HADMÉRNÖK

IV. Évfolyam 4. szám - 2009. december

Leitold Ferenc
fleitold@veszprog.hu

VÍRUSVÉDELEM KIVÁLASZTÁSA

Absztrakt

A vírusvédelmi rendszerek kiválasztása nem egy egyszerű feladat. Alapvetően három módszer közül választhatunk:
1. Hagyatkozunk a gyártókra, hogy mit állítanak termékeikről.
2. Saját magunk próbálkozunk meg a vírusvédelmek vizsgálatával, tesztelésével.
3. Független tesztelő szervezet elemzéseire bízzuk választásunkat.
Az első eset nem feltétlenül járható, hiszen minden gyártó a saját termékeit dicséri, hiszen azt szeretné eladni. A másik két lehetőség alapja a gyártótól független vizsgálat. Azonban ennek a kivitelezése sajnos nem kevés erőforrást igényel, így egy korrekt vizsgálatot nem feltétlenül tud egy átlagos felhasználó elvégezni. A CheckVir tesztlaborban automatikus módszerek alkalmazásával arra a feladatra vállalkozunk, hogy rendet tegyünk a vírusvédelmi rendszerek tesztelésének káoszában. Ebben a cikkben a vírusvédelmi rendszerek tesztelésének legfontosabb problémáit tekintjük át és a CheckVir tesztek legfontosabb újdonságait összegezzük.

It is not easy to select an antivirus solution. We can use three ways:
1. *We select the product using the information provided by vendors.*
2. *We can examine and test the products.*
3. *We select the product using the information provided by an independent testing organization.*

The first case is not a good solution because every vendor suggests only his products. The base of the other two cases is the examination which is independent from the vendor. On the other hand the correct examination is a very expensive task and it can not be executed by an average user. In the CheckVir laboratory we would like to execute automatic testing jobs and provide exact information related to the antivirus solutions. In this paper the most important problems are discussed and new methods of the CheckVir project are summarized.

Kulcsszavak: *vírusvédelem, tesztelés, vírus, kártevő ~ virus protection, testing, virus, malware*

1. Vírusvédelemek tesztelésének problémái

A vírusvédelmek tesztelése, vizsgálata a szoftvertesztelésnek egy teljesen különálló területévé nőtte ki magát, ez az „általános" szoftverteszteléshez képest egy teljesen más terület. Ezen a területen ugyanis számos speciális körülmény nehezíti a tesztelést.

Egyrészt talán a vírusvédelmeken kívül nincs még egy olyan szoftverterület, ahol olyan gyorsan jelennének meg az új verziók. Ennek oka, hogy – mint matematikailag is bizonyított – általános vírusvédelem nem létezhet. A vírusvédelmek gyártói így az ismert kártevőkkel foglalkoznak elsősorban, ellenük próbálnak 100%-os védelmet biztosítani. Természetesen voltak és vannak is próbálkozások, kísérletek arra vonatkozóan, hogy bizonyos módszerekkel ismeretlen kártevőket is azonosítsanak. De mitől lesz egy kártevő ismert? Az "ismertség" itt arra vonatkozik, hogy a vírusvédelem gyártója a konkrét kártevő (vagy esetleg egy csoportjuk) ismeretében készíti el a kártevő ellenszerét (felismerési és eltávolítási algoritmusát). Így viszont az újabb és újabb kártevők megjelenésével újabb és újabb verziójú védelmeket kell kibocsátani. Létezik olyan gyártó, mely átlagosan 5-6 percenként (!) ad ki egy új verziót. Nyilvánvaló, hogy ennek még csak a követése (minden verzió letöltése és archiválása) is komoly feladat.

A gyakorlatilag folyamatosan megjelenő új verziók mellett a vírusvédelmek esetén egyre inkább megjelenik a "cloud technology" fogalom, ami azt jelenti, hogy a védelmek a folyamatos internetes kapcsolat révén a gyártó által felállított központtól segítséget kérve hozzák meg döntéseiket. Ez azt jelenti, hogy a védelem működése különböző időpontokban más és más lehet.

További lényeges különbséget jelent, hogy a kártevők nagy száma miatt rengeteg tesztkörnyezetet feltételezhetnénk. Tegyük fel ugyanis, hogy létezik 10 millió kártevő. Elméletileg (de hangsúlyosan csak elméletileg) egy kártevő vagy jelen van egy tesztelési környezetben vagy nem, azaz így $2^{10 \text{ millió}}$ db tesztelési környezetet képzelhetnénk el. Ez természetesen kezelhetetlen, de még akkor is gondot okoz a mennyiségi probléma, ha csak azokat az eseteket vesszük alapul, ahol csak egyetlen kártevő van jelen (ez nyílván 10 millió a példa szerint).

Nagyon nehéz definiálni, hogy egy vírusvédelmi szoftvernek milyen elvárásoknak kell megfelelnie, mi az a működés, ami a vírusvédelem szempontjából korrekt, megbízható működésnek tekinthető. Az sem egyértelmű – és ebben is vannak különbségek a gyártók között –, hogy mely programokat tekintünk kártevőknek és melyeket nem. Sok védelem ugyanis olyan programokat is kártevőként azonosít, melyek esetén kérdéses a megítélés. Például egyes védelmek valamely szoftverhez licenszkulcsot generáló ún. feltörő (crack) programot is kártevőként azonosítanak. Hasonlóan gondot jelent – mint ahogy az Juraj Malcho (ESET) előadásában a minapi Virus Bulletin konferencián elhangzott, hogy egyre másra jelennek meg az olyan, gyakorlatilag hivatalosnak tűnő alkalmazások, melyek mögött teljesen legális cégek állnak. Ezek a cégek aztán mindent megtesznek azért, hogy a vírusvédelmek gyártóinak adatbázisából kikerüljön a termékük.

Az AMTSO-t (Anti-Malware Testing Standards Organization) 2008-ban alapították elsősorban kártevők elleni védelmi szoftvereket fejlesztők, de a munkában számos tesztelést végző szervezet mellett néhány magánszemély is részt vesz. A szervezetnek jelenleg 38 tagja van, köztük természetesen valamennyi jelentős fejlesztő. A hazai színeket a Virusbuster mellett a Checkvir tesztlabort üzemeltető Veszprog Kft. képviseli. Az elsődleges célkitűzések

között szerepel, hogy a vírus- és kártevővédelmi szoftverek tesztelésére vonatkozóan ajánlásokat, oktatási anyagokat dolgozzon ki, tesztelést segítő eszközöket bocsásson rendelkezésre , illetve az érintett felek között fórumot biztosítson.

2. Tesztelési módszerek

Vírusvédelmi rendszerek esetén a fent vázolt problémák miatt elképzelhetetlen, hogy a szoftver minden ágát teszteljük. Bizonyos szempontokat – melyek megfelelnek az AMTSO ajánlásainak is – azonban célszerű betartani.

A tesztelési eljárásnak célszerűen nyíltnak és átláthatónak kell lennie. Ez vonatkozik egyrészt a tesztelési metódus és a konkrét tesztelési folyamatok átláthatóságára is. A nyílt és átláthatóság követelménye nem feltétlenül jelenti a reprodukálhatóság biztosítását. Ennek legfőbb oka a dinamikus internethasználat, mely mind a védelmek (fenti 2. probléma), mind a kártevők részéről egyaránt jelen van. A nyíltságot és átláthatóságot azonban a dokumentáltság szintjének emelésével javíthatjuk. Ehhez célszerű a tesztelt állapotot (installált rendszert), a naplóállományokat, hálózati forgalmat elmenteni, illetve a problémás szituációkról képet és videót is rögzíteni.

Mint azt a fenti problémák esetén láthattuk, a vírusvédelmek valamennyi folyamatát nem tesztelhetjük. Bizonyos szempontokat vizsgálhatunk “csak”, célszerűen azokat, melyek a legfontosabbak a felhasználó szempontjából:

a. **Hatékonyság**, azaz a védelem milyen biztonsági szintet tud biztosítani (kártevők korrekt ismerete – felismerés, eltávolítás). Ez a szempont kifejezetten az antivírus képességeire vonatkozik: Milyen kártevők elleni védelemre készítették fel a védelmet? Milyen területeket képes vizsgálni? Ide tartoznak a különböző diszkterületek vizsgálati lehetőségei, fájlformátumok, tömörítők és egyéb fájlok tárolására használható formátumok vizsgálati lehetősége. Ebbe a csoportba tartozik természetesen a téves riasztások tesztelése is.
b. **Megbízhatóság** (stabil, hibamentes működés). Képes-e a védelem úgy működni, hogy folyamatosan ellássa feladatát. A stabilitás vizsgálatánál a tesztelés során hangsúlyozottan kell törekedni a reprodukálhatóságra.
c. **Teljesítmény** (sebesség, számítógép lassítása, bootolás időigénye). Általában a teljesítménnyel szembeni felhasználói elvárás akkor jelenik meg, ha védelem képes hibamentesen megfelelő biztonsági szintet nyújtani. Nem véletlen, hogy az AMTSO alapelvei között szerepel az, hogy a hatékonyságot és a teljesítményt együtt, kiegyensúlyozottan kell tesztelni. Ennek oka az, hogy ha például egy védelem nem vizsgál meg bizonyos dolgokat, nyílván gyorsabban képes végrehajtani feladatát. De hiába gyorsabb, ha nem véd bizonyos problémákkal szemben.

Vírusvédelmi rendszerek tesztelése esetén – a kártevők ismeretének vizsgálata során – kulcskérdés, hogy milyen és mennyi mintát használunk. A minőségre vonatkozóan az AMTSO május elején fogadott el egy dokumentumot, mely a használt minták validálásával foglalkozik. E szerint a tesztelés során olyan kártevő mintákat kell alkalmazni, melyek egyrészt működőképes kódot tartalmaznak, illetve valóban képesek a kártékony tevékenységet végrehajtani. Ennek biztosítása/bizonyítása nem egyszerű feladat. Mint az a fenti problémák között is szerepelt, nem egyértelmű, hogy mi tekinthető kártevőnek és mi nem. Hasonlóképpen számos védelmi rendszer olyan állományt is kártevőként azonosít,

amelyet semmilyen környezetben nem lehet futóképessé tenni, azaz végrehajtani, például tönkretett futtatható állományok. A tesztelés szempontjából a mennyiség is nagyon fontos kérdés, azonban a használt minták mennyiségét (mint a tesztelési metódus egy lényeges elemét) a tesztelés céljának kell meghatároznia. A tesztelés kiterjedhet egyrészt a létező kártevők összességére. Ebben az esetben nyílván milliós nagyságrendű készlettel lehetne statisztikailag helyes következtetésre jutni (itt viszont a korrekt validálás okozna nagy problémát). A szűkítésre két lehetőség kínálkozik: egyrészt vizsgálhatunk bizonyos típusú kártevőket, amivel egy adott területre fókuszálva vonhatunk le következtetéseket a vírusvédelmek tudására vonatkozóan. Másrészt figyelembe vehetjük az elterjedtségi adatokat is. A létező kártevők mennyiségéhez képest az elterjedt kártevők száma több nagyságrenddel kevesebb. Ezen kártevők körében már statisztikailag is helyes következtetést vonhatunk le néhány ezres mintakészlettel. Felmerülhet még továbbá olyan vizsgálatnak a lehetősége is, ahol nem az a cél, hogy a felismerési képesség alapján rangsoroljuk a védelmeket, hanem hogy egy-egy konkrét biztonsági probléma kezelésére következtessünk, azaz például válaszoljunk olyan kérdésekre, amelyek arra terjednek ki, hogy valamely felmerült probléma jelent-e biztonsági kockázatot. Ilyen esetben néhány minta is elegendő lehet a vizsgálathoz.

A védelmi rendszerek számos tevékenységgel azonosíthatnak egy kártevőt. Amennyiben a felhasználó indítja el interaktív módon a vizsgálatot, on-demand eljárásról beszélünk. Az azonosítást a folyamatosan figyelő ún. on-access védelem is megteheti. Ez utóbbi felfedezheti a kártevőket a fájlhoz történő hozzáféréskor, de előfordulhat, hogy bizonyos kártevőket a védelem csak akkor képes azonosítani és persze blokkolni, ha azokat elindítjuk. Ez utóbbi – proaktívnak hívott – működés egyre gyakoribb a vírusvédelmek esetén. A tesztelési folyamat során tehát célszerű proaktív vizsgálatot végezni. A módszer természetéből adódóan ezt viszont csak úgy szabad elvégezni, ha az operációs rendszer bootokását követően minden egyes alkalommal csak egyetlen kártevőpéldánnyal tesztelünk. Az ily módon történő tesztelés ezért rendkívül időigényes. Gondoljunk csak bele: Ha egy számítógépet használunk egy védelmi rendszerhez és egyetlen kártevő tesztelése 5 percet vesz igénybe – amibe beletartozik az eredeti, biztosan vírusmentes operációs rendszer visszaállítása is –, akkor egymillió minta teszteléséhez mintegy 9 és fél évre lenne szükség. Proaktív tesztet mindezek alapján tehát nagy mennyiségű mintán belátható időn belül nem végezhetünk. További nehézséget jelent, hogy egyes kártevők az azonosításukhoz szükséges tevékenységeket esetenként internetes kapcsolat révén biztosítják. Ebben az esetben viszont bonyolult feladat annak biztosítása, hogy a kártevő ne kerülhessen ki a tesztkörnyezetből.

3. Valós idejű tesztelés a CheckVir tesztlaborban

A CheckVir tesztlaborban az NKTH Baross Gábor Programjának (KD_INTEG_07-FB124076), illetve a Gazdaságfejlesztési Operatív Program (GOP-2.1.1-09/A-2009-1298) támogatásával a vírusvédelmi rendszerek folyamatos, valós idejű (real-time) tesztelésére alkalmas rendszert alakítunk ki. A tervezett rendszer alkalmas arra, hogy a vizsgált rendszerek tesztelését naponta néhány alkalommal automatikusan elvégezze. A rendszer tehát naponta (többször is) megpróbálja a védelmi rendszereket (és persze az operációs rendszert is) frissíteni, majd ha ténylegesen történt frissítés, akkor előre definiált tesztek sorát hajtja végre. Az eredmények pedig a tesztelést követően azonnal megjelennek az interneten. Nyilvánvaló, hogy a tesztek végrehajtásához idő kell, néhány percenként megjelenő frissítések esetén a rendszer nem lehet képes valamennyi verzió átfogó tesztjére. A termékenként naponta 1-2 teszt végrehajtása esetén azonban így is sokkal inkább a valós tudáshoz és teljesítményhez közelít a vizsgálatok eredménye.

A CheckVir projekt célja, hogy aktuális információkat biztosítson a vírusvédelmi eljárásokat tartalmazó megoldások hatékonyságára, megbízhatóságára és teljesítményére vonatkozóan. Tekintettel arra, hogy a teszteredmények a vizsgálatot követően azonnal publikálásra kerülnek, így a tesztelés nem a múltra, hanem a jelenre vonatkozik.

3.1. Technikai háttér

A CheckVir tesztlaborban jelenleg mintegy 15 Intel P4-es számítógép végzi a kártevők szaporítását és a teszteléseket. A GOP pályázat támogatásával most olyan cluster rendszer alakulhat ki, amely további 24 db négymagos Intel számítógépet jelent. Ez pedig a jelenlegi számítási kapacitást megsokszorozza.

A tesztfeladatok egy keretrendszerben kerülnek végrehajtásra. A rendszert vezérlő számítógép folyamatosan generálja a végrehajtandó feladatokat, amit aztán a kliens számítógépek elvégeznek, majd egy újabb feladatot keresnek maguknak. A keretrendszer olyan feladatokat képes végrehajtani, amely például egy termék frissítésére, vagy akár egy tesztelési részfeladat végrehajtására vonatkozik. Amennyiben egy termék frissítése sikeresen megtörtént, a teljes környezet archiválásra kerül és automatikusan létrejönnek az új verzióra vonatkozó tesztfeladatok. A rendszer működése során a fentieken túlmenően az operációs rendszer frissítését végző feladatok is végrehajtásra kerülnek.

A teszteredmények a kiértékelést követően egy helyi adatbázisba kerülnek, ahonnan percenkénti szinkronizálással másolódnak a web kiszolgáló adatbázisába. A webkiszolgáló pedig dinamikusan jeleníti meg az eredményeket.

A valós idejű tesztelés folyamatos működése így természetesen a webszerver és a tesztlabor közötti kapcsolattól is függ. Ennek bizonytalansági kockázatát alternatív internet-szolgátatókkal próbáljuk csökkenteni.

3.2. Internet-kapcsolat kezelése

A rendszer megvalósítása szempontjából egy lényegesen külön részfeladat a keretrendszerben lévő, a frissítéseket és magukat a tesztfeladatokat is végrehajtó kliens számítógépek internet-kapcsolatának a kezelése. A projekt szempontjából két probléma merült fel, melyek együttesen szükségessé tették egy úgynevezett intelligens proxy szerver kialakítását:

- Mind a kártevők, mind pedig a védelmek esetén megjelent a már említett “in-the-cloud” technológia használata. Kártevők esetén ez azt jelenti, hogy a kártékony kód egy részét az internetről tölti le, a védelmek esetén pedig folyamatos kapcsolatot jelent a gyártó rendszerével.
- A kártevők jelentős részének azonosítása ma már csak proaktív védelemmel oldható meg. Ez azt jelenti, hogy a védelem csak akkor képes azonosítani és blokkolni a kártevőt, ha a felhasználó azt elindítja.

A két problémával külön-külön a projekt során tervezett rendszer képes lenne megbirkózni, azonban a két probléma együttes kezelése már nehézkes, hiszen az alábbiakat kell biztosítani a tesztkörnyezetben:

- A védelmi rendszereknek legyen lehetősége a gyártó rendszeréhez korlátozás nélkül csatlakozni.

- Egyazon kártevő különböző időpontban (különböző védelmi rendszerekkel) történő vizsgálatakor biztosítani kell, hogy az internetről ugyanazt a kódot töltse le. Ellenkező esetben az eredmények összehasonlítása nem lehet megalapozott. A letöltött kódot persze archiválni is célszerű.
- Biztosítani kell, hogy a kártevők ne legyenek képesek arra, hogy az internetkapcsolaton keresztül más számítógépeket megfertőzzenek.

Mindezek alapján olyan internetkapcsolatra van szükség, ami a védelmi rendszerek gyártói felé engedik a védelmi rendszer forgalmát, a kártevők esetén pedig gyakorlatilag egy virtuális internetet biztosít.

A tesztkörnyezet kialakítása során kézenfekvőnek tűnik az a megoldás, hogy használjunk valamilyen proxy szervert. Egy proxy képes tárolni a „külső" hálózat felől érkező adatokat és azt továbbítani egy „belső" tesztkörnyezetben futó számítógép felé. Egy proxy fontos eleme lehet a tesztkörnyezetnek, azonban önmagában semmiképpen sem tudja a feladatot ellátni. Használata során különböző problémák jelentkezhetnek. Ezek közül a legfontosabbak:

- Nem minden hálózati protokoll támogatja a proxy használatát, bár a leggyakrabban használtak (pl.: http, ftp) igen.
- Egy proxy szerver a cache-ét folyamatosan frissíti, így egy „korábbi állapot" megőrzése nem automatikus.
- Egy http proxy bizonyos fejlécek segítségével utasítható arra, hogy ne tárolja le a cache-ében az adott tartalmat.
- Egy futó alkalmazás képes érzékelni, hogy proxy-n keresztül kapcsolódik a külső hálózat felé, azáltal másképp működhet, mint azt „normális esetben" proxy nélkül tenné.

Az előzőek figyelembe vételével a tesztkörnyezetben tehát nem támaszkodhatunk kizárólag egy proxy szerverre. Fontos feladat a tényleges hálózati forgalom rögzítése, ennek analizálása. A proxy szervert ezen túlmenően fel kell vértezni azzal a képességgel, hogy képes legyen adott szituációhoz igazodni. A megfigyelésnek vannak olyan szakaszai, amikor észlelni és rögzíteni kell a hálózati forgalmat. Amennyiben ez rendelkezésre áll, akkor csakis ezt felhasználva oldható meg az, hogy a proxy szervert „intelligenciával" lássuk el.

A tesztkörnyezettel szemben támasztott legfontosabb működési követelmények:

- Legyen lehetőség előre és dinamikusan is meghatározni, hogy a tesztkörnyezetből mely gépek érhetik el a külső hálózatot, illetve ezen belül mely számítógépeket.
- Biztosítani kell a belső tesztgépek számára a névfeloldást és az IP címek automatikus hozzárendelését (DHCP).
- Naplóznia kell a bekövetkező eseményeket.
- A tesztkörnyezetnek a „lehetőségekhez mérten" zártnak kell lennie, tehát biztosítani kell azt, hogy „nem kívánt módon" számítógépek ne fertőződjenek meg.

Az előző kritériumok alapján célszerű olyan operációs rendszert használni a környezet vezérlésére, ahol minimálisra csökkenthető a vírusveszélyből és a támadásokból eredő veszély. Linux operációs rendszerben az előző feladatok megoldhatók, a Linux tűzfala segítségével finoman szabályozható a külső elérés, illetve az átmenő forgalom.

3.3. Tesztfeladatok

A végrehajtandó tesztfeladatok a vírusvédelmi rendszerek három lényeges tulajdonságát mérik: hatékonyság, megbízhatóság, teljesítmény. A hatékonyság és a megbízhatóság a védelmi rendszerek estén bizonyos mértékig összetartozó fogalmak, hiszen egy felhasználó nyilvánvalóan a hatékony védelmi rendszert tekinti megbízhatónak.

A hatékonyság problémakörébe olyan vizsgálatok tartoznak, amelyek a védelmi rendszer azon képességét mérik, amelyek arra vonatkoznak, hogy milyen problémák esetén képesek védelmet nyújtani. A kártevők elleni védelemre szorítva a kérdéskört: mely kártevőket, mely területeken képes megakadályozni tevékenységében. Az viszont itt másodlagos, hogy ezt milyen módszerrel teszi. Továbbá ide tartozik természetesen annak a vizsgálata is, hogy a kártevőktől mentes, tiszta állományok közül melyeket és milyen arányban minősít a védelem kártékonynak.

A megbízhatóság ellenőrzése a védelmi rendszer korrekt, hibamentes működésének vizsgálatát jelenti. Ebben a kérdéskörben olyan teszteket képzelhetünk el, amelyek a védelmi rendszert megpróbálják akár az átlagosnál nagyobb feladat elé állítani, mintegy terheléses vizsgálatot végezve. A korrektséghez hozzátartozik a védelmi rendszer konzekvens viselkedése is például a különböző védelmi módszerek (on-demand, on-access) esetén.

A teljesítmény a felhasználó szempontjából leginkább a különböző védelmi módszerek sebességét jelenti: milyen gyorsan képesek végezni egy adott feladattal, mennyire lassítják a számítógép működését.

A CheckVir projektben az alábbi tesztfeladatokat tervezzük automatikusan végrehajtani:

Kártevők ismeretének a vizsgálata (felismerés): Alapfeltétel, hogy a tesztelt megoldás képes legyen valamilyen naplóállomány készíteni a vizsgálatról. A felhasználó által indított on-demand ellenőrzés során a kártevők találatainak az összesítése történik. A folyamatosan figyelő on-access védelmek esetén a vizsgálat kicsit bonyolultabb. Itt nemcsak a naplóállományokat elemezzük, hanem megvizsgáljuk, hogy a kártevők másolása esetén mi történik a forrással és mi lesz a "célban". Természetesen itt olyan beállításokat kell alkalmaznunk, ami arra utasítja a védelmet, hogy kártevő esetén valamilyen módon gátolja meg a másolást. Tekintettel arra, hogy a védelmi rendszerek jelentős része ma már proaktív védelemmel is rendelkezik, az olyan kártevők esetén, amiket a másolás során nem azonosított a védelem, szükség van a proaktív védelem vizsgálatára. Ehhez a kártevőket el kell indítani a tesztkörnyezetben és vizsgálni a védelmet, hogy meggátolja-e a kártevő működését. Ehhez azonban minden egyes kártevő vizsgálatát megelőzően kártevőmentes környezetre van szükség.

Kártevők ismeretének a vizsgálata (eltávolítás - helyreállítás): A kártevők kódjának eltávolításának, az eredeti kártevőmentes állapot visszaállításának vizsgálata mind az on-demand, mind az on-access, illetve a proaktív esetben is elvégezhető. Mindegyik esetben a vizsgálat arra terjed ki, hogy a védelem működése előtti állapot hogyan változott meg. A változásokat természetesen a védelmi rendszer naplójával is összevetjük.

Téves (false positive) riasztások vizsgálata: Az on-demand ellenőrzések esetén a védelmi rendszerek által generált naplóállományok vizsgálata alapján, on-access esetben pedig a

naplóállományok, illetve a forrás és a célterületek alapján történik azon fertőzésmentes állományok körének a meghatározása, amelyek esetén a védelem – tévesen – kártevőt jelez.

Sebesség ellenőrzése kártevőmentes környezetben: A sebesség ellenőrzését alapvetően kártevőmentes környezetben célszerű elvégezni. Kártevő(k) találata esetén ugyanis már másodlagossá válik a sebesség, a felhasználók számára sokkal fontosabb lesz a biztonságos helyreállítás. Másrészt pedig a sebesség nagyban függ a találat esetén elvégzendő akciótól (törlés, eltávolítás, karanténba helyezés, …). Kártevőmentes környezetben a sebességet nagyban befolyásolja a hardver és szoftver környezet is. On-demand esetben az indítás és befejezés közti időtartam jelenti az ellenőrzés idejét. On-access esetben ez egy kicsit bonyolultabb. Ekkor ugyanis, ha például másolással történik a vizsgálat, akkor – a korrekt összehasonlíthatóság érdekében –, az indítás és befejezés között eltelt időtartamot csökkentenünk kell a védelem nélküli rendszerben ugyanezen feladat elvégzéséhez szükséges időtartammal. Így kapjuk meg ugyanis tisztán a védelem időszükségletét. A sebességellenőrzések mindegyikét – a hiba mértékének csökkentése érdekében - legalább 20-szor végezzük el. A vizsgálat eredménye ezen időtartamok statisztikai jellemzői (minimum, maximum, átlag, szórás).

“Konténer” teszt: A számítógép használata során átlagos körülmények között is előfordulhat, hogy egy felhasználó olyan állomány(oka)t hoz létre, amely fájl(oka)t tartalmaz. Ez történhet a felhasználó intereaktívitásával, vagy akár – mintegy észrevétlenül – teljesen automatikusan. A felhasználó közreműködése lehet például, ha egy ZIP állományt hoz létre vagy egy dokumentumba beleágyaz egy másik állományt. Az automatikus esetben például a felhasználó egy e-mail üzenetet tölt le és azt az e-mail kliense tárolja el. Az ilyen típusú állományok esetén a védelmi rendszernek ismernie kell ezen állományok felépítését. Az ilyen vizsgálatok elvégzése azért nagyon fontos, mert ha például egy védelem azt az üzenetet adja – miután egy számítógépet telesen átvizsgált -, hogy az vírusmentes, akkor legyünk tisztában ezen üzenetnek a korlátaival. A vizsgálat során olyan állományt hasnálunk, melyet valamennyi, a tesztelésben résztvevő védelem ismer (például az EICAR tesztfájlt: www.eicar.com) és ezt az állományt csomagoljuk be különböző “konténerekbe”. A vizsgálat során azt teszteljük, hogy képes-e a védelem az adott „konténerben” megtalálni a kártevőt illetve, hogy ezt konténerbe ágyazott konténerek sorában milyen mélységig képes megtenni. Ezen túlmenően speciális körülmények között is vizsgáljuk a védelem képességeit: hosszú fájlnevek használata, jelszóval védett konténerek, …

3.4. Minták

A kártevők elleni védelmi rendszerek tesztelésének egy sarkalatos kérése mindig a használt kártevőminták körében rejlik. Manapság már több, mint 20 millió kártevő létezik a világon. Ezen kártevők nagyon nagy hányada nem jelent potenciális veszélyt egy átlagos felhasználó számára. Nagyon nehéz azonban meghatározni, hogy melyek azok a kártevők, amelyek valóban veszélyeztetik az átlagos felhasználó számítógépét. És persze az is igaz, hogy ezek – mivel különböző mértékben vannak elterjedve -, különböző arányban számíthatnánk a teszteredményekbe. Az elterjedtségi adatok azonban folyamatosan változnak, így ilyen súlyozás gyakorlatilag kivitelezhetetlen. Nem beszélve arról, hogy az elterjedtségi adatok gyártóról gyártóra változnak.

A CheckVir tesztlaborban a valósidejű tesztekhez a kártevők kétfajta mintakészletét használjuk. Az első csomagban a védelmi rendszerek gyártóitól származó elterjedtségi adatokat dolgozzuk fel a Wildlist (www.wildlist.org) listáival együtt. Ebben a készletben az

elmúlt három hónapban elterjedtként kezelt kártevőket teszteljük. A második csoportba a mindenkori utolsó fél évben a tesztlaborba érkezett minták kerülnek. Mindkét csoportban a kártevőket típus szerint osztályozzuk. Ez azonban nem minden esetben esik egybe néhány gyártó osztályozásával, itt is vannak ugyanis különbségek. A kártevőminták esetén alapvető feladat továbbá a minták validálása. Ez két dolog igazolását jelenti: Egyrészt igazolnunk kell a kártevő kódjának működőképességét, másrészt pedig azt, hogy valóban képes a káros tevékenység végrehajtására. Ehhez olyan automatikus rendszert használunk, amely mind natív, mind pedig virtuális környezetekben képes az említett feladatok elvégzésére.

Összegzés

Mint azt láthattuk, a vírusvédelmi rendszerek tesztelése messze nem triviális feladat. Általában is igaz, mindenfajta tesztelésre, hogy az nem azt igazolja, hogy egy szoftver hibamentes, funkcionálisan megfelelően működik, csupán azt képes bizonyítani (ha hiba adódott), hogy a szoftver nem tökéletes. Ez fokozottan igaz a vírusvédelmekre is, hiszen gyakorlatilag megoldhatatlan, hogy teljes, átfogó tesztet végezzünk egy-egy antivírus verzióval. Mindezek ellenére a vírusvédelmi rendszereket oly módon kell (és lehet is) vizsgálni, amivel kimutathatóak azok a problémák, melyek az átlagos felhasználónál is előfordulhatnak. A CheckVir projekt célja egy folyamatosan működő, valós időben történő, automatikus tesztelés biztosítása, mely lehetőséget ad arra, hogy a felhasználók (a laikustól a rendszergazdáig) megfelelő információkkal rendelkezzenek a vírusvédelem kiválasztásához.