző. A BBS-es korszakban a vírusos részhez általában csak akkor kapott valaki hozzáférési jogot, ha egy új vírust töltött fel – igaz, ezután mennyiségi megkötés nélkül tetszőleges számú vírust tölthetett le akár forráslistával együtt, Internet-es világunkban ez a kitétel és mennyiségi korlát alapvetően nem létezik.

Az antivírus cégeknek, kutatóintézeteknek szinte még nem volt kapcsolatuk egymással, amikor a vírusterjesztők és vírusgyűjtők létrehozták első ilyen típusú **BBS**-eiket. Ezeken **BBS**-eken és honlapokon más országokbeli vírus **BBS**-ek telefonszámai és honlapok linkjei találhatóak, tartják egymással a kapcsolatot és rendkívül szervezettek.

Sajnos ezek a **BBS**-ek és a honlapok inkább rossz-, mint jó hatásúak. Jó hatás alatt értem azt, hogy egy antivírus cég tetszőleges számú, új vírust letölthet, ugyanakkor ez a letöltés **gonosz szándékkal – vírusterjesztés, forráslisták felhasználásával újabb vírusok kifejlesztése –** is történhet. Szerte a világon találhatók ilyen **BBS**-ek és honlapok: USA-ban, Németországban, Svédországban, Olaszországban, Nagy-Britanniában, a FÁK-ban, Szlovákiában és Magyarországon is ! Magyarországon nem létezik vírusterjesztők elleni törvény, így bárki szabadon létrehozhat egy ilyen **BBS**-t vagy – szabad Web területet biztosító szolgáltató cégeknél – honlapot. Igaz ez utóbbinál nem is igazán lenne utolérhető a tettes.

Saját vírusgyűjteményemet bővítgetve többször is találkoztam olyan vírusos file-okkal, melynek a végén egy vírus csere-bere **BBS** hirdette önmagát. A szöveg a következő volt:

> "This file was downloaded from the Virus Exchange BBS.
> Phone: +359-220-4198, working hours: 20:00 – 6:00 GMT.
> Give it a call!"

Ami magyarul kb. ennyit tesz:

> "Ez a file a Vírus Csere-bere BBS-ről lett letöltve.
> A telefonszám: + 359-220-4198, nyitva: 20:00 – 6:00 GMT.
> Hívd fel!"

Az említett telefonszám szófiai (Bulgária) volt és magyar időszámítás szerint: 19:00-tól hajnal 5:00-ig lehetett hívni, mint **BBS**. Az 1990-ben létrehozott Virus Exchange BBS rendszeroperátora a *Szófia egyetemen akkor számítástechnikát tanuló Todor Todorov volt.* Ez a saját otthonában elhelyezett **BBS** két részre volt osztva, az egyik részen az antivírus programokat teljesen szabadon bárki letölthette, míg a vírusos részhez való hozzáférési jog feltétele egy új vírus feltöltése volt. Ha valaki új vírust töltött fel, hozzáférési jogot kapott az egész kollekcióhoz: letölthette azt a vírust, amelyiket csak akarta, de akár az összeset is. *A **BBS** felhasználói mindenféle hamis* – pl. Goerge Bush New Yorkból, Szaddam Husszein Bagdadból, Ozzy Ozburn – és egyéb kitalált neveken jelentkeztek be. Ez a fajta szabad vírusletöltés lehetővé tette, hogy Bulgáriában készített vírusok nagyon távol és elég gyorsan el tudtak terjedni, ugyanakkor az országban olyan vírusok jelentek meg, amelyek erre a régióra nem voltak jellemzőek: pl. így terjedt el 1991-ben Bulgáriában a Typo. A **BBS**-t vírusíró konferenciák tartására is felhasználták.

Az első kiadásban említett magyar **BBS**-ek telefonszáma azóta megszűnt, az egyik ellen a törvény erejével léptek fel.

Az Internet roham léptékű terjedésével lehetővé vált, hogy hasonlóan a **BBS**-ekhez bárki bármilyen vírust letöltsön vírusíró/gyűjtő csapatok honlapjáról és ráadásul olcsóbban is. Ellentétben a **BBS**-es filozófiával itt már nem szükséges új vírus feltöltése, viszont talán még szervezettebbek, mint valaha. *Egy-egy ilyen honlapra Internet-en felhasználható kereső segítségével pillanatok alatt rá lehet bukkani, én ettől függetlenül inkább nem közölném itt egynek a címét sem.*

Mint láthattuk, *egy jó hierarchiával rendelkező és nem vírusos állományoktól tiszta gyűjtemény felépítése nem könnyű dolog, mert rengeteg munkát és szakértelmet igényel. Egy ilyen vírusgyűjtemény karbantartása és rendszerezése csak egy az antivírus-kutatók feladata közül.*

9. ELLENŐRZŐ KÉRDÉSEK ?

PC VÍRUS TIPOLÓGIA

1. Milyen szempontok szerint csoportosíthatók a "klasszikus" vírusok?
2. Milyen funkcionális egységekből áll egy vírus?
3. Mi a reprodukciós rutin feladata?
4. Milyen eseményeket figyelhet az aktiválási mechanizmus?
5. Mi a boot szektor? Hogyan fertőzheti meg vírus?
6. Mi az oka a boot vírusok kiemelkedő veszélyességének?
7. Milyen kiterjesztésű állományok lehetnek a programvírusok potenciális célpontjai?
8. Hogyan kapcsolódhatnak a programokhoz a nem felülíró típusú vírusok?
9. A felülíró vírustípus mindig működésképtelenné teszi a gazda programot? Miért?
10. Milyen DOS sajátosságot használnak ki a CEB programok?
11. Mi a különbség a klasszikus boot vírus és a hibrid vírusok terjedési módja között?
12. Hogyan csoportosíthatjuk az objektív rutin szándéka szerint a vírusokat?
13. A memória mely részeibe telepedhet rezidens vírus?
14. Mi a kapcsolat a szoftver megszakítások és a rezidens vírusok között?
15. Hogyan terjednek a nem rezidens vírusok?
16. Mi jellemző a lopakodó technikát alkalmazó vírusokra?
17. Miért alkalmazzák egyes vírusok a mutációs technikát?
18. Mikor sorolunk egy kártékony programot a "trójai" kategóriába?
19. Mi lehet egy hálózat felderítő trójai program konkrét célja?
20. Hogyan terjed egy ASCII vírus?
21. Hogyan aktiválható egy ASCII vírus?
22. Milyen módszerrel csapható be egy megerősítést igénylő DOS parancs?
23. Mi a feladata az ANSI.SYS rendszer állománynak?
24. Hogyan lehet egy ANSI bombát "élesíteni"?
25. Mit tud az ANSI vírusok jövőjéről?
26. Hogyan szerezhetünk vírust a DIR parancs kiadásával?
27. Mik a program férgek?
28. Mi lehet a program férgek célpontja?
29. Milyen vírus tároló programokat ismer?
30. Milyen vírus jellemzők állíthatók be egy vírusgenerátor program segítségével?

KÜLÖNLEGES VÍRUSTÍPUSOK

1. Hogyan változtathatja meg megjelenési formáját egy vírus?
2. Melyek egy polimorf vírus funkcionális egységei?
3. A polimorf vírus mely részei kódolatlanok?
4. Mi a mutációs vírusok változtathatóságának alapja?
5. Soroljon fel néhány vírus mutálásra alkalmas technikát!
6. Hogyan változtatható a víruskód hossza mutációs vírusoknál?
7. Hogyan tévesztik meg a víruskeresőket a permutáló vírusok?
8. Milyen lopakodási módszereket ismer?
9. Milyen változatai vannak a CEB vírusoknak?
10. Melyek a FAT vírus fontosabb jellemzői?
11. Mi történik, ha programot másolunk FAT vírussal fertőzött lemezről, és a vírus nem aktív?

MAKRÓVÍRUSOK

1. Mi az azonosság és az eltérés a batch (ASCII) és a makróvírusok között?
2. Mit tud a makróvírusok múltjáról és (közel) jövőjéről?
3. Melyek a fő okai a makróvírusok hirtelen elterjedésének?
4. Hasonlítsa össze a "klasszikus" programvírusok és a makróvírusok jellemzőit!
5. Milyen elemeket tartalmazhat egy Word6 dokumentum?
6. A dokumentumokhoz képest milyen többlet elemei vannak egy dokumentum sablonnak?
7. Mi az eltérés a makrónyelv szempontjából a Word jelenleg használt változatai között?
8. Hogyan töltődhet be a memóriába egy makróvírus?
9. Soroljon fel néhány olyan módszert, ahogy egy makróvírus átveheti a vezérlést?
10. Mi a Word vírusok elsődleges célpontja? Miért?
11. Milyen külső erőforrásokat vehet igénybe egy makróvírus büntető rutinja?
12. Hogyan lehet a Word-ben egy makróvírust kódolni?
13. Milyen rejtőzködési lehetőségei vannak egy Word makróvírusnak?
14. Milyen mutálási lehetőségei vannak egy Word makróvírusnak?
15. Hogyan segít az RTF formátum használata a vírusvédelemnek?

16. Lehetséges-e Word vírust szerezni egy Excel táblázat használata által?
17. Milyen jelei lehetnek egy Word vírus jelenlétének?
18. Mit tud az Excel vírusokról?
19. Hogyan, honnan töltődhet be egy Excel makróvírus?
20. Milyen módon válhat aktívvá egy Excel makróvírus?
21. Hol helyezkednek el az Excel makróvírusok?
22. Mit tud az AnmiPro makróvírusokról?

VÍRUSVÉDELMI MÓDSZEREK

1. Milyen változások utalnak egyértelműen vírusfertőzésre?
2. Milyen elvek alapján működhetnek a víruskeresők?
3. Mi a hosszabb szignatúrák alkalmazásának előnye, illetve hátránya?
4. Mikor alkalmazunk heurisztikus víruskeresőt?
5. Mely esetekben jelez vírusveszélyt egy heurisztikus kereső?
6. Milyen információ szükséges ahhoz, hogy egy vírus eltávolítható legyen?
7. Hogyan figyelmeztet a vírusveszélyre az ellenőrző összeges védelem?
8. Mi a "védőoltás" módszer lényege?
9. Hová helyezhetünk el - természetesen vírustól függő - védőoltást?
10. Melyek a rendszerfelügyelő védelem alkalmazásának előnyei, illetve hátrányai?
11. Mi jellemzi a jó hardver vírusvédelmet?

VÍRUSVÉDELMI TANÁCSOK

1. Mit lehet tenni a vírusfertőzés megelőzése érdekében?
2. Mik azok a víruskapuk? Írjon néhány példát!
3. Mire hívná fel (a vírusvédelemmel kapcsolatban) kollégái figyelmét?
4. Hogyan, milyen gyakran készítsünk biztonsági másolatokat?
5. Mit kell tartalmaznia a vírusok elleni fellépéshez készített floppy lemeznek?
6. Soroljon fel néhány olyan jelenséget, amely vírus jelenlétére utal!
7. Milyen módszerekkel szüntethető meg a vírusfertőzés?

ANTIVÍRUS PROGRAMOK

1. Milyen funkcionális egységekből áll egy univerzális vírusvédelmi rendszer?
2. Soroljon fel néhány ismertebb vírusvédelmi rendszert!

10. IRODALOMJEGYZÉK

SZÁMÍTÁSTECHNIKAI SZAKKÖNYVEK, SZAKDOLGOZATOK:

1. Nagy Gábor: Vírusvédelem a PC-n. Computerbooks, 1995. Budapest
2. Dr. Nagy Gábor: Makróvírusok. Műszaki Könyvkiadó, 1996. Budapest
3. Dr. Nagy Gábor: Makró-kozmosz. TallSoft Könyvek, 1997. Budapest
4. Toldi Péter: A számítógépes vírusok elleni védelem lehetőségei és eszközei. Gábor Dénes Főiskola, Szakdolgozat, 1998. Budapest
5. Tóth J. Szabolcs: PC Vírusok. LSI, 1995. Budapest
6. Tóth J. Szabolcs: PC Vírusok (kiegészítés). LSI, 1998. Budapest

SZÁMÍTÁSTECHNIKAI ÚJSÁGCIKKEK:

A Chip Magazin rendszeresen közölt vírusokkal kapcsolatos cikkeket. Továbbiakban: CHIP.

1.	Tóth J. Szabolcs. Fertőző Word-dokumentumok!	CHIP 1996/04. 89. o.
2.	Rudnai Tamás – Tóth J. Szabolcs: Makrójárvány	CHIP 1996/06. 72. o.
3.	Kis János: Új makróvírus-ellenes program	CHIP 1996/12. 62. o.
4.	Kis János: Áthelyeződő súlypontok	CHIP 1996/12. 62. o.
5.	Kis János: Az a hír, hogy nincs hír	CHIP 1997/01. 58. o.
6.	Kis János: McAfee új stratégiája	CHIP 1997/03. 84. o.
7.	Rudnai Tamás: 6ékony védelmek	CHIP 1997/08. 74. o.
8.	Rudnai Tamás: Hisztéria! Heurisztika!	CHIP 1997/09. 91. o.
9.	Rudnai Tamás: Eső előtti köpönyeg	CHIP 1997/10. 130. o.
10.	Csábi József: Új seprű jobban seper	CHIP 1997/10. 134. o.
11.	Rudnai Tamás: Makrómánia	CHIP 1997/11. 120. o.
12.	Kis János: Új utakon a McAfee	CHIP 1997/12. 64. o.
13.	Kis János: Kiterjedt honi védelmek	CHIP 1997/12. 64. o.

Az Új Alaplap rendszeresen közölt vírusokkal kapcsolatos cikkeket. Továbbiakban: Új Alaplap.

1.	Szappanos Gábor: Makróvírusok	Új Alaplap 1998/05. 63. o.
2.	A hónap témája: Vírusriadó	Uj Alaplap 1998/09.
3.	Galántai Zoltán: Az Internet Worm	Új Alaplap 1998/11. 63. o.
4.	Szappanos Gábor: Hazai körkép	Új Alaplap 1998/11. 65. o.

LEÍRÁSOK, ÉRTEKEZÉSEK, FELMÉRÉSEK:

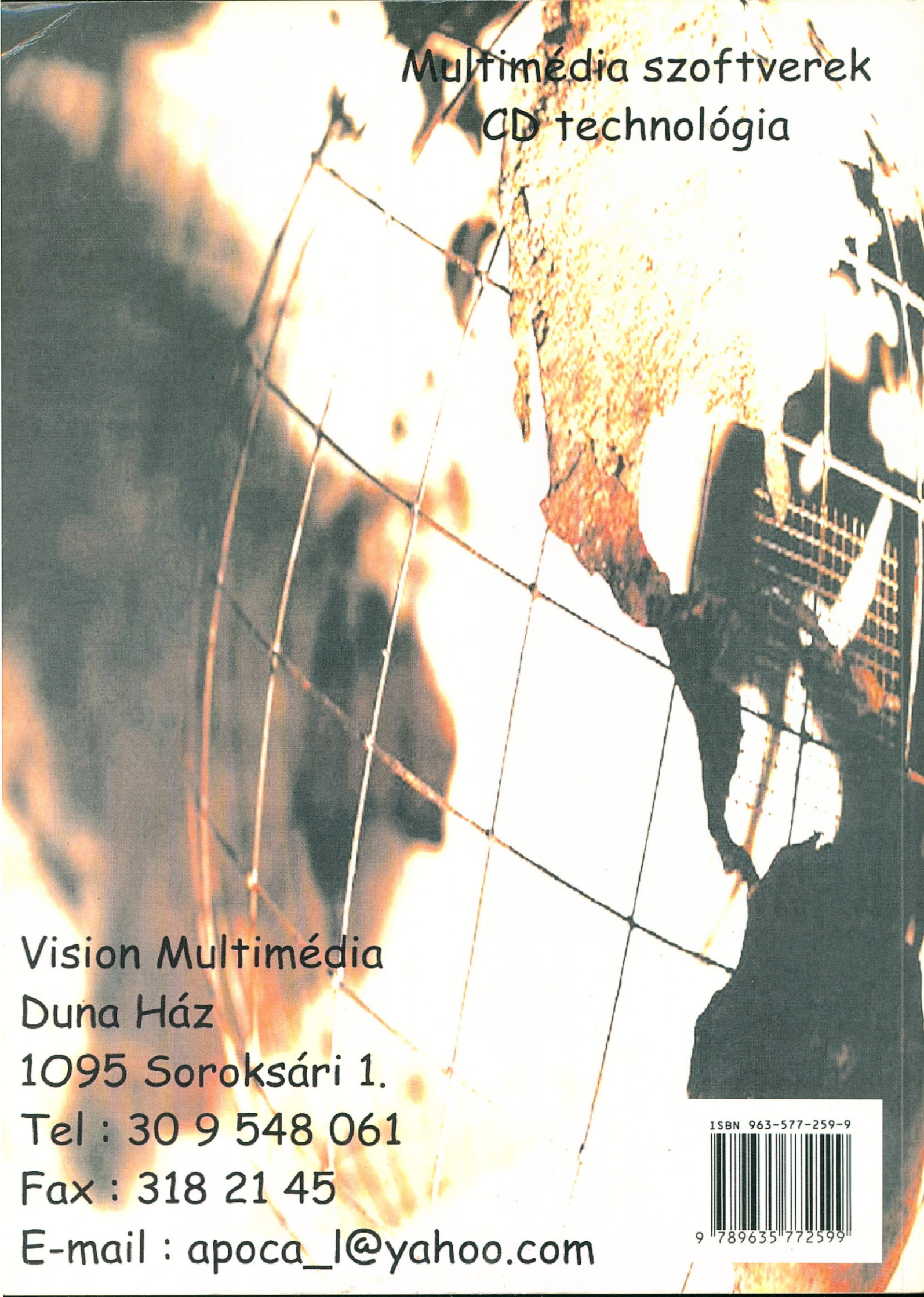
1. Data Fellows: Microsoft Excel macro viruses. 1997
2. Data Fellows: First Ami Pro macro virus. 1996
3. Joel McNamara: Document Macro Viruses. 1994
4. Joel McNamara: Excel Document Macro Viruses. 1994
5. Matthew Probert: The Virus Researcher Handbook 3rd Edition
6. Morgan Cyber Systems: Laroux. 1996
7. NHA: First Excel Macro Virus, aka Laroux. 1996
8. NCSA 1997 Computer Virus Prevalence Survey
9. Richard John Martin: Ms Word 6.x Macro Viruses FAQ V2.0, 1996
10. Softland News: Excel Sofa
11. Stiller Research News: Macro Viruses. 1997
12. Stiller Research News: Excel Macro Viruses. 1997
13. Stiller Research News: New MS Access Macro Virus. 1998
14. Stiller Research News: TOX/Detox MS Access Macro Virus. 1998

SZÁMÍTÁSTECHNIKAI SZAMIZDAT ÚJSÁGOK, ÚJSÁGCIKKEK:

1. Alchemy: Advanced Macro Virus Techniques Issue #1 #2
2. AuRoDrEpH: The Underground MS Word 6.x Macro Viruses FAQ V2.0
3. b0z0: Wordmacro Viruses
4. Dark Night: The Macro Virus Writing Tutorial 1-2.
5. Executioner: Polymorphism
6. Knowdeth / Metaphase: Batch Virii
7. So0ky: MS-Word Macro Viruses
8. Tegwar: Computer Virus Funny Business With Win Word Documents
9. VicodinES: Macro.Poppy Construction Kit v1.0b Documentation
10. VicodinES: What's New in VMPVK v.1.0b
11. VicodinES: What is „Class Object Infection"?
12. VicodinES: Why Access Macro Viruses Will Never Become a Problem
13. Wavefunc: Batch viruses, Issue 1-2-3, 1995

PROGRAM SÚGÓK:

1. Microsoft Word 6.0 help
2. Microsoft Excel 5.0 help
3. Microsoft Access 2.0 help
4. Microsoft Office 95 help
5. Microsoft Office 97 help



Multimédia szoftverek
CD technológia

Vision Multimédia
Duna Ház
1095 Soroksári 1.
Tel : 30 9 548 061
Fax : 318 21 45
E-mail : apoca_l@yahoo.com

ISBN 963-577-259-9
9 789635 772599