TUDNIVALÓK A HEURISZTIKUS KERESÉSRŐL

Az a gyanús, ha nem gyanús

Sorozatunk nyolcadik epizódjában a kéretlen reklámprogramokról lesz szó. Azt már tudjuk, hogy ezek nem vírusok, és nem is férgek, sőt még kártevőnek sem szabad nevezni őket, mert különben vaskos pereket veszünk a nyakunkba. Jó lenne tudni, mennyire veszélyesek, hogyan kerülnek gépünkre, és főképp: miként szabadulhatunk meg tőlük? A cikk végére minden kiderül.

Sorozatunk kilencedik epizódjában a víruskeresés technikájáról lesz szó. Röviden áttekintjük, mivel ad többet, illetve mást a hagyományos adatbázis-alapú (szignatúra szerinti) kereséshez képest a heurisztika.

A különféle víruskeresők lassan már húsz esztendeje velünk vannak. Bár a kezdeti DOS-os világban parancssoros és Turbo Pascal-szerű menüs eszközöket használtunk, a legelső időkben kizárólag a minta alapján történt a vizsgálat. Ez egy jó ideig remekül működött: amíg meg nem jelentek a különböző variánsok, mutációk, illetve – hogy a technikai szempontok mellett az idősík is belépjen – az egyre gyakoribb és egyre növekvő számú változatok miatt a szakemberek kénytelenek voltak kiegészíteni az egyébként jól szolgáló, szignatúraalapú keresést. A heurisztika kifejezetten előnyös volt makróvírusok esetében, míg az egyéb bináris állományoknál problémát okozott a fokozottabb vakriasztás előfordulása. A heurisztikát alkalmazó programok komoly előnye, hogy nemcsak a már ismert, hanem a még ismeretlen kártevőket is képesek észlelni.

Ilyenek voltak a víruskereső programok a DOS világában – jó 15-20 éve

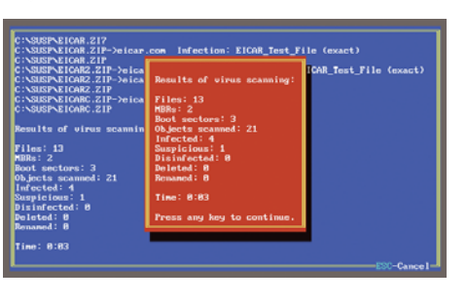
Mi is ez?

Ha felütjük az Idegen szavak szótárát, a heurisztika címszó alatt a következőket olvashatjuk: „A heurisztica görög eredetű szó, a feltalálás, a valamire való rájövés művészete; az elméleti kutatás logikai eljárásainak és módszerbeli szabályainak rendszere”.

A vírusok elterjedésének kezdeti időszakában komoly szerepet játszott az úgynevezett nem dokumentált, illetve alacsony szintű utasítások használata; ezeket az utasításokat többnyire az operációs rendszer és egyéb rendszerszintű alkalmazói programok hajtattak végre. Közismert ilyen utasítás volt a lemez szektorszintű, a fájlrendszer kötöttségeit megkerülő direkt írása. Ezt használták például a particionáló- és formázóprogramok, de ennek segítségével települtek a bootvírusok is. Nem csoda, hogy ezen utasítások alkalmazása a kiemelkedő szakmai tudás és hatékony programozás jele is lehetett, de a heurisztikus víruskeresők általában gyanús jelként értékelték, ha ilyenre bukkantak a vizsgált programban – és tegyük hozzá, többnyire nem is alaptalanul.

Heurisztika – Vegyünk egy példát!

Szemléltetésképpen egy kutyafajtát fogunk keresni többfajta állat között, legyen ez a Louisiana Catahoula Leopard Dog. Ez egy spanyol és indián eredetű fajta, eseménydús történelemmel, és mindössze 1977 óta kezelik önálló elismert fajtaként, zömmel az USA-ban tartják. Azoknak, akik korábban sosem láttak ilyen állatot, homályos elképzelés élhet a fejükben, de valószínűleg nem gondolnak macskafélére, inkább valamilyen leopárdpettyes kutyát képzelnek maguk elé a név hallatán. És nem is tévednek nagyot. A heurisztikus keresés is valahogy így működik: bár nem tudjuk pontosan, mit is keresünk, de vannak elképzeléseink arról, hogy milyen „fajtajegyek” alapján fogunk majd azonosítani a kártevőt.



Az F-Prot programban már igen korán megjelent a heurisztika: ily módon próbálta meg felvenni a versenyt a sok variánssal támadó makró- és egyéb vírusokkal szemben

A jobb alsó sarokban találjuk a Louisiana Catahoula Leopard Dogot. Nem macskaféle – pláne nem leopárd –, hanem egy különleges kutyafajta

Ha az azonosítási technikáknál a klasszikus, adatbázisra (vírusszignatúra) épülő detektálást vesszük alapul, akkor ez egy olyan kétfázisú szabálygyűjtemény, amely egy futtatandó programot vizsgálva vagy engedi annak végrehajtását, vagy a megszakítást javasolva megóv minket a következményektől. Ha viszont egy állomány egy újfajta, eddig ismeretlen fenyegetést tartalmaz, ez a módszer csődöt mond.

A virtuális kódemuláció és a homokozó (sandboxing) segíthet az elemzésben, de itt azt kell eldönteni egy adott programról, hogy az ártalmatlan vagy veszélyes.

Ha lerajzolunk egy táblázatot betűkkel, és azt a feladatot adjuk a jelentkezőknek, hogy keressenek meg bennük adott szavakat, az valószínűleg rövidebb-hosszabb idő alatt mindenkinek menni fog. Ha viszont azt kérnénk, hogy valaki találjon meg benne egy olyan szót, amelyet nem ismer – nos, itt már csődöt mond a mintaegyeztetésen alapuló észlelés.

Ha egy víruskereső jelez egy tiszta állományra – amire nem kellene neki –, azt vakriasztásnak, idegen szóval false positive-nak hívjuk. Az adatbázisalapú keresés is hajlamos a hamis riasztásokra, hiszen nem az egész vírust, hanem csak egy mintát keresünk a vizsgálandó állományban. Ha például egy bizonyos népszerű rajzfilmfigurát keresünk az interneten, nem szükséges tudnunk a nevét ahhoz, hogy megtalálhassuk. Üssük be például a Google keresőbe, hogy „gorilla” és máris kapunk 17 millió találatot. A mi gorillánk is benne van a merítésben, de ott van mellette majdnem 17 millió hamis találat is. Szűkíteni kell a kört például a rajzfilmekben szereplő lényekre, a „cartoon gorilla” kifejezésre már „csak” 416 ezer találat születik, amiből a számunkra 415 999 fals. Természetesen felismerhetjük közöttük a „mi” gorillánkat, ám ha nem emlékszünk pontosan gyerekkorunk rajzfilmjére, nem feltétlen fogjuk az eredetit megtalálni. Ha végül beírjuk, hogy „Magilla Gorilla”, első helyen megkapjuk azt a bizonyos figurát, amelyet eredetileg kerestünk. Ennek mintájára azok a vírusleírások, amelyek nem elég specifikusak, hamis találatokat is fognak eredményezni, és nem lesznek képesek egy fenyegetést egyértelműen azonosítani.

Könnyű a dolgunk, ha fix karaktersorozatot kell megtalálni akár függőlegesen, akár vízszintesen. De mi van akkor, ha nem tudjuk megmondani pontosan, mi az, amit keresünk?

Egy bizonyos gorillát keresünk, de a „gorilla” szóra több mint 17 millió találatot kapunk

Szűkítsük a kört a rajzfilmekben szereplő jószágokra, így már „csak” 416 ezer majmunk van

Itt látható a Magilla Gorilla, ő ez a csodalény, erős, ügyes, a feje sem üres, kész főnyeremény :-)

Generikus észlelés

Az úgynevezett generikus észlelés már egy továbblépés: ismeretlen objektumot vizsgálunk, hogy a kritikus szempontok szerint mennyiben hasonló a rosszindulatú kódhoz. Ha nagyon hasonló, feltételezhetjük, hogy a már ismert kártevőnek egy új variánsa. Ismét egy példával rukkolunk elő. Aki már látott ír szettert, de angol szettert még sosem, képes lesz felismerni azt a hasonló jellemzők révén.

Ugyanez a számítógép nyelvén így nézhet ki: van egy képzeletbeli kártevőnk: „Win32/Ez_csak_egy_példa” néven. Az alábbi jellemzőit ismerjük:

– három speciális kulcsot ad a Rendszerleíró adatbázishoz,
– saját levélküldő (SMTP) motort tartalmaz,
– letölti e-mail címlistánkat,
– levelet küld az alábbi tárgysorral (subject): „Itt a világ vége”.

Ha most fájlokat ellenőrzünk, feltűnhet egy olyan, amelyik az alábbi tulajdonságokkal bír:

– három speciális kulcsot ad a Rendszerleíró adatbázishoz,
– saját levélküldő (SMTP) motort tartalmaz,
– letölti e-mail címlistánkat,
– levelet küld az alábbi tárgysorral: „Nyertél 10 millió Uerot”. (Az éles szemű olvasók egyből kiszúrják, hogy itt Eurónak kellene szerepelnie. A kártevőkészítőknél és spamterjesztőknél gyakran tapasztalhatunk betűhibákat, elgépeléseket – helyesírásuk nem a legjobb, de gyakran a keresőmotorok megtévesztésére alkalmaznak az emberi gondolkozás számára gyorsan javítható betűcseréket.)

Jól látható, hogy a vizsgált állomány szinte mindenben megegyezik a keresett mintával, ezért feltételezhetjük, hogy a „Win32/Ez_csak_egy_példa” vírus egy újabb variánsával lehet dolgunk.

Ha bepillantunk a hamis riasztások kezelésébe, azt láthatjuk, hogy a kutatók azt is tesztelik, hogy a rosszindulatú kód keresését nagy mennyiségű tiszta és nagy mennyiségű fertőzött mintaállományon is elvégezve ne adjon hamis riasztást.

Passzív és aktív heurisztika

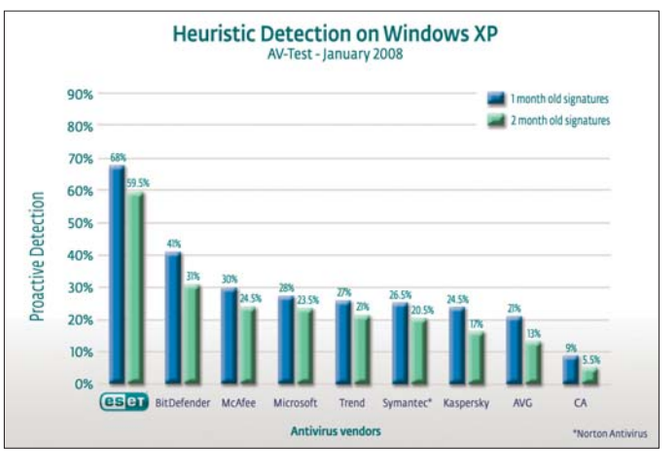
Jó példa a passzív heurisztika által generált hamis riasztásra a Microsoft DEFAULT.CAT nevű állománya, amely nem végrehajtható, és egyáltalán nem kártékony. Ha megvizsgáljuk az állományt, felfedezhetjük ugyan benne a „CD 26” kódsorozatot, amely a DOS alatt futó .COM típusú programokban a közvetlen lemezre írást jelzi, ennek ellenére ez a találat egyértelműen vakriasztás, tehát nem szabad rá figyelmeztetést kapnunk.

Az aktív heurisztikát gyártónként különböző elnevezésekkel illetik. Néhányan sandboxingnak (homokozóban való futtatás), mások virtualizációnak vagy kódemulációnak hívják. Legyen a neve bármi, az ötlet lényege, hogy egy biztonságos virtuális környezetben történik a gyanús kód lépésenkénti végrehajtásos vizsgálata, és közben ellenőrzésre kerül mindenfajta aktivitás, viselkedés. Ennek fényében kell felbecsülni a kockázatot. Nehezítheti mindezt, hogy számos állomány valós idejű futtatható tömörítővel van becsomagolva és/vagy elkódolva. A különféle tömörítésekkel a kártevők készítői pontosan a visszafejtést és az elemzést igyekeznek nehezíteni, eszközeikkel (AsPack, Morphine, PECompact, FSG, Armadillo és társaik). Ezekkel egyrészt drasztikusan megnövelhető egy adott fájl variánsainak a száma, másrészt ezzel a változatlan futtatási képesség mellett a futtatható állományok fejlécszerkezetében is sikerül olyan erőteljes változtatásokat végrehajtani, ami nagyban megnehezítheti az elemzést.

A polimorfikus – alakváltó – vírusokat is sok esetben így lehet leleplezni, hiszen fizikai kódjuk rendre egyedi és különböző.

Az aktív heurisztikus elemzés a fájlokat különböző szempontok szerinti vizsgálatoknak veti alá, jellegzetesen kártevőben előforduló is-mertetőjegyeket keresve. Olyan tulajdonságait fürkészi a vizsgált állománynak, hogy ténylegesen végez-e valamilyen adatátvitelt a portok valamelyikén, használ-e IRC-csevegőklienst, beleír-e a rendszerállományokba stb. Vagyis az ezekhez a tevékenységekhez szükséges kódrészleteknek nemcsak a meglétét vizsgálja (passzív heurisztika), hanem azt is keresi, hogy ezek a kódok tényleg az előre feltételezett akciókat hajtják-e végre.

A Windows alatti állományok szerkezete hihetetlen nagy, összetett, de szép téma, és különösen a futtatható állományok tömörítésével kapcsolatban válik még bonyolultabbá. A windowsos EXE fájlokban például a következő tevékenységek önmagukban is gyanúsak lehetnek: a kódvégrehajtás a program utolsó szekciójából indul (utólag hozzáfűzött kódrészre utalhat), szokatlan vagy hibás, esetleg többszörös PE-fejlécszerkezet (szándékos kódolás a visszafejtés ellen), üres területek a szekciók között (ez is az utólagos módosításra, illetve megtévesztő fájlstruktúrára utalhat), szokatlan KERNEL32.DLL hivatkozások (a Windows rendszer „megpatkolására” utaló jel), és még sorolhatnánk. A gyanús jelek esetén minden szimptóma kap egy súlyozott pontrendszer alapján egy valószínűségi értéket, a vizsgálat végén pedig az összegyűlt pontérték alapján minősül az adott állomány gya-

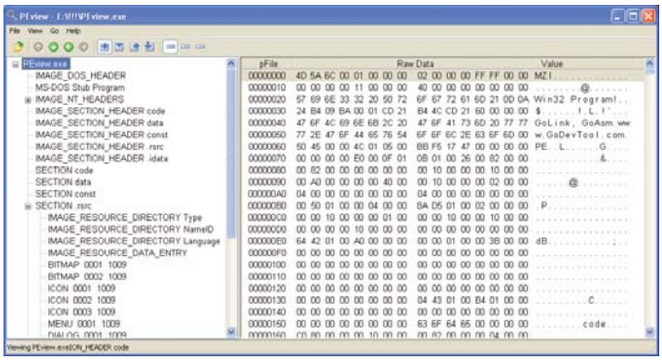


Egy független minősítő labor kísérlete arra irányult, hogy szándékosan régebbi vírus-adatállományokkal végzett vizsgálattal kiderítse, mennyire hatékonyan képesek a vírusirtók felismerni az új, ismeretlen veszélyeket. A teszteredményen jól látható, hogy a különféle víruskeresők nem egyformán alapozzák működésüket a heurisztikára, illetve heurisztikus képességeik alaposan eltérhetnek

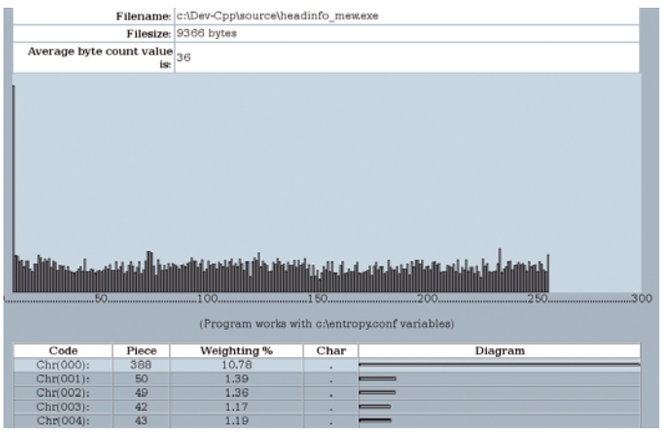
Nem számít, hogy életünkben nem láttunk még angol szettert: ha korábban volt már dolgunk ír szetterrel, fel fogjuk ismerni, ha szembetalálkozunk vele

```
6F 00 6C 00 65 00 74 00 65 00 3E 00 3E 00 3E 30 21 | o.l.e.t.e.>.>.>0!
30 09 06 05 2B 0E 03 02 1A 05 00 04 14 9F 7F 48 96 | 0...+..........H.
8E 59 F9 CD 26 21 17 79 A7 FC CA 83 98 8C 3F B5 30 | .Y..&!.y......?.0
62 06 0A 2B 06 01 04 01 82 37 0C 02 02 31 54 30 52 | b..+.....7...1T0R
1E 4C 00 7B 00 44 00 45 00 33 00 35 00 31 00 41 00 | .L.{.D.E.3.5.1.A.
```

Hiába szerepel a DEFAULT.CAT állományban a DOS-os vírusokra jellemző, direkt lemezre írásra utaló bájtsorozat, nem igazi találat: ez itt és most vakriasztás lenne



Sok mindenről árulkodik a Windows alatti futtatható állományok fejlécszerkezete



Egy egyszerű C programmal is kimutatható egy tetszőleges állományról, hogy használtak-e vajon EXE-tömörítést benne. A képen egy MEW segítségével készült állományt látunk

núsnak vagy egy bizonyos vírusvariánssal fertőzöttnek.

S hogy még ennél is tovább lehet lépni: megvannak azok a speciális módszerek, amelyekkel bárki – így sajnos a vírusíró is – tesztelni tudja egy adott futási környezetről, hogy az valós vagy virtuális-e. Ezzel pedig megnehezíthetik a visszafejtést, a kártevő viselkedésének feltérképezését.

Sok kicsi sokra megy

Ha ismeretlen állományt vizsgálunk, az alábbiakat ellenőrizhetjük, és adhatunk rá képzeletbeli pontokat.

El van-e kódolva (encrypted) a fájl? Ha igen, akkor ez gyanús, bár a másolásvédett programok is használják ezt – mindenesetre 1,5 pontot kaphat.

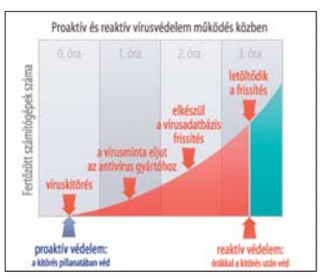
Hallgatózik-e a program egy adott kommunikációs porton? Ha igen, akkor ezért is jár 2 pont.

Beleír-e már létező fájlokba? Ha igen, akkor erre is oszthatunk 3 pontot, igaz, itt is súlyozni kell, hogy pontosan milyen állományhoz kíván hozzáférni.

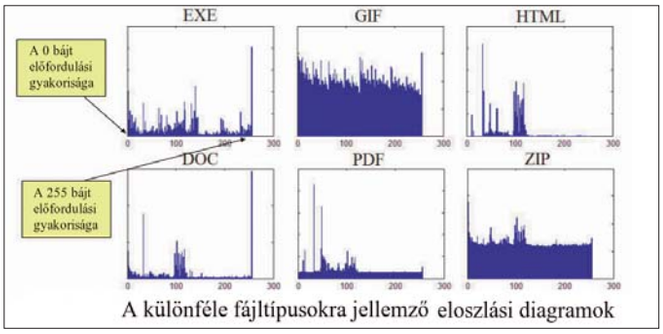
Végez-e módosítást a Registryben (Rendszerleíró adatbázis)? Ha igen, akkor kapjon egy pontot ő is. Persze itt is szükséges súlyozni: pontosan mely területekről van szó, egy ártalmatlan, saját beállításbejegyzés nem egyenlő azzal, mint amikor az automatikus programindítási részt piszkálja valami.

Hoz-e létre az adott program ideiglenes könyvtárat? (Az ideiglenes könyvtárak létrehozására a Windowsnak saját funkciója van, ezért ez egyszerűen kideríthető.) Ha igen, jár érte 5 pont, a levelezőrendszeren keresztül terjedő kártevők például éppen így járnak el.

Ha minden fontos szempont vizsgálata megtörtént, és a fenti kombinációkon felül az összes vizsgált tulajdonságra adott válasz magas pontszámot kapott, gyaníthatjuk, hogy kártékony az állomány. A lényeg, hogy mindeközben ne növekedjen a téves riasztások száma.



Ezért fontos a proaktív védelem: még a frissített adatbázis is csak a már ismert kártevők ellen képes védelmet nyújtani



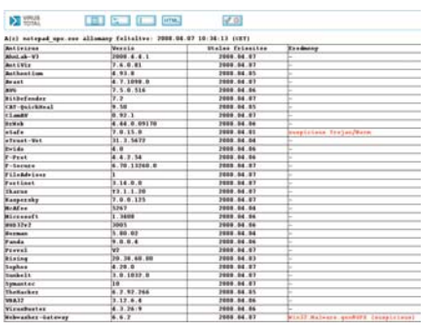
Külön tanulmányok léteznek már az entropikus – azaz bájteloszlás-vizsgálat alapján történő fájltípus-, valamint kártevő-felismerésről

Kártevők észlelése

Amikor az adatbázis-alapú felismerés fog meg egy állományt, akkor olyan konkrét nevet ír ki, mint például „Win32/Stration.YQ”, míg a heurisztika találatainál inkább a „Win32/Nuwar.Gen Worm” szóveggel találkozhatunk. Itt a „.Gen” szócska utal egyértelműen a generikus felismerésre. Más esetekben szintén a heurisztikus keresést láthatjuk. Például a „Variant of Win32/Exploit WMF Trojan” (... variáns), vagy a „Probably unknown NewHeur_PE virus” (feltételezhetően ismeretlen ... vírus) kategória is erre utal. A kereső megadhat ugyan egy konkrét kártevőnevet, de egyben jelzi, hogy ez az állomány nem egyértelműen az, viszont várható viselkedése alapján akár egy – korábban is ismert – kártevő is lehet. Minden ilyen típusú jelző, mint amilyen a „probably”, a „variant”, „possibly”, „potentially” (valószínűleg, variáns stb.) csak és kizárólag a heurisztikus motortól származik. Független tesztelők szerint a vadon élő (ITW, In The Wild, azaz valós környezetben is előforduló) vírusok közül maga a heurisztika legalább 86%-os hatékonysággal észleli a fenyegetéseket, a teljes laboratóriumi kollekcióban (In The Zoo) pedig 50-60% az a rész, amelyet ily módon észlelni képes.

A heurisztikus keresők a még ismeretlen, új típusú fenyegetésekkel szemben, illetve az ismert károkozók variánsainak felderítésében, valamint a polimorfikus, azaz alakváltó vírusok felismerésében érik el a legjobb eredményt. Egyes férgeknek több száz, néhány esetben több ezer variánsuk is lehet.

Mára a víruskeresők túlnyomó többségét felvértezték valamilyen szintű heurisztikus keresési képességgel, de a termékek e téren nyújtott teljesítményében jelentős tudásbeli különbségek észlelhetők.



Egy elrettentő példa, hogyan ne működjön egy heurisztika: a gyári Jegyzettömb Windows-alkalmazást (notepad.exe) tömörítettük UPX segítségével. A víruskereső motorok közül ketten is gyanús trójainak, illetve kártevőnek aposztrofálják

***

Kérjük kedves olvasóinkat, ha a témában kérdésük, hozzászólásuk van, juttassák el hozzánk (velemeny@pcworld.hu).

Csizmazia István,
vírusvédelmi tanácsadó
Sicontact Kft., a NOD32 antivírus magyarországi képviselete
antivirus.blog.hu

IRODALOMJEGYZÉK

Randy Abrams: Understanding and Teaching Heuristics
Pavel Cerven: Programvédelem fejlesztőknek
Columbia IDS Lab: Fileprint Analysis for Malware Detection (letölthető: www1.cs.columbia.edu/~wl318/papers/wormpaper2005.pdf)
Joanna Rutkowska: Red Pill... or how to detect VMM using (almost) one CPU instruction (web: invisiblethings.org/papers/redpill.html)
Péter Ször: Virus Research and Defense