A hálózati vírusvédelem és a szolgáltatásmegtagadásos támadások elleni védekezés problémái és kapcsolatai

Bencsáth Boldizsár
boldizsar.bencsath@crysys.hit.bme.hu
BME, Híradástechnikai Tanszék

Absztrakt

Előadásomat a problémakör és a javasolt, ismert megoldások rövid bemutatásával kívánom kezdeni. Be kívánom mutatni, hogy a hálózati vírusvédelmet jelenleg milyen eszközökkel szokásos és érdemes elvégezni: kliens oldali védelem, levelező szerver / relay szerver védelme kiemelve a nyílt forráskódú megoldásokat (pl. linux, amavis, mailscanner, clamav, unix víruskeresők és „mail gateway" védelmi szoftverek), tartalomszűrési lehetőségek (web forgalom szűrése), fájlhozzáférés-védelem vírusvédelemmel kombinálva (pl. RSBAC malware scan). Hasonló módon röviden ismertetni kívánom a DDoS (elosztott szolgáltatás-megtagadásos támadás) támadások különböző fajtáit (protokoll hiba, hálózati túltöltés, szerver-leterhelés) és az azok elleni védekezési alapvető védekezési módszereket (hibajavítás, tűzfalas védelem, anomália alapú szűrés (SYN védelem, stb.), forgalomanalízis alapú lehetőségek). A DDoS támadások kapcsán meg kívánom említeni az aktuális támadások főbb fajtáit, gyakorlatát (spam elleni védekező szolgáltatások támadása, Ebay, SCO stb. támadások, zombie hálózatok).
A tömör bevezető után be kívánom mutatni a vírusvédelmi rendszerek főbb problémáit a DDoS támadások szemszögéből: A vírusvédelem teljesítményigényét, az elárasztás lehetőségeit, a vírus-visszajelzések által okozható és okozott károkat. A problémák vázolása után a védekezési lehetőségek kiterjesztését kívánom bemutatni az általunk vizsgált egyik megoldási lehetőséget: A vírusvédelmi rendszer kombinálását a forgalmi analízis technikájával a DoS támadások ellen. A megoldás a beérkező levelek egyszerű statisztikai analízisével teremti meg annak lehetőségét, hogy a vírusvédelmi szerver ellen a DDoS támadások lehetőségét lecsökkentsük. A módszer felhasználható még az ismeretlen vírusok elleni védekezésben is a korai járvány-detekció érdekében. A javasolt megoldás alátámasztásaként ismertetem jelenlegi mintaimplementációnk felépítését.

Bevezetés

Az Internet biztonságát számos különböző dolog veszélyezteti. Az elmúlt tíz év hálózatbiztonsági fejlesztései ezek közül gyakorlatilag egyetlen veszélyt sem tudtak megszüntetni. Ez főként annak köszönhető, hogy az alapvető infrastrukturális megoldások nem változtak meg, és a védelem fejlődésével a támadások is fejlődtek.
Az Internet egyik legjobban elterjedt, és épp ezért legjobban ismert veszélyének az internetes férgek és vírusok tekinthetők. Cikkemben megpróbálom bemutatni a problémakör jelenlegi helyzetét és kiemelni néhány érdekes megoldást, így saját fejlesztésünket is.

A hálózati vírusvédelem eszközei

A következőkben rövid betekintést kívánok nyújtani a Linux vírusvédelmi lehetőségeibe megemlítve a határterületeket. Nem célom egy összefoglalás írása és mélyreható analízis, a fejezetet azoknak szánom, akik nem ismerősek a területen és némi kezdeti tudásra kívánnak szert tenni a mélyebb vizsgálatok előtt.

A hálózati vírusvédelemnek ma már számos különböző, többségében jól ismert eszköze van. Az eszközök kategorizálása előtt azonban fontos a vírus fogalom pontosítása.

A számítógépes vírus fogalma

A számítógépes vírus fogalma az utóbbi években folyamatos változásokon ment keresztül. Kezdetben egyértelműen olyan azonosítható, önmagában működésképtelen programkód-részletet értettünk rajta, amely önreprodukcióra képes, erőforrásokat emészt fel és esetenként kárt okoz.
Ezt a definíciót számos tényező módosította:

- Nagy számban jelentek meg internetes, főként e-mail alapú vírusok.
- Megjelentek a nem futtatható kód hibáit kihasználó programok.
- Olyan terjedő programok jelentek meg, amelyek önállóan is futtatható programok, hordozóközegük az e-mail
- Olyan programok jelentek meg, amelyek a levelező rendszer vagy más szoftverek hibáit aknázzák ki terjedésük során, így a terjedés hatékonyabb lehet
- Olyan programok jelentek meg, amelyeknek terjedése a felhasználó megtévesztésén alapul. (Ezek többségét régebbe az ún. trójai kifejezéssel nevezték meg)
- A rosszindulatú programokon túl megjelentek a vírusként terjedő megtévesztő információk, az ún. hoax-ok, amelyek a felhasználók segítségével tudnak terjedni.
- Elszaporodtak a hálózati hibákat kiaknázva terjedő programok, amelyeket ma is főként *worm*, azaz a féreg szóval neveznek.
- Olyan programok és funkciók jelentek meg, amelyek célja a felhasználókról szóló információgyűjtés (reklám céljára: ki mit tölt, használ stb., egyéb célra: billentyűzetlenyomást rögzítő kémprogramok)
- Megjelentek olyan programok, amelyek a számítógépen kiskaput hoznak létre, hogy azon keresztül a gép bizonyos funkciói távolról is elérhetőek legyenek, rosszindulatú fél számára
- Létrejöttek a hálózati szolgáltatásmegtagadásokat lehetővé tevő osztott, központilag koordinált rendszerek, és ezek kliensei
- Az on-line hirdetési piac megjelenésével létrejöttek azok a programok, amelyek a számítógép tulajdonosait kért vagy kéretlen reklámokkal árasztják el, pl. a képernyőn rendszeresen megjelenő ablakok segítségével.
- Tömegprobléma lett az internetes reklámlevelek küldése, és az ezek elleni harc miatt a küldők olyan kódokat hoznak létre, amelyek segítik a hirdetőt abban, hogy

kéretlen küldeményeit a szűrőket megkerülve juttathassa nagy számban az emberekhez.

A vírusok körében is megfigyelhető *konvergencia* során programkódok jelentek meg amelyek a fentiekben ismertetett számos funkció közül jónéhányat egyszerre valósítanak meg. Ezeket a programokat nehéz minden esetben vírusnak nevezni, hiszen ezek az eredeti definícióinkat már nem mindig fedik le.
A kialakult gyakorlat szerint az ilyen kódokat inkább „malicious software" azaz „malware" magyarul *ártalmas kód* néven nevezhetjük.

Könnyű lenne azonban azt mondani, hogy ezentúl a vírusok, férgek, kémszoftverek (spyware), kiskapuk, trójaiak, feltörések, stb. helyett használjuk az ártalmas kód fogalmat, de még ezzel is gondjaink adódhatnak, amelyet az alábbi példa szemléltet.

> *A Gator cég GAIN szoftvere egyike azon programoknak, amelyek a felhasználó szándékával nem mindig megegyezően, ám gombnyomásával elfogadva kerülnek a felhasználó gépére. A szoftver kereskedelmi hirdetések terjesztését segíti a szerzőnek, ám a felhasználók többsége nem vágyik több, személyre szabottabb hirdetésre, elég nekik bőven az, ami érkezik. A Gator szoftverét több víruskereső illetve speciálisan spyware (adware) kereső program eltávolítja. A Gator azonban nemrég bejelentette, hogy nem ért egyet azzal, hogy szoftverét kémszoftvernek nevezték és mint ártalmas kódot távolítják el egyes gyártók szoftverei és jogi ellenlépésekkel fenyegetőzik.*

A példából az látható jól, hogy még az elnevezésekkel is vigyázni kell, amikor egyes ártalmasnak tekinthető kódokról beszélünk. A továbbiakban tehát az ártalmas kódokra vírusként fogok utalni, ahol valamelyik speciális tulajdonságú részhalmazukról beszélek, úgy azt jelezni fogom.

A vírusvédelem helye

A vírusok elleni védekezés alapjaiban véve a klienseknél kezdődött, főként víruskereső és –irtó szoftverek megjelenésével. A keresés kezdetben esetleges volta később automatizált, folyamatos vírusvédelemmé alakult a kliens oldalon. A vírusok és még inkább férgek által kihasznált programhibák ellen a védekezés a szoftverek és a rendszer frissen tartása és annak megfelelő ellenőrzése, ami túlmutat a vírusvédelmen. Ha a frissítés nem valósítható meg, úgy a tűzfalas védekezés jelent további védelmi lehetőséget az ilyen támadások ellen.

A kliens oldali védelem mai napig legnagyobb problémája az, hogy nincsen, vagy rosszul működik: A védelmi szoftverek pénzbe kerülnek, amit még mindig nem mindenki fizet meg, vagy ha meg is fizet, akkor sem tudja megfelelő frissességben tartani a védelmi rendszerét, amely így hatástalan lesz.

A vírusok jelenlegi legnagyobb veszélye nem az a károkozás, amelyet az egyedi kliensekre jelentenek. Nem az jelenti ma a legnagyobb veszélyt, hogy egy vírus le fogja törölni a merevlemez teljes tartalmát. A veszély jelenleg inkább abban áll, hogy a cég titkos információi napvilágra jutnak, a cég napokig nem tud működni, hogy a cég vagy akár az ország számítógépes hálózata lelassul, működésképtelen lesz, vagy nehezíti a munkát, csökkenti a teljesítményt.

A veszély tehát jelentős, érdemes megerősíteni a védelmet. A megerősítés, és egyszerűsítés elve mentén haladva jelentek meg a központosított megoldások: Egy szervezet minden munkaállomására víruskeresőt telepítünk, amelyek a cég központi szerverével állnak kapcsolatban. A központi szerver ismerheti a kliensek állapotát, és ellenőrzött módon hajthatja végre a kliensek vírusadatbázisainak frissítését.

Természetesen a klienseken túl a fájlszerverek is védelemre szorulnak, így a víruskereső programok többsége megjelent a fájl szerverekre szánt különleges verzióban is.

A megoldás használata során felmerülhet a kérdés: Ha az internetes levelezés lett a vírusok bekerülésének legfontosabb közege, miért nem kapcsolunk az internetes levelezésre is vírusvédelmi szoftvert. Nos a kérdést tett követte és még ma is egyre-másra jelennek meg az e-mail szervereket védő szoftverek.

Sokak tévesen úgy gondolják ma is, hogy ha a vírusok e-maileken jönnek, és az e-mail rendszert megfelelő szoftver védi, akkor a kliens oldali védekezés elhanyagolható, felesleges. Számos példa mutatja azonban, hogy ez koránt sincs így. A vírusok gyorsabban terjedhetnek el, mint a amilyen gyors a vírusvédelmi szoftverek frissítése. Ilyen esetben elképzelhető, hogy egy vírus védelmi rendszerünk frissítése előtt eljut valamely munkaállomásunkra. Amennyiben a védelem a munkaállomásra is kiterjed és megfelelően van beállítva, úgy a megfelelő frissítés letöltése után a fertőzött munkaállomás gyorsan felismerhető, illetve a fertőzés akár a szoftver automatikus működésével is megszüntethető. Ha azonban a bejutott vírus korlátlanul viselkedhet belső hálózatunkon, úgy könnyen átterjedhet a gond a többi munkaállomásra is, és innentől a helyreállás hosszú időt és sok munkát igényel.

Elmondhatjuk tehát, hogy a vírusok elleni védekezés egy cég belső hálózatán több helyen szükséges feladat. A következő részben azt vizsgálom meg, hogy ez a feladat hogyan oldható meg a hálózat e-mail kiszolgálóján / e-mail „tűzfalán”, ha azt nyílt forráskódú, lehetőleg ingyenes szoftverekkel kívánjuk üzemeltetni.

Linux e-mail átjáró védelem

A linux átjáró védelmében számos különböző funkciót megvalósító komponens vehet részt. Egyes szoftverekben, főként a víruskeresőt gyártó cégek szoftvermegoldásaiban ezek a komponesnek egyetlen szoftverbe kerülnek integrálásra, azonban a szokásos UNIX „KISS” (keep it simple, stupid) hozzáállásnak megfelelően célszerű lehet az egyes komponensek jól definiált szétválasztásra.

A levél kézbesítési folyamata alapvetően a következő szakaszokra bomlik:

- A levél átvétele a szomszédos SMTP szervertől (MTA- Mail Transport Agent funkció)
- A levél átvizsgálása vírusok és/vagy spam szempontjából
- A levél kézbesítése lokálisan, vagy továbbküldése

A levél helyi kézbesítésekor természetesen gyakorta lehetőség van még további szűrők használatára is, ilyen megoldás lehet, amikor a levél helyi kézbesítését a *procmail* szoftver végzi, és az egyes felhasználók egyedi módon kiszűrik a spam vagy vírus gyanús küldeményeket.

A fenti példánál maradva a levél átvizsgálához is számos funkció társul:

- A levél fejlécének analízise
- Levél fejléce alapján spam ellenőrzés pl. RBL szűrőlisták segítségével
- A levél kicsomagolása, a MIME komponensek kibontása, esetleges tömörített fájltartalmak kibontása
- Az egyes fájlok, tartalmak egyedi átvizsgálása vírusvédelmi szempontból (akár több különböző vírusirtó segítségével)
- Vírus észlelése esetén riasztás küldése a megfelelő személynek, esetleges értesítés a küldő és/vagy fogadó felé a vírus észleléséről, a vírusos küldemény tárolása
- Levél tartalma alapján heurisztikus és egyéb spam szűrési vizsgálatok

A levelezés vírusmentesítése *Amavis* segítségével

Az Amavis nem víruskereső, csak egy köztes szoftverkomponens, amely összefogja a védelmi rendszert. Milyen más komponenseket tartalmaz az Amavis?

- Kitömörítő rutinokat a különböző fájlok kicsomagolásához
- A levélhez csatolt fájlok kiterjesztése és más információk alapján előzetes tiltó szabályok is beállíthatóak
- Az Amavis képes kooperálni a legtöbb unix alapú víruskereső programmal. A programnak egy ideiglenesen létrehozott alkönyvtárat kell csak átnéznie, nem lell értenie sem a levelezéshez, sem a kitömörítéshez, ez az Amavis dolga.
- Naplózási funkciójával kövhető a vírusmentesítés folyamata, időszükséglete.
- A kéretlen reklámlevelek kiszűrését a Spamassassin segítségével tudja megvalósítani, amely további komponensként használja a Razor spam-adatbázisát és különféle RBL spam-védelmi szűrőlistákat.
- Kivételeket kezelő rutinjaival beállíthatóak olyan felhasználók, akiknek a vírusos/kéretlen reklám levelek elküldhetőek és olyanok, akiknek nem
- Értesítő rutinjai segítségével értesítés küldhető a feladó számára levele sikertelen kézbesítéséről és a címzett számára is. Az értesítő letiltható a feladót hamisító vírusok esetén.

Az Amavis tehát egy olyan szoftver, amely integrálja és megszervezi rendszerünk különböző funkcióit. Az Amavis segítségével tehát gyakorlatilag bármilyen víruskeresőt alkalmazhatunk a levelek vírusmentesítésére, azonban maga az Amavis semmilyen víruskeresőt nem tartalmaz. Az Amavisnak több különböző verziója létezik, ezeket jelenleg párhuzamosan fejlesztik. Át kell nézni, hogy mely szolgáltatások szükségesen számunkra és ezalapján dönteni.

Az Amavis PERL-ben van programozva és minden nagyobb levelezési kiszolgálóval együtt tud működni. Egyes esetekben a levelező kiszolgáló eleve lehetőséget nyújt levelezési filterek üzemeltetésére. Más esetekben az Amavis azt tudja kihasználni, hogy a levelezési kiszolgáló rávehető a továbbításra az Amavis irányában, majd az Amavis mint egy új levelet tudja visszainjektálni a szűrt adatokat a rendszerbe. Természetesen ilyen esetekben meg van oldva, hogy a levél ne kerüljön hurokba.
Sendmail esetén a szűrést a Sendmailhez szabványosított milter (mail-filter) protokollon keresztül tudjuk csatolni, amihez az Amavis hasonló nevű komponense, az Amavis-milter tud csatlakozni.
Telepítéskor fontos kérdés annak a vizsgálata is, hogy csak azokat a leveleket akarjuk szűrni, amelyek helyi kézbesítésre kerülnek, vagy azokat is, amelyeket a rendszer továbbküld (relay-zik).

Mailscanner

A Mailscanner az Amavishoz hasonlatos elven működik. Maga a Mailscanner nem tartalmaz víruskeresőt, de számos különböző víruskeresővel tud együttműködni a szűrés során. Az Amavishoz hasonlóan a Mailscanner is kapcsolódhat számos MTA-hoz, képes Spam ártalmatlanításra, és felfedezhet alapvető problémás leveleket önmagában is (pl. tiltott fájlnév).
A Mailscanner Sendmail MTA esetében kihasználja a Sendmail funkcionalitásának szétbontását. A Sendmail egyik komponense elvégzi a levelek beérkeztetését, azaz a hálózatról megkapott levelek tárolását a bemeneti sorba. A Mailscanner a bemeneti sort dolgozza fel és a leveleket szűrés után áthelyezi a leveleket kezelő rendszer feldolgozási sorába. Ez a funkcionalitás biztosítja azt, hogy a levelek csakis vírusmentesítés után kerülhetnek kézbesítésre, nem lehetséges az, hogy bármely hiba folytán a szűrést megkerülve jussanak el a célhoz.
Ez a működési mód egyesekben azonban ellenszenvet is kelt: A Mailscanner önmaga kezeli a levelezési sorokat, és esetleg nem szabványos vagy megfelelő módon manipulálja azt, amiből inkompatibilitások alakulhatnak ki. A probléma megítélése ízlés kérdése.

Vannak lényeges különbségek az üzenetek feldolgozásában, és a rendszer működésében is. A Mailscanner a vírusos mellékletet képes kiszedni, kicserélni, illetve tartalmaz néhány biztonsági kiegészítőt: Óránkénti automatikus vírusadatbázis-frissítési funkció, rendszeres újraindulás az esetleges memória felszabadítási hibák ellen, stb. Fontos azonban eldönteni, hogy milyen funkcionalitásra, és milyen teljesítményre van

szükségünk, valamint ezt melyik MTA-val akarjuk létrehozni, és ezalapján kell eldönteni a használt eszközt.

Démoziáció

A nagy levelezési forgalommal rendelkező szerverek esetében fontos kérdés, hogy mekkora teljesítményt emészt fel a vírusvédelem. Míg a Mailscanner esetében egyetlen PC vírusmentesítési teljesítményét (adott mérési feltételek mellett) mintegy 10 millió üzenet/nap-ra teszik, addig más mérések szerint egyetlen levél ellenőrzése 2-3 másodpercet vehet igénybe, ami 10-20 ezer napi e-mail üzenetnél több továbbítását nem teszi lehetővé. A rendszer teljesítménye sok tulajdonság együttesén alapul, azonban vannak alapvető strukturális megfontolások.
Amennyiben a levél kezelését végző rutinokat (MIME szétdarabolók, kitömörítők) és a vírusmentesítés rutinjait (vírusirtók) minden egyes levél esetében be kell tölteni, a hozzájuk kapcsolódó könyvtári rutinokkal egyetemben, akkor az felesleges többletmunkát jelent. A többletmunka természetesen annál nagyobb, minél több levél ellenőrzését kell elvégezni.

Ebből a nagy munkaigényből jelentős részt lehet megtakarítani azzal, ha szoftvereinket folyamatosan futásra kész állapotban tartjuk a memóriában. A megoldás biztosításra érdekében a felhasznált szoftverek egy része daemon, azaz folyamatosan futó háttéralkalmazás formájában is elérhető. A „démonizált" Amavis betöltődéskor megpróbálja betölteni mindazon rutinokat, amelyek működéséhez szükségesek. Futáskor egy kis, C-ben programozott rutint kell csak futtatnia a rendszernek, amely egy szabványos UNIX domain socketen keresztül adja át az adatokat vizsgálatra, és várja meg a vizsgálat eredményét. Az így megvalósított szűrés tehát nagyobb teljesítményre képes, hiszen csak egy kis rutin betöltődését kell a rendszernek megvárnia.
A háttérben futó Amavis számára fontos azonban, hogy a víruskereső is a háttérben fusson, hiszen az is sok időt emészt fel, amíg a vírusirtó betölti vírusadatbázisát és saját rutinjait. Ma már a víruskeresők jelentős része elérhető ilyen, háttérben futó, daemon változatban. A vírusirtó mellett démonként futhat még a spam szűrésére alkalmas spamassassin is. Természetesen egyes komponensek továbbra sem a háttérben futnak, ilyenek például egyes kitömörítő algoritmusok, illetve a démonokat összekötő rutinok.

Víruskereső kiválasztása

A víruskereső kiválasztása igen kényes eleme a vírusvédelmi rendszernek. A kiválasztást egyszerre határozza meg az ár, a hatékonyság, a megbízhatóság, az intergrálás minősége, és egyéb paraméterek és feltételek.
Jelen rendszerünkben a legfontosabb kérdések, hogy az adott vírusirtó alkalmas-e unixos futásra, van-e licenszelési probléma ezzel kapcsolatban (pl. mailboxok számától függő licensz).
Fontos kérdés, hogy a szoftver tud-e démon módban futni, hogyan lehet frissíteni a vírus adatbázisát.

ClamAV

A Clam AntiVirus egy közösségi fejlesztésű, GPL licenszű unix alapú víruskereső. A ClamAV korábban nem tudta felvenni a versenyt a kereskedelmi szoftverekkel, mivel sebessége parancssori módban futtatva viszonylag alacsony volt, kevés vírust ismert, és lassan kerültek be az új virusok információi.
Jelenleg viszont elmondhatjuk, hogy a ClamAV már elfogadható minőségben és sebességben egyaránt, és így valódi alternatívát jelent a kereskedelmi szoftverekkel szemben. A ClamAV létezik parancsori futásra alkalmas módban, de van démonként futni képes verziója is. Vírusadatbázisa közösségi alapon fejlődik, bárki beküldhet új vírust, de akár saját maga elkészítheti a vírushoz tartozó szignatúrát is és kérheti annak beillesztését a szoftverbe. A ClamAV a vírusadatbázis frissességét garantáló démonnal is rendelkezik, amely megfelelő rendszerességgel képes lekérdezni a friss vírusadatbázis megjelenését.
A levelezés vírusellenőrzésénél több vírusirtó egyidejű használata is megoldható, így célszerű számba venni azt a lehetőséget, hogy egy kereskedelmi szoftver mellett a ClamAV kiegészítőként futva támogatja azt, hiszen két víruskeresőnél nagyobb lehet az esély egy új vírus korai felismerésére. Természetesen nagyobb a hibás pozitív válaszok esélye is, hiszen a tévedés esélye is nagyobb.
A közösségi munkát ilyen esetben az is előre lendítheti, ha beküldjük azokat a vírusokat, amelyet más vírusirtó megtalált, ám a ClamAV még nem ismerte fel. Erre gyakorlatilag automatizmusokat is lehet rendszerünkbe építeni a nyílt és módosítható forráskódnak köszönhetően.

OpenAntivirus

Az OpenAntivirus egy ingyenes, GPL licenszű antivirus-fejlesztési kezdeményezés. Valójában itt nem egy szoftver fejlesztését tűzték ki célul, hanem több, különböző célú projekten dolgoznak. A projektek között találhatunk Java környezetben írt víruskereső alkalmazást, Squid és Samba vírusvédelmi modulokat, stb. Főleg azok számára lehet érdekes az OpenAntivirus, akik különleges környezetben kényszerülnek víruskereső alkalmazás használatára.

Természetesen a fent említetteken kívül számos más szoftver és alkalmazás létezik, beszéljünk akár kereskedelmi, akár ingyenes rendszerekről.

Tartalomszűrés

A vírusvédelem nem áll meg az e-mailek vírusmentesítésével. Ártó kódok természetesen letölthetőek a webről is, ftp szerverekről és más forrásokból. Amikor pedig már belső hálózatunkban jelenik meg a vírus, akkor védendőek szervereink, főként a munkacsoportok kiszolgálását végző fájl-szerverek. A védelmet a tartalomszűrő

alkalmazások, tűzfalak jelentik. Voltaképpen itt azért nem vírusvédelemről írunk már, mert ezen a területen végképp keverednek a vírusok, az ártó kódok és a szűrendő információk. A tartalomszűrés során éppúgy szükség lehet a pornográf tartalmak kiszűrésére, mint az Internet Explorer hibáit kihasználó kódrészletek elvetésére.

A létrejött nyílt forráskódú, ingyenes projektek száma a tartalomszűrés területén is igen nagy, nem beszélve a kereskedelmi termékekről. A projektek között említhetjük httpf tartalomszűrő proxyt, a DansGuardian Anti-Virus Plugint amely vírusvédelmi kiegészítés egy proxy alkalmazáshoz, stb.

Squid

A Squid proxy egy általánosan használt web és ftp caching proxy alkalmazás, amely a gyorsítótár alkalmazása és a belső gépek elrejtése mellett igen alkalmas hozzáférésvédelmi feladatokra is. A hozzáférésvédelem eszköze lehet a letöltendő URL-ek szűrése, amelyet statikus megoldásokon túl pl. a SquidGuard alkalmazás segítségével is elláthatunk.
Érdemes azonban a proxyt is kiegészíteni vírusvédelemmel.
A *Viralator* programcsomag PERL nyelven írt eszköz, néhány víruskereső integrálását teszi lehetővé a Squid proxyba.
Az OpenAntivirus Squid-vscan modulja a squid-filter csomag segítségével az OpenAntivirus ScannerDaemon projektét kihasználva oldja meg a víruskeresést.

Samba

A Samba 3-as verziója óta támogatja VFS (Virtual File System) modulok használatát. Ezek a beilleszthető modulok segítik a Samba fájlrendszerének biztonságosabbá tételét. A jelenlegi mintamodulokkal megoldható a törölt fájlok tárolása „kuka” könyvtárban, megoldható a fájlhozzáférések naplózása, és a samba-vscan modul segítségével megoldható a vírusellenőrzés is a fájlhozzáférések alkalmával.
A VFS rendszer jelenleg még túl fiatal, hogy igazán stabil működésről beszéljünk, így bevezetésekor körültekintően kell eljárni.

Fájlrendszer szintű szűrés

A vírusos fájlok ellenőrzésének támogatására használható mintamegoldás az RSBAC rendszer Malware Scan modulja. Az RSBAC rendszer a UNIX (Linux) fájlrendszer szintű védelmének kiterjesztése. Az RSBAC segítségével nemcsak a szokásos hozzáférésvédelmi rendszereket használhatjuk, de lehetőség van Mandatory Access Control (MAC) illetve például szerep alapú hozzáférés-engedélyezés (Role Based Access Controll - RBAC) használatára is. Az RSBAC azonban nem egy zárt rendszer, a hozzáférésvédelem újabb modulokkal bővíthető. A programcsomag egyik mintamodulja ezt mutatja be a vírusellenőrzés folyamatán keresztül.

Az RSBAC Malware Scan modulja a fájlhozzáférés valós idejében tudja ellenőrizni kernel szinten azt, hogy egy fájl vírusos-e, és amennyiben vírusos, megtilthatja a fájlhoz való mindenfajta hozzáférést. A védelem a beprogramozott információkon túl alkalmas külső vírusirtó kezelésére, de mivel az implementáció csak egy minta, jelenleg csak az F-Prot illetve a ClamAV és annak démonként futó verziója van támogatva.
A megoldás hasznos lehet pl. Samba szerver esetén is, hiszen így a hozzáférés nem a szerver-program szintjén, hanem egyel mélyebb rétegben, általánosabban akadályozható meg. Fontos viszont, hogy ha rosszul állítjuk be a rendszert és a vírusirtó nem fut megfelelő módon, úgy adott esetben lehetőségünk van arra, hogy a rendszert összeomlasszuk: Amennyiben minden fájlhoz való hozzáférés megszűnik, úgy esetleg még a védelem kikapcsolására sem lesz lehetőségünk. Körültekintő implementációval természetesen ez a probléma áthidalható.

A fentiekben a vírusvédelmi projektek közül csak néhány jobban elterjedt alkalmazást mutattam be, számos kisebb-nagyobb, különféle célú egyéb alkalmazás is létrejött már, nem beszélve a kereskedelmi célú termékekről.
Rövid összegzésként elmondhatjuk, hogy a UNIX alapú vírusvédelem már ingyenes eszközökkel is számos különböző módon és céllal véghezvihető feladat, így érdemes is alkalmazni. Unixos állomásaink biztos pontot jelenthetnek hálózatunk vírusvédelmében és elősegíthetik szerverezetünk biztonságának megerősítését.

DDoS támadások

A szolgáltatás-megtagadásos támadások (Denial-of-Service, DoS) kiemelt jelentőséggel bírnak az Internet biztonsági problémái között. A DoS támadások során a támadó célja nem az, hogy a hálózatba behatoljon, kizárólag az, hogy megakadályozza annak megfelelő, üzemszerű működését. Ennek következtében a támadó eszköztára igen tág, míg a védekezés eszköztára kicsi. Gyakran a hálózat biztonsági alrendszerei maguk teszik a hálózatot veszélyeztetetté DoS támadásokra.

A DoS támadások általános esetben elosztott támadások, ahol több támadó együttes cselekedettel kívánja előidézni a rendszer összeomlását. Ezt Distributive DoS támadásnak, azaz DDoS-nek hívjuk. A DDoS extrém esete az, amikor csak egyetlen állomásról indítanak támadást (DoS).

A DoS típusai

Protokoll hiba

A DoS támadások egyszerű változata az, amikor valamelyik protokoll vagy szoftver hibáját egy rosszindulatú támadó arra tudja felhasználni, hogy az adott szolgáltatást, vagy szervert működésképtelenné tegye. Ilyen közismert támadás volt a ping-of-death, amikor egy megfelelően formázott IP csomaggal „lefagyaszthatóvá” váltak egyes számítógépek.

A hibás szoftverekből, protokollokból adódó problémák egyszerű javítása a hiba korrigálása, hosszútávon a szoftverek frissességének és hibajavításának koordinált, jól szervezett telepítése és fenntartása.
Ennek megfelelően a protokoll hibák esetében egyrészt megoldható a hiba megszüntetése kijavítással, továbbá az ismeretlen hibák elleni védekezés is támogatható különböző módokon. (pl. tűzfal, nem használt szolgáltatások letiltása, heterogén több kiszolgálós környezet felépítése stb.).

Hálózati elárasztás

A DoS támadások veszélyes esete a hálózati elárasztás. Ilyen esetben többnyire nagyszámú kliens egyszerre kezd forgalmazni nagy adatmennyiségeket egy vagy több szerver irányában. a nagyszámú kliens összforgalma olyan magas lehet, hogy azt a szerver hálózati kapcsolatai és esetleg erőforrásai sem bírják kiszolgálni.
Az ilyen hibák ellen védekezni a megtámadott gépnél többnyire igen nehéz.

- Amennyiben az elárasztó forgalmat nem tudjuk különválasztani a legitim forgalomtól, úgy a forgalom kitiltása csak úgy lehetséges, ha a legitim felhasználók forgalmát is kitiltjuk, és ezzel már önmagában megvalósul a szolgáltatás-megtagadás célja.
- Amennyiben az erőforrásaink növelésével kívánunk védekezni, úgy a támadó is növelheti erőforrásait, a támadás méretét több támadó bekapcsolásával.
- Amennyiben sikerül azonosítani a támadó forgalmat és különválasztani azt a legitim forgalomtól (pl. forgalomstatiszikai alapon, azonosítással, vagy a lekérdezések kategorizálásával) úgy a felesleges forgalom kiszűrhető. Ha azonban a szűk keresztmetszetet közvetlen Internet-kapcsolatunk jelenti, úgy a fogadó oldali szűrés hasztalan: A szűrést ott kell elvégezni, ahol még elég hálózati kapacitás áll rendelkezésre a legitim és támadó forgalom egyidejű érkeztetésére. ([4]) Ennek megfelelően az ilyen jellegű védekezés igen nehéz, vagy akár lehetetlen akkor, ha a védelmet csak a fogadó fél oldalán kívánjuk kiépíteni.

Szerver túlterhelés

A hálózati elárasztás többnyire buta módon megpróbál felemészteni minden hálózati kapacitást, és kihasználva a támadó nagy számú támadó állomásából adódó óriási kapacitását éri el a célját. Sokkal kifinomultabb módszer azonban az, ha a hálózati protokollokban, szolgáltatásokban olyan elemet sikerül találni, amelynél a kérdező fél (kliens) viszonylag kis ráfordítással (számítási, hálózati kapacitással) el tudja érni, hogy a kiszolgáló nagy ráfordítással válaszoljon a kérésre.
Ilyen lehet az, ha egy kiszolgáló egy kis kérésre hosszú adatmennyiséggel válaszol, vagy az, ha egy rövid kérés kiszolgálása hosszú számítást követel meg a szerver oldalán. Egy rosszul kivitelezett adatbázis alapú weblap esetén elképzelhető például, hogy egy kérés kiszolgálása az adatbázisban olyan, rosszul indexelt, nagy adatmennyiséget feldolgozó feladatokat indít el, amely kiszolgálás erőforrás-felhasználása jóval meghaladja a kérés igényeit.

Az ilyen támadások esetében a támadó legitim, kis adatforgalom mellett is meg tudja bénítani a szervert. A védekezés természetesen elképzelhető, ha tudjuk mérni, mely kérések veszélyesek, és meg tudjuk automatikusan vizsgálni, hogy valaki direkt ilyen kérésekkel fordul-e hozzánk. Kétséges azonban, hogy pusztán a kérés analízisével megkülönböztethető a támadó és egy legitim felhasználó, valamint az, hogy ennek segítségével úgy sikerül megvédenünk a szervert, hogy legitim felhasználót nem veszélyeztetünk.

Legutóbbi támadások

A legutóbbi évek DoS támadásai a legtöbb esetben egyszerű elárasztásos támadások. Az elárasztás is lehet sokféle, a gyakorlatban felmerült támadások többsége a TCP SYN adatcsomagjait kihasználó támadás, illetve ICMP forgalomra építő támadás. Mindkét támadási formánál mód van a feladó IP címének hamisítására is, míg például a túlterheléses támadásoknál ez ritkábban fordul elő, a feladó IP címének hamisítása pedig segíti a támadó elrejtését.
Alkalmanként előfordul protokoll hibát kihasználó DoS támadás is, de ezek többnyire rövid ideig tartanak, hiszen a hiba kijavításával a probléma gyorsan megszüntethető.

Számos DoS támadást hajtottak és hajtanak végre folyamatosan a kéretlen reklámlevelek ellen létrejött megoldásokat nyújtó cégek ellen. Ez a folyamat azt sugallja, hogy a kéretlen-reklám-ipar egyre veszélyesebb lehet az Internet számára.

Az utóbbi időszak vírustámadásainál egyre gyakoribban derült ki az is, hogy a vírusba olyan rutint programozak, amely valamely cég hálózatának DDoS támadással történő lebénítását célozza meg. A Mydoom.A vírus így támadta meg 2004. februárjában az SCO hálózatát, illetve a vírus átírt, Mydoom.B változata hasonló módon a Microsoft hálózatát. A támadás itt elárasztásos támadást jelent, oly módon koordinálva, hogy az egyes víruspéldányok azonos időpontban kezdik támadni a cég lapját kiszolgáló szervert.

Az SCO elleni támadás látszólag sikeres volt, mivel a cég weblapja elérhetetlenné vált, igaz, többen megkérdőjelezik azt, hogy az SCO nem tudott volna-e védekezni a támadás ellen, valamint megemlítik annak a lehetőségét, hogy az SCO maga tiltotta le weboldalát a támadás kezdetekor. A Microsoft oldala ezzel szemben működőképes maradt, ami jórészt annak köszönhető, hogy a cég weblapját egy igen nagy kapacitással rendelkező tükröző hálózatok keresztül teszi elérhetővé.

A vírusvédelem és a DoS támadások kapcsolata

A vírusvédelem és a DoS támadások több helyen kapcsolódnak egymáshoz. A víruskódba elhelyezett DoS támadást segítő rutin mellett számos más probléma is létezik.

- Amennyiben a vírus éppen járványszerűen kitör, elárasztathatja a felhasználók postaládáját, akik (amennyiben nem védekeznek a vírusok ellen) a sok levél miatt nehezen találják meg fontos leveleiket vagy betelik postaládájuk
- A vírusvédelmi eszközök a feladók részére gyakran hibaüzenetet küldenek („az ön által küldött üzenet vírust tartalmaz így nem került továbbításra”), amely elárasztathatja a hamisított feladó postaládáját. (lásd dumaru vírus)
- Amennyiben a vírus hamisított feladóval továbbítja magát ám a címzett hibás, úgy a hamisított feladóhoz nagyszámú kézbesítési hiba visszaigazoló üzenet érkezhet, amely DoS-t eredményezhet.
- Amennyiben valamely levelező szerverre a vírus egyszerre igen sok levelet kézbesít (pl. lassú Internet kapcsolat kifele mutató levelező tűzfalára egy belső fertőződés után), úgy az leterhelheti a hálózati kapcsolatot, továbbá a vírusos levelek víruskeresése a levelező kiszolgáló teljes leterheltségét eredményezheti.

Mit tehetünk az említett problémák ellen?

Egyes problémák ellen gyakorlatilag semmit nem tehetünk: A nevükben hamisított levelek kézbesítési értesítőit szűrhetjük, különválogathatjuk, de más védekezést nem tudunk tenni.
Segítséget jelenthet az egyelőre kísérleti üzemben levő SPF rendszer bevezetése, ennek használatával elérhető lenne, hogy a nevünkben hamisított levelet feladni ne lehessen, azonban ennek is számos problémája van, így jelenleg még nem jelent megoldást a feladatra.

A víruskereső rendszerektől visszaérkező vírus riasztások által okozott károk viszont csökkenthetőek. A megoldást egyrészt az jelenti, ha a vírusos levelekről nem küldünk riasztást a feladónak. Ilyen esetben természetesen arról sem értesül, ha egy csatolt word fájlt tartalmazó levele makróvírus miatt nem került továbbításra, így egyes esetekben a riasztás teljes kikapcsolása nem célszerű.
Az Amavis frissebb változataiban azonban például beállíthatóak azok a vírusok, amelyek hamisítják a feladót, és ezen vírusok esetében a rendszer nem küld kézbesítési riasztást. A vírusok listája jelenleg egy reguláris kifejezés, amelynek a vírus nevére kell illeszkednie. Remélhetőleg a probléma később fejlettebb eszközökkel is megoldható lesz.

Vírus-szűrő levelező kiszolgáló elleni támadások

A vírusok által okozott DoS problémák mellett a vírusokkal kapcsolatos másik gond az, hogy maguk a vírusvédelmi rendszerek is érzékenyek lehetnek DoS támadásra.
Ilyen tipikus támadások:

- A vírusvédelmi e-mail átjáró elárasztása vírusos vagy nem vírusos levelekkel
- A vírusvédelmi átjáró elárasztása olyan formai hibás levelekkel, amely az átjáró működését megbénítja.

Az első típusú támadás esetén a támadó azt használja ki, hogy a vírusvédelmi rendszer nagy erőforrásokat emészt fel. Egy Amavissal, vagy Mailscannerrel felszerelt rendszer kapacitása, főként ha több víruskeresőt is használunk, véges. A kapacitás elérésénél előfordulhat, hogy a rendszer a túlzott kapacitásfelhasználás miatt újraindul (pl. watchdog alapján), vagy a levelek elfogadását a túlterhelés megszűnéséig a rendszer megszünteti. Bárhogyan is történik, a túlterhelés mindenképpen a legitim levelek késését fogja okozni.

A második típusú támadás esetében a protokoll ill. szoftverhiba típusú DoS támadásról beszélhetünk. Ilyenre minták:

- Olyan levél küldése, amely olyan tömörített állományt tartalmaz, amely nagyon hosszú, de igen redundáns (pl. több GB méretű, csak azonos betűből álló fájl). A csatolt fájl mérete így korlátos, azonban a víruskeresés céljára történő kicsomagolás esetén a szerveren a hely elfogyhat, a rendszer megbénulhat.
- Olyan levél küldése, amely sok, kise fájlt, vagy további tömörített fájlokat tartalmaz több mélységben, így leterheli mind a kitömörítő, mind az ellenörző rutinokat.
- Formai hibás fájl küldése, melynek dekódolását végző rutinok a rendszer bénulását eredményezhetik.

A fenti, második típusú problémák a hibák kijavításával, a védelem megerősítésével többnyire kiküszöbölhetőek. Az Amavis programcsomagban például korlátozva van a beágyazott tömörítési szintek száma, továbbá mód van arra is, hogy bizonyos tömörítési arányon túl ne lehessen tömörített fájlt csatolni, így a szabad kapacitást az ellenőrzés már nem fogja felemészteni. Hasonló módon korlátozható a vírusvédelmi rendszer teljesítményszükséglete is: Egy kvóta meghaladása után a védelmi rendszer nem vizsgálja tovább az adott levelet. Természetesen az így kiszűrt levelek a hálózati szabályozásunknak megfelelően kerülhetnek azonnali elutasításra, vagy későbbi vizsgálatok céljára karanténba stb.

Vírusvédelmi rendszer DoS front-end

Mit tehetünk az ellen a támadás ellen, amikor a rendszerünket legitim levelekkel árasztják el, annak céljából, hogy a nagy teljesítményigényű védelmi rendszerünk a szerver túlterhelését okozza?

A kérdés megválaszolása nehéz. Természetesen fokozhatjuk rendszerünk teljesítményét, és megpróbálhatjuk elkülöníteni a támadó szándékú e-maileket a legitim forgalomtól. Elképzelhető azonban, hogy ez nem tehető meg, vagy hogy rendszerünk már viszonylag kevés (néhány tíz, néhány száz) e-mail segítségével megbénítható rövidebb időre, amit el szeretnénk kerülni. A helyzet javítását eredményezheti a levelezés protokoll szintű módosítása client-side puzzle ([2]) technika segítségével. A technika segítségével javítható a teljesítmény-ráfordítás aránya a kiszolgáló és a küldő között, és így megelőzhető a DoS helyzet kialakulása. A technika javítására magunk is javaslatot tettünk ([3]) játékelméleti megközelítés alkalmazása segítségével.

A megoldáshoz a [1] cikkben ismertetett forgalomanalízis módszerét használó elméleti munkánkat kívánjuk bemutatni és javasolni.
A módszer a következő algoritmusokra épít:

- A támadás észlelése
- A támadók elkülönítése a legitim felhasználóktól
- A támadó forgalom kitiltása
- Sikeresség ellenőrzése

Az algoritmus felületes leírása a vírusvédelem esetén a következő:

A támadás észlelése

A támadás észlelése a levelezési forgalom folyamatos statiszikai analízisével zajlik. Amennyiben a forgalom hirtelen megnő, úgy a rendszer azt támadásként érzékeli, és innentől kezdve egy korlátozott ideig részletes statisztikát készít, hogy milyen állomások hány üzenetet küldenek.

A támadók elkülönítése a legitim küldőktől

A hosszú távú és rövid távú statisztikai mérések alapján a rendszer megbecsüli, hogy a támadó mekkora forgalmi többletet generál.
A részletes statisztikák alapján a rendszer feladat kiválasztani azokat az állomásokat, akik a legtöbb forgalmat generálják, méghozzá annyi ilyen állomást kell kiválasztani, amelyek összforgalma megegyezik a támadás becsült méretével.

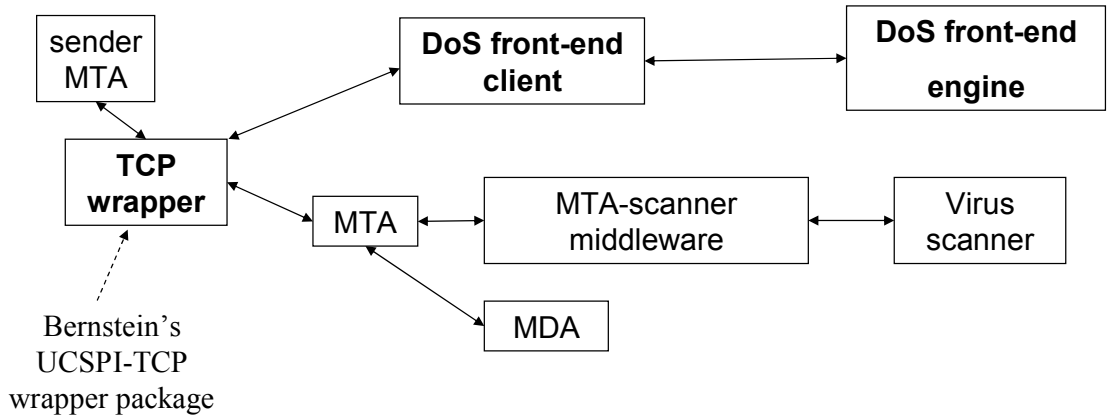
Miért azokat az állomásokat kell kiválasztani, amelyek a legtöbb forgalmat generálják? Ez az algoritmus kulcsa. A támadó, ha DDoS támadást hajt is végre, mindenképpen megpróbálja a támadó gépek számát minimalizálni, hiszen a támadó gépek számának növelésével növekszik annak az esélye, hogy a támadás előkészítése vagy végrehajtása során kiléte kiderül. A minimalizálás mellett viszont olyan mennyiségű forgalmat kell generálnia, amely alkalmas a célszámítógép túlterhelésére. Ennek megfelelően a támadó által használt munkaállomások rendszerint nagyobb forgalmat generálnak, mint a legitim munkaállomások.

A támadó forgalom kitiltása

A felismert támadóktól a leveleket (megfelelő időn át) nem fogadjuk el. Amennyiben ily módon véletlenül legitim felhasználókat is kitiltottunk, úgy ezek a felhasználók az SMTP protokoll hibatűrése folytán később még megpróbálhatják leveleik elküldését. A rendszer ezért robosztus: A levelek még akkor sem vesznek el, ha téves pozitív választ adna védelmi rendszerünk.

A védekezés sikerének ellenőrzése

A védekezés sikerességének ellenőrzése természetesen a forgalom vizsgálatán alapul: Amennyiben a forgalom megfelelő mértékben csökken egy elfogadható szintre, úgy a



1. ábra. Vírusvédelmi rendszer DOS védekezés prototípusa

védekezést sikeresnek tekinthetjük. Amennyiben a védekezés nem sikeres, úgy a B. C. illetve D. pontok újbóli alkalmazása válhat szükségessé.

Prototípus

A DoS védelmi rendszerre prototípust készítettünk el, mely az 1. ábrán látható. A mintaimplementáció több részből áll: A Bernstein-féle UCSPI-TCP csomag átírt RBLSMTPD komponense fogadja a levelezést, és adja tovább az EXIM felé a beérkezett kapcsolatokat. Az Exim-re épülő antivirus struktúra képezi a védett rendszert. Ez egy amavisd-new vírusszűrő modult használ (MTA-Scanner middleware), amely ClamAV illetve egy másik kereskedelmi vírusirtó segítségével védi a levelezést.
A forgalmi analízis a átírt TCP Wrapper (RBLSMTPD, itt aDoSSMTPd) segítségével zajlik. A rendszer másik komponense az általunk aDoSd-nek hívott statisztikai adatgyűjtő mag. Az TCP Wrapper (ADoSSMTPd) egy Unix domain socket segítségével kérést intéz a statiszikai maghoz, hogy az adott IP szám küldhet-e levelet. Erre a statiszikai mag a szűréseknek megfelelően visszajelez, illetve frissíti belső statisztikai komponenseit. (Tárolja a kérést).
A statiszikai mag rendszeresen, jelenleg 2 másodpercenként összegzi statiszikai eredményeit és elvégzi a kívánt vizsgálatokat a fentebb ismertetett algoritmusok szerint.

A statisztikai magot a könnyebb bővíthetőség és átláthatóság érdekében PERL-ben írtuk meg. A statisztikai mag természetesen naplózást is végez, illetve egy külön kis programocska segítségével adatok kérdezhetőek le belső állapotváltozóinak jelenlegi értékeiről.

Működőképesség

A jelenlegi prototípus egyelőre teszt üzemben vizsgáltuk meg. Az hamar kiderült, hogy a levelezési forgalom egyes specialitásai az [1] publikációban ismertetett algoritmusok megváltoztatását igénylik. A levelezés forgalma hajlamos burst-ök kialakulására, amelyek az ablakméretek jó kalibrálását igénylik, illetve a kis forgalmú szakaszokban szükséges az algoritmus olyan korlátozása, hogy bizonyos minimális forgalom alatt ne tekintsen támadásnak egy hirtelen forgalmi ugrást (éjjeli üzem).

Vírusok

A prototípus véleményünk szerint nemcsak a DoS támadók, de a vírusok ellen is hatékony lehet. Amennyiben egy munkaállomás megfertőződik, és elkezdi terjeszteni saját magát, úgy a forgalmi ugrás alapján a DoS védelmi rutinok ezt támadásnak ítélhetik, és az illető állomást kitilthatják. A kitiltásról a rendszergazda értesítést kaphat, és felderíthető, hogy mi volt a tiltás valódi oka, felfedezve a vírust. A rendszer érdekessége, hogy mivel a forgalom analízisére épül, így a még ismeretlen vírusok is felfedezhetőek, illetve a detektált támadások összefoglalhatóak egy nagyobb, disztributív rendszerbe is, amely alkalmas lehet a vírusjárványok korai detekciójára és a hatékonyabb ellenintézkedések elvégzésére.

Természetesen elképzelhető olyan eset is, amikor nem támadó, és nem is vírus okoz vélt támadást, hanem egy vagy több belső vagy külső felhasználó körlevele indítja be a támadást érzékelő algoritmust.
A rendszer ez ellen úgy védhető, hogy adott állomások ún. „white list” módszerrel a statisztikai magból kikerülhetnek, ezeknek joguk van bármilyen mennyiségű levél küldésére. Másrészt gyakran előfordul, hogy a túlterhelést hasonló módon belső felhasználók legitim levelei okozzák. Ilyen esetben a rendszer a felhasználót egy időre ki tudja tiltani, aminek eredményeképpen a felhasználó figyelmezethető. Hasonlóképpen, a tiltás lejártával a felhasználó számítógépe a körlevelet el tudja küldeni azokhoz a címzettekhez, akikhez az előző futás alkalmával nem sikerült, így a levelek nem vesznek el, viszont megkíméltük szerverünket egy esetleges túlterheltségtől.

Összefoglalás

Cikkemben röviden bemutattam a vírusvédelem jelenlegi helyzetét Linux környezetben. Megismertünk néhány megoldást, a kapcsolódó problémákat, fejlődési lehetőségeket, irányokat. Ismertettem a nyílt forráskódú fejlesztések főbb irányait a vírusvédelem területén.
Láthattuk, hogy a szolgáltatásmegtagadás problémája a vírusvédelem világában is fontos probléma, és számos helyen jelentkezhet. Láthattuk, hogy a problémák egy része további fejlesztéseket, vagy egyes internetes protokollok alapvető módosítását igénylik. Bemutattam továbbá egy olyan rendszert, amely reményeink szerint hasznos kiegészítője lehet az vírusvédett e-mail átjárók DoS (és túlterhelés) elleni védelmének.

Hivatkozások

[1] B. Bencsáth, I. Vajda, Protection Against DDoS Attacks Based On Traffic Level Measurements, 2004 International Symposium on Collaborative Technologies and Systems, 2004, edited by Waleed W. Smari, William McQuay, pp. 22-28., The Society for Modeling and Simulation International, San Diego, CA, USA, January, Simulation series vol 36. no. 1., ISBN 1-56555-272-5

[2] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In Advances in Cryptology -- Crypto '92: 12th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science volume 740, pages 139-147, Santa Barbara, California, August 1992. Springer.

[3] B. Bencsáth, L. Buttyán, I. Vajda, A game based analysis of the client puzzle approach to defend against DoS attacks, Proceedings of SoftCOM 2003 11. International conference on software, telecommunications and computer networks, 2003, pp. 763-767, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split.

[4] Ioannidis, J. and S. M. Bellovin. "Implementing Pushback: Router-based Defense Against DDoS Attacks." In Proceedings of Network and Distributed System Security Symposium, Reston, VA, USA, Feb. 2002, The Internet Society.