DHA VÉDELMI RENDSZER EREDMÉNYEINEK STATISZTIKAI VIZSGÁLATA

Szabó Géza (szabog@crysys.hu), Bencsáth Boldizsár (boldi@crysys.hu)

BME Híradástechnikai Tanszék, Laboratory of Cryptography and System Security

Abstract

Az alábbi cikkben a Directory Harvest Attack (DHA) támadásokkal fogunk foglalkozni. A DHA támadás lehetősége ismert volt idáig is, ám a sok alternatív levélcím összegyűjtési lehetőség miatt eddig nem kapott kiemelt fontosságot a kéretlen levélküldők által célpont összegyűjtés során használt eszközök között. Ahogy a felhasználók egyre jobban vigyáznak e-mail címeikre a DHA előtérbe került és a támadók elkezdték előszeretettel alkalmazni. A DHA támadó célja, hogy megszerezze a levelezőszerveren karbantartott felhasználói elektronikus levél címeket. Ezt úgy éri el, hogy nagy mennyiségű levelet küld a megtámadott levelező szervernek különböző címzettekkel és összegyűjti azokat a címeket, amiről nem kapott visszajelzést a szervertől arra vonatkozóan hogy a levél kézbesítése nem sikerült. A cikkben számba vesszük a lehetséges védelmi megoldásokat és ezen alapelveket felhasználva bemutatjuk az általunk implementált több program komponensből álló hálózaton keresztül együttműködő rendszert. A rendszer egy feketelistás megoldás, ahol a feketelista a támadók IP-címeit tartalmazza egy központi adatbázisban. A levelezőszerverek védelmét egy támadást bejelentő modul és a támadó levélküldését megakadályozó front-end modul látja el. A feketelista szerver egy DNS kiszolgáló tulajdonképpen, ahova a kliensektől érkező bejelentések és lekérdezések is DNS lekérdezés formájában utaznak. A DNS lekérdezésre a szerver egy IP-címmel válaszol ami a kliens oldalon jelentheti akár azt, hogy a kérdezetett IP támadóhoz tartozik, akár azt, hogy ártatlan. A meglevő rendszerek mellé beépítve azoknál erőforrás megtakarítást lehet elérni, mivel a DHA támadók levelei nem kerülnek a lassabb, erőforrás igényesebb tartalomszűrő mechanizmusok rostája alá, korábban ki lehet tiltani őket a rendszerből. A rendszer megfelelő működéséről és a védelem eredményességéről a valós környezetben való helytállása a legjobb bizonyíték. A rendszer által összegyűjtött eredményeket elemezzük és rendszerezzük, bemutatva, hogy milyen lehetséges csoportosításai lehetnek a támadóknak típusok szerint.

1. Bevezető

Az emberek az egyre növekvő kéretlen levelek áradatának, levélben terjedő vírusok és más kártékony kódok hatására egyre jobban meggondolják azt, hogy kinek is adják oda az e-mail címüket. Átgondolják, hogy meg merjék-e kockáztatni, hogy valamilyen online fórumon használják címüket, vagy akár azt is, hogy a weblapjukon rajta hagyják-e ezt a fontos személyi adatukat. Mindkét esetben ugyanattól kell tartani: a keresőrobotok képesek összegyűjteni a `mailto:user@levelcim` alakú hivatkozásokat. A fórumok ilyen szempontból kiemelten veszélyesek, hiszen ha kifejezetten nem korlátozzuk az e-mail címünk szerepeltetését, akkor minden hozzászólásunk mellé odakerülhet.

Egy alternatív e-mail cím gyűjtési lehetőség az emberi hiszékenységet kihasználó támadási forma: a hamis oldalak és kérdőívek módszere (*phishing*). Egy népszerű, látogatócsalogatónak tűnő oldalon valamilyen ügyes fogással (p.l nyereményjátékok, reklámajándékok) ráveszik az áldozatot, hogy beírja az adatait. Így garantáltan működő címeket tudnak a támadók összegyűjteni.

A fenti okok miatt a felhasználók általában tartanak más, akár egyszer használatos e-mail címet, gyakran valamilyen ingyenes szolgáltatónál, ami ha "odavész", sem baj. Ha a címet elkezdik elárasztani kéretlen levelek, akkor a felhasználó rövid idő után átvált egy másik címre, a régit lemondja, vagy magára hagyja és később a szolgáltató is törli.

Ha a felhasználó a fent említett e-mail cím gyűjtési lehetőségeket kizárta, mégis gondosan vigyázott e-mail címére egyszer csak elkezdenek kéretlen levelek özönleni, akkor e-mail szolgáltatója nagy valószínűséggel egy cím-kinyerő támadásnak esett áldozatul. A DHA témája sokszor előkerül, és a kereskedelmi anti-spam termékek egy hirtelen mozdulattal ki is pipálják az általuk nyújtott szolgáltatások listáján, elfelejtvén megemlíteni, hogy

milyen megoldást is használnak a támadás kivédésére. Ezeket a módszereket szeretnénk összefoglalni és javaslatot tenni egy hatékony védelmi mechanizmusra.

1.1. A címkinyerő támadás miért lehetséges?

A DHA problémája az SMTP protokollban [1] gyökeredzik: az e-mail szerverek, ha megfelelő e-mail címre kapták a levelet, úgy nem adnak visszajelzést, elfogadják a levelet. A szerver, ha nem létező felhasználó címére kap levelet, úgy vagy azonnali, vagy későbbi visszajelzést adhat arra nézve, hogy a felhasználó postafiókja nem létezik. Ez a folyamat információval szolgál a levelező-szerver által karbantartott e-mail címekről. A támadók ezt az információt használják ki, rengeteg levelet küldve az adott e-mail szervernek. Azokról a címekről, amelyekről nem érkezik válasz (a szerver negatív visszajelzés nélkül elfogadja a levelet), nyilvántartást vesznek fel. Ezek a címek minden valószínűség szerint érvényes felhasználói azonosítókhoz tartoznak, így érdemes lehet rájuk a későbbiekben kéretlen leveleket küldeni. A cím kijutás mellett problémát jelenthet a levelezést kiszolgáló szerver összeomlása. Az e-mail címek megszerzése érdekében a támadó rengeteg téves levelet küld a szervernek, amely így jelentősen, hosszú időre, és akár több támadótól is leterhelésre kerül. A leterhelés leköti a kiszolgáló hálózati kapacitását és processzorát is. Ez végeredményben egy DoS[1] támadást eredményez.

1.2. A támadás fajtái

A DHA támadásnak, azaz a címlista-kinyerő támadásnak, két típusa létezik: az egyik "brute force" jelleggel az összes lehetséges karakter, illetve szótag kombinációt kipróbálja, mint e-mail címet. A másik jóval szofisztikáltabb: tipikusan előforduló e-mail címeket generál vagy gyűjt emberek vezeték és keresztnevéből, gyakran előforduló szavakból, szóösszetételekből, továbbá ismert e-mail azonosítókból. Másik lehetséges csoportosítása a DHA támadásnak a felhasznált IP-címek száma alapján történhet: az "alap" változatban a támadó ugyanarról az IP címről próbálkozik, a másik esetben több IP címmel rendelkezik és ezeket felváltva használja a támadáshoz (*Distributed DHA*)

2. DHA-val kapcsolatos munkák

A fent bemutatott védelmi módszerekre épülnek kereskedelmi termékek is. Ezek funkciójukat tekintve inkább anti-spam termékek, és nem a DHA támadás ellen vannak kihegyezve. Nagy részt a RBL-alapú megoldásokat támogatják levél érkezésekor, azaz nyilvános RBL-listákon[2] ellenőrzik a feladó címét, hogy támadónak minősítették-e már korábban.

A *Kerio MailServer* [14] felfigyel a nem létező postafiókoknak küldött levelekre és egy bizonyos szám felett elkezdi szűrni a lehetséges támadókat. A *Secluda Inboxmaster* [15] konfigurálható SMTP hibaüzenetek beállítását teszi lehetővé: ha egy spamet detektálnak a levél kézbesítés közben, a szerver egy válasz üzenetet küld a feladónak, hogy nem létező e-mailre próbált levelet küldeni. Ezzel a megoldással az a legfőbb probléma, hogy a DHA támadást nehezen szűri ki, hiszen az ebben a támadásban résztvevő e-mailek általában nem tartalmaznak spamet, amit a spam-szűrő módszerek így nem jeleznek. A *Styx Mail Filter* [12] egy hardver-szoftver együttes, ami a kéretlen reklám levelek és vírusos tartalmak szűrését végzi a levelek levelező rendszerbe jutása előtt. Az alapkivitel szabad szoftvereket használ, így megtalálható benne a *ClamAV* [9] víruskereső és *SpamAssasin* [8] spam-szűrő. Ez utóbbi egy RBL-alapú megoldást foglal magában, amely kiegészül egy Bayes szabály-tanuló rendszerrel, *Razor*[3] és *DCC*[4] komponensekkel, ami a levelek szűrését elvégzi, de DHA támadást nem jelent az RBL-szerverek felé. Egyes termékek dokumentációja alapján nem tudni, hogyan működnek, de a hatékonyság miatt, nagy valószínűséggel RBL-alapúak, ilyen pl. az *eSafe Advanced Anti-spam Software* [13].

Az egyszer használatos e-mail címek szükségessége esetén egy lehetséges megoldást nyújthat a *mailinator* [7]. Egy autentikáció nélküli e-mail szervert valósítanak meg, ami semmi másra nem jó, mint hogy levelet fogadjon. Bármilyen címzettnek erre a doménjére érkező levelet elfogad, aminek a postaládáját meg lehet tekinteni belépve az oldalra. A leveleket és az ideiglenes postaládákat óránként ürítik, így arra lehet jó, hogy pl. egy fórumra való bejelentkezéshez szükséges megerősítő e-mailt elküldenek erre a helyre, amit megnézünk egyből, és megerősítjük belépésünket. Mivel semmilyen autentikáció nincs, ezért bárki meg bírja nézni bármelyik

---

[1] *Denial of Service*-Egy adott szolgáltatást az azt biztosító hardver vagy szoftver megbénításával vagy működésének zavarásával elérhetetlenné tevő támadás.

[2] *Real-time Black/Block List* - valós időben frissített fekete/tiltó lista

[3] *Vipul's Razor* [10] - egy elosztott, kollaboratív, spam felismerő és szűrő hálózat. A rendszernek egy állandóan frissülő adatbázisa van a felhasználók és a rendszert használó kliensek által beküldött spamek ujjlenyomatáról. (Azaz azokról a levelekről, amit a felhasználók spam-nek ítéltek.) Egy levél ellenőrzése úgy történik, hogy a levél ujjlenyomatát ellenőrzik, hogy szerepel-e a Razor feketelistáján.

[4] *Distributed Checksum Clearinghouse* [11] - A *Razor*-hoz nagyban hasonló megoldás, de a kliensek itt minden e-mail hash lenyomatát átküldik, és a rendszer azt a lenyomatot ítéli egy kéretlen reklám levél lenyomatának, amit nagyon gyakran jelentenek neki

postafiókot. Ezzel a módszerrel egyben egy honeypot[5]-ot is megvalósítanak, és meg tudják mutatni egy órára visszamenőleg, hogy ki az aki a legaktívabban küldözget nekik levelet.

3. A lehetséges védekezések

A DHA támadás ellen szóba jöhető védekezési mechanizmusokat fogjuk bemutatni a következő részben.

3.1. Új program elemet nem igénylő módszerek

A védekezés egyik lehetséges formája, ha nem telepítünk új programokat a meglévő levelező rendszer mellé, hanem valamelyik a következő pontokban bemutatott módszert használjuk.

3.1.1. E-mail cím választással

A védekezés a DHA támadás ellen történhet egyszerűen bonyolultra választott e-mail címekkel, ami a szótáras támadás ellen ideig-óráig véd, de a környezetünk nehezen fogja tudni megjegyezni új e-mail címünket. A védekezés ezen formája brute-force támadások ellen haszontalan.
Az e-mail címmel való védekezés másik lehetséges módja, ha egyszer használatos e-mail címet használunk. Ezzel a megoldással nyílván az a baj, hogy a kommunikáció elég egyoldalú lehetőségét teremti csak meg, mivel küldeni gond nélkül fogunk tudni levelet bárkinek, de ha választ is várunk egy levelünkre, akkor a válaszcímeknek léteznie kell mindenképpen. Ez pedig vagy azt jelenti, hogy nem is teljesen egyszer használatos a címünk, hanem csak gyakran cserélt, ami miatt rengeteg e-mail címet kell ellenőriznünk és karban tartanunk; vagy tényleg egyszer használatos, ekkor viszont az elküldött leveleinkre jövő válaszra nem számíthatunk.

3.1.2. Szerver konfigurálással

Megoldás az is, ha a szervert úgy konfiguráljuk, hogy fogadjon el minden e-mailt és ne jelezzen vissza róla senkinek, a téves leveleket pedig egyszerűen eldobjuk. A megoldás több okból is problémás: a levélküldők nem tudják meg, hogy a cím nem létezik, és elárasztathatják a szervert téves levelekkel. Fontos az is, hogy a legitim felhasználók sem kapnak visszajelzést a tévesen címzett levelekről. Mindezek miatt a visszajelzés letiltása nem javasolható.
A legmegfelelőbb természetesen az SMTP protokoll finomítása lenne, de mit tudunk addig is tenni, amíg ez nem következik be?

3.2. Új program elemet igénylő módszerek

Ebben az esetben már valamilyen aktív komponens kerül az eddig használt levelező-rendszer mellé. Két eltérő megoldást lehet alkalmazni, illetve ezek együttesét, növelve egymás hatékonyságát.

3.2.1. Egyénileg védekező rendszer

Az egyik megoldás az egyénileg védekező rendszerek. Ekkor minden résztvevőnek van egy saját önműködő rendszere, amely a döntéseit egyéb rendszerektől függetlenül hozza. A támadás szűrését a levéltovábbítás során keletkező hibaüzenetek alapján lehet elvégezni.
Ha a támadó DHA támadás során levelet küld, akkor téves címzettnek küld e-mailt egy adott IP címről, majd később újra fog próbálkozni ugyanarról az IP címről másik tévesen címzett levéllel. Elosztott DHA esetén is általában több e-mail címet próbál ki a támadó ugyanazon IP-címről még mielőtt IP-címet váltana. Azonban van olyanra is példa, hogy nem küldenek sok levelet egy címről. Ez valószínűleg attól függ, hogy észreveszik-e, hogy kitiltjuk támadás detektálása esetén az adott IP-címet, illetve, hogy mennyire fontos a támadónak a cím.

3.2.2. Hálózaton alapuló védelem

Másik lehetséges védelmi mechanizmus a hálózaton alapuló védelem. Ekkor a rendszer a hálózat egyéb résztvevőivel együttműködve próbál védekezni a DHA támadás ellen.
Ha egy támadó egy ismeretlen címre küld egy e-mailt a megtámadott szerveren, a megtámadott szerver küld egy hiba jelentést a központi szervernek. Ez a hiba jelentés tartalmazza a támadó IP-címét, a kipróbált e-mail címet, és a támadás idejét. A központi szerver gyűjti ezen jelentéseket, és ha túllép egy küszöböt ezen IP-ről jövő próbálkozások száma, akkor behelyezi a támadó IP-címét a fekete-listára. A szerver a listára kerülés után is jegyzi a támadó kísérleteit, így nem hagyja elévülni a bejegyzést. A feketelista tartalmát le lehet kérdezni a szervertől, ami a feltett kérdésre, hogy egy e-mail fekete-listás-e vagy sem, egy igen-nem választ ad.

---

[5] Általában egy, vagy több hálózati csatlakozással bíró, valamilyen sebezhető operációs rendszert és szolgáltatást emuláló rendszer. A támadók könnyű célpontnak vélik és felfedik ezáltal magukat és szándékukat, így tőlük már az éles rendszer védhetővé válik.

4. A javasolt rendszer működésének leírása

A javasolt rendszer felépítése a következő: egy rendszernapló elemzőből, és egy ennek eredményét később felhasználó front-end modulból áll. Az eredményeket központi nyilvántartásban összegezzük, azaz nyilvántartjuk azokat a gépeket, amelyek DHA támadásban érintettek. Ezt a feketelistát [2] szerint csoportosíthatjuk: IP-címeket tárolunk, és passzív monitorozást végzünk.
Ha DHA támadó küld levelet a rendszernek, akkor a működés a következő: a támadó első levele átsiklik az ellenőrzésen, amiatt, mert az IP-címe még nem került be a szerver adatbázisába, még nem volt olyan résztvevő, aki támadást jelentett volna erről a címről. A levél továbbmegy a levelező-szerverbe, ami egyfelől ellenőrzi, hogy kézbesíteni tudja-e a helyi postafiókokba a levelet, ami támadás esetén mivel sikertelen, bekerül a jelentés róla a rendszernaplóba. A rendszernapló elemző rendszer az e-mail kiszolgáló jelentéseiből megnézi, hogy a téves címzéssel rendelkező e-mailek honnan jönnek hozzánk (milyen IP-címről), és ezekről részletes jelentést tesz a központi adatbázisnak.
A rendszer a DNS protokollt használja a lekérdezésekre és jelentés küldésre a védett szerverek és a RBL-szerver között. Azért esett a választás a DNS protokollra, mert előnyei közé tartozik a robosztusság, a cache-mechanizmusa a DNS szervereknek (ha egy kérés fennakad valahol, akkor sem veszik el jó ideig és a szerver terheltségét is lehet csökkenteni ezen cache-mechanizmus által, mivel így akár burstökben is ki lehet szolgálni a kéréseket), illetve a tűzfal konfigurációkon is átjut ez a mechanizmus, nem igényel újabb portokat. (A DNS teljesítményének részletes vizsgálatát [3]-ban lehet megtalálni.)
Csökkenthető azon IP-címek támadó adatbázisba kerülésének esélye akik egyszer-egyszer csak véletlen elgépelik a címet azáltal, hogy a központi adatbázisban nem kerülnek be egyből a bejelentett IP-címek a támadók közé, hanem előtte az előző bejelentéseket is alapul véve a bejelentési időközök gyakoriságát kiszámolja a szerver. Ha ez egy bizonyos értéket átlép, akkor teszi át a bejelentett IP-címet a támadók listájára.

5. A javasolt megoldás implementációja

A rendszer prototípusát a következő környezetben hoztuk létre: linux rendszert használunk, standard levelezést lebonyolító megoldásokkal. Új levél érkezése esetén az *inetd* démon működésbe hozza a levelező szervert. Ez hagyományosan a 25-ös portra érkező kérésekre figyel, és elindít hozzá egy MTA-t (sendmail, postfix, exim, stb.). Ám a mi esetünkben, nem közvetlenül adjuk át a kérést a levelező szervernek, hanem egyik modulunkon keresztül átvezetjük a kérést. Ez a modul felelős a DHA támadások kivédéséért, azaz az ismert támadók kitiltásáért. A DoS frontend (bővebben lásd. [4]) előbb ellenőrzi statisztikai módszerekkel, hogy az adott IP-ről nem hajtanak-e végre DoS támadást a levelező szerver ellen, és ha ezen a szűrésen átment az IP, akkor kerül sor a DHA támadással kapcsolatos ellenőrzésre. A DoS frontend modul ha DoS támadást érzékel akkor úgy viselkedik, hogy eldobja az adott támadó felől jövő TCP kapcsolatokat. Ez a DHA támadó esetén is megfelelő működés, így a DHA támadó felől jövő levél nem megy tovább a levelező rendszer felé, tehát nem kerül kézbesítésre ami miatt nem keletkezik újabb bejelenteni való a szerver felé.
A szűrésen átment TCP kapcsolatot továbbadjuk a levelező szervernek, ami a teszt környezetekben volt *sendmail* és *exim4*-es levelező szerver is. A rendszernapló elemző rendszer a linux *syslog* file-jában keresi a levelező rendszer által generált jelentéseket. A különböző levelező szervereknek a rendszernapló bejegyzései eltérőek, így más-más bejegyzések vizsgálatára is fel kellett készíteni az elemző modult. Ezt egy külső konfigurációs fájlban lehet beállítani a modul regisztrálásakor használt azonosító és titkos kóddal együtt. (Ugyanerre szükség van a DoS front-end esetén is, ami szintén konfigurációs fájllal állítható be.) Az RBL-adatbázis *MySQL*-ben lett megvalósítva. Az adminisztrációs felület Apache és PHP futtató környezetet igényel. A rendszernapló elemző modult és a RBL-szervert Perl-ben lett implementálva. A rendszer működés közben is megtekinthető [16]-on, illetve a kliens és szerver program is letölthető.
A rendszer nagy előnye hogy a központi nyilvántartás segítségével a komponenseinket használó összes résztvevő profitál egymás bajából is, azaz egy támadó nemcsak egy helyen lesz kitiltható, de másoknak sem fog tudni károkat okozni. A rendszer kliens oldalának megvalósítása komponens-alapú, aminek több előnyös következménye is van:

- a támadók bejelentési és a tiltási mechanizmusa különválasztható, (az *ados* frontend segítségével)
- egy már meglevő rendszer is kiegészíthető vele, illetve akár csak bizonyos komponenseivel, így növelve a meglévő hatékonyságát is. Ezzel szemben más megoldások esetében az egész rendszert meg kell vásárolni és egészében át kell térni az új rendszer használatára, ha a DHA ellen védelmet akarunk kapni.
- a komponensek transzparensek kívülről, így a kiesésük esetén nem teszik a rendszert használhatatlanná
- a DHA komponenssel együttműködhetnek vírus és spamszűrő modulok is: a DHA támadás alapja a haszon. Ha valaki egyetlen levéllel esetleg "feldobja" a zombie-ját amiről DHA-t csinál, akkor később spamet sem fog tudni küldeni arról a gépről a komponensek együttműködése miatt. Mivel egy adott DHA kísérlet esélye csekély, ezért nem biztos hogy megéri a támadónak kockáztatni egy DHA miatt a zombie-ját, megéri inkább máshonnan címeket szereznie. A vírus jelentő modul, pedig megakadályozhatja, hogy ha az adott gépre

olyan vírus került, ami trójai programot telepít, vagy saját maga nyit rajta backdoor-t, akkor a támadó nem fogja tudni használni, mint zombie-t, hiszen már a támadás előtt bejelentették.

5.1. Téves riasztások kezelése

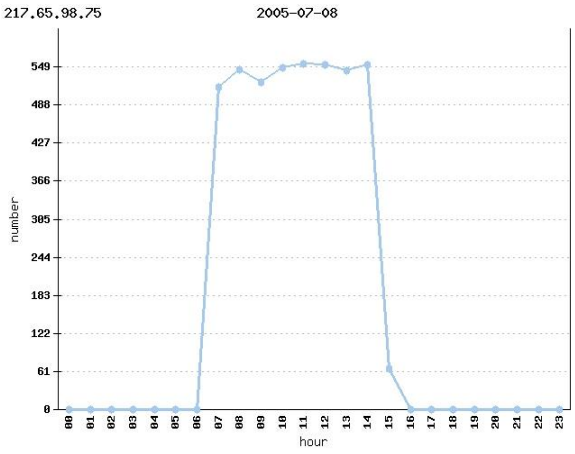
A rendszer tekinthet támadóknak olyan levél küldőket is, akik csak véletlen elgépelik a címet és így nem létező postafióknak küldik a levelet. Ezzel az a probléma, hogy a központi adatbázisba így bekerül mint támadó. A téves riasztások alacsonyan tartása érdekében a következő módszerek használhatóak, hogy az egyedi téves levelek elválaszthatóvá váljanak a valódi támadóktól: Egyrészt a központi adatbázisban az is nyilvántartható, hogy az egyes IP-címek mennyire "veszélyesek". Azaz pontozni lehet őket aszerint, hogy hány bejelentés érkezett arra az IP címre vonatkozóan. Másrészt alkalmazható öregítés (aging)[6] a központi adatbázisban. Az öregítés nagyon fontos szerepet játszik a rendszerben, ezért jól kell megválasztani a használt metódust: ugyanis, ha egy támadót eltávolítunk a listáról, akkor tovább támadhat, ha viszont túl sokáig van rajta, akkor akár a rendes felhasználók forgalmát is megakadályozhatja.

5.2. A védelem eredményessége

A központosított szűrés eredményeképpen a támadó csak egy nagyon korlátozott számú próbát tehet a védett doméneken. Mivel tipikusan támadás során időben nagy gyakorisággal küldenek e-mailt, ami jóval az alapbeállításként használt óránkénti limit alatt van, így nagyon hamar besorolódnak a támadók közé. Ezek után újabb jelentés nem érkezik, de az öregítés hatására, amikor nagyjából másfél óránként újrakalkulálódnak a támadási intenzitások, úgy a támadók ismét kikerülnek, és ha a támadást ismét folytatja egy-két levelet újból sikerül elküldenie a védett kliensnek. Ekkora ismét az újra nagy intenzitású támadástól visszakerül a támadók közé.

A rendszerünk a támadó címeit sorban feljegyzi, így azt a többi védett doménen sem tudja felhasználni támadásra (ezt egyéni védekezés esetén megteheti nyugodtan). A zombie gépeit is elveszti ezáltal, tehát a támadó költségeit megnöveli jelentősen. A támadó által kiküldendő e-mailek számát is megnöveli, mivel az nem tudja eldönteni, hogy csak késleltetődik a válasz és helyes a címzett, tehát folytatni érdemes a támadást, vagy hogy eldobáljuk a leveleit. A nyeresége pedig minimális lesz, mivel a lehetséges próbálkozásiból, amíg fel nem kerül az RBL-listára, majdnem biztos, hogy nem fog hozzájutni érvényes felhasználó névhez, azután, pedig a leveleit megszűrjük. Természetesen a rendszerünk nem nyújt védelmet a nem védett doméneknek, így azokon korlátlanul próbálkozhat a támadó. Tehát a védelem csökkenti a támadó nyereségét, gazdaságtalanná téve a DHA támadást.

6. A támadók csoportosítása támadási intenzitás alapján



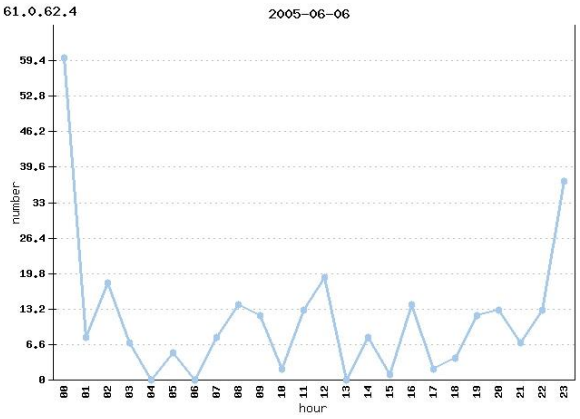
0. ábra Egy tudatos támadó napi statisztikája

A DHA támadókról feltételezzük, hogy megkülönböztethetőek. Egyik ilyen megkülönböztető jegyük a támadási statisztikájuk. Ha megvizsgáljuk a valós rendszerekből gyűjtött támadási statisztikákat, tipikus támadó modelleket lehet kialakítani. Az egyik tipikus támadó modell a *tudatos támadók*, akiknek a viselkedésén látszik, hogy nem véletlenül küldenek néha-néha egy-egy levelet, hanem masszívan DHA támadnak. Főleg napközben aktívak (reggel 7- este 11 között), éjjel a támadás is abbamarad. Az időpont alapján az tűnik a legvalószínűbbnek, hogy mikor dolgozni mennek, akkor otthoni számítógépüket bekapcsolva hagyják, ami egész nap támad így. Hétvégén a támadások általában szünetelnek. Ekkor az internet elérését nyílván másra is használják. Jellemző rájuk egy állandó levél küldési sebesség, mivel a sávszélességüknek egy fix hányadát használják a támadásra. Ez általában nem nagyon ingadozik, félgázzal sohasem támadnak, ha más dolguk van, akkor teljesen leállnak és akkor kezdik újra, amikor megint meg van hozzá az erőforrásuk.

A *vírusokkal fertőzött, így a vírus által támadó gépek* és a trójaiakon keresztül távirányított gépek levél küldési sebessége nagyon ingadozó. Ennek egyik oka, hogy nagyon elterjedtek és hatalmas mennyiségű zombie-gép áll a támadók rendelkezésére, amik sávszélességei elég változóak. A felhasználók többsége (világ viszonylatban) manapság még mindig betárcsázós, perc vagy forgalomdíjas előfizetéssel rendelkezik. Egy-egy ilyen kliens levélküldési sebessége 1-2 levél/óra mindössze, viszont nagy létszámukból kifolyólag bármelyik szélessávú Internet eléréssel rendelkező DHA támadóval felveszik a versenyt. A másik ok a levélküldési sebesség

[6] Az öregítés az adat idővel való eltávolítása

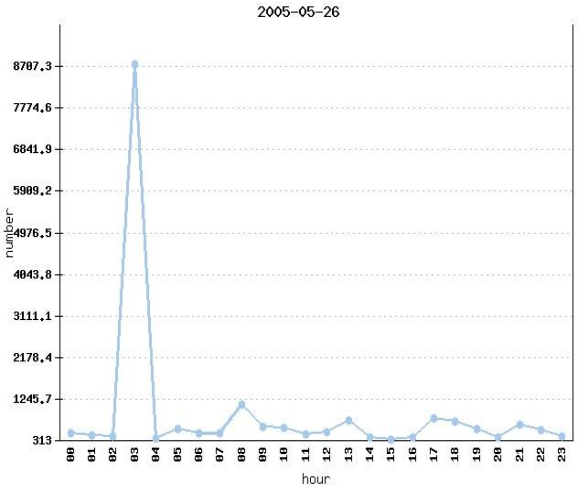
ingadozásának, a cél, hogy jelenlétük rejtve maradjon a tulajdonos gépén, így ne zavarja a rendes munkája során a felhasználót, hanem csak üresjárati erőforrásokat használjon. Egy feltehetőleg vírussal fertőzött gép napi támadási statisztikája van ábrázolva az 2. ábrán, ahol jól látszik az ingadozó levél küldési sebesség.



2. ábra vírussal fertőzött gép támadásának napi statisztikája

A *trójain keresztül távirányított gépek* is főleg napközben aktívak. A trójai program kliensével rá kell csatlakozni a célgépre korábban sikeresen telepített trójai szerverre. Ehhez a támadó felhasználónévvel és jelszóval azonosítja magát, majd a bejelentkezés után használhatja támadásra. Mindegyik zombie gépről 10-20 levelet küldenek óránként majd kijelentkeznek, hogy ne legyen feltűnő az aktivitásuk. A kijelentkezés után egy másik zombie-t keresnek fel, ahol kezdődik ugyanez elölről. A trójaiak *professzionális felhasználása* a botnetek használata. A támadóknak lehetőségük lett arra, hogy ha egy vírussal eljuttatnak egy bot-ot egy megtámadott gépre, és azt ott elindítják, akkor az bejelentkezik egy chat-szerverre, csinál egy új csatornát, és elkezdi várni türelmesen egy társalgó partner érkezését aki megfelelő parancsok továbbításával a chat-szerver felé értelmezhető utasításokat tud adni ezen bot-oknak. A chat-szerver működéséből adódóan ezt nem csak egy bot teheti meg, hanem több ezer is, akik más és más fertőzött számítógépről jelentkeznek be a chat-szerverre. Ez a támadási módszer lehetővé teszi a támadóknak, hogy egyszerre ne csak egy trójait irányítsanak, hanem rengetegnek adjanak parancsot. Távirányított DDHA-ra jó példa a rendszerünket ért egyik nagy arányú támadás, ami a 3. ábrán látható. Az azt megelőző, illetve rákövetkező órákban átlagosan 1000 támadó szándékú levél érkezett, míg 3 órakor hirtelen majdnem 9000!

A fenti állításokat abból tudtuk leszűrni, hogy volt egy olyan rendszer, amit csak bejelentőnek használtunk, nem alkalmaztuk rajta a védelmet.



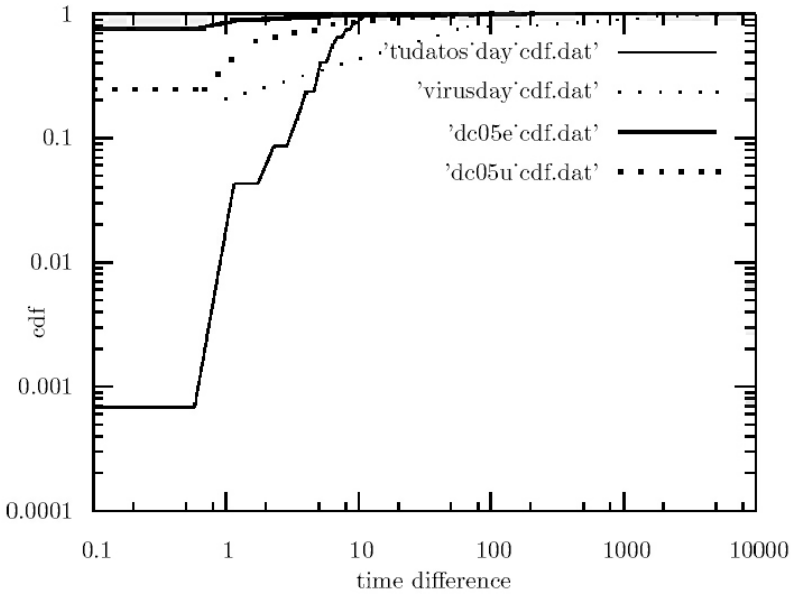
3. **ábra DDHA eredménye egyik védett rendszerünk ellen**

7. *A támadók csoportjainak statisztikai vizsgálata*

Statisztikai vizsgálatnál jó kiindulási pont az adatsor kumulatív eloszlás függvényének vizsgálata, hogy képet kapjunk az adatsor tulajdonságairól. A 4. ábrán a 6. pontban ismertetett támadó csoportok által generált támadási statisztikák cdf-je van ábrázolva. A 'tudatos_day' adatsor egy tudatos támadó napi statisztikájának adatsora (0. ábra), a 'virusday' adatsor egy valószínűleg vírussal fertőzött gép napi támadási statisztikájának adatsora (2. ábr), a 'dc05e' az egyik védett rendszerünket ért támadások aggregált napi statisztikája (3. ábra) az 5 órát megelőző időszakban, a 'dc05u' pedig az 5 óra után időszak adatsora. (A trójain keresztül távirányított gépek támadási statisztikáinak vizsgálata azért nem szerepel itt, mert nagyon rövidek voltak az adatsorok és nem lehetett statisztikai vizsgálatra felhasználni.) Az `y` tengely logaritmikus ábrázolása azért indokolt, mivel az intenzív támadási időszakok rövid támadási időközök tartományából nem sokat látnánk lineáris skálán, az `y` skála logaritmikussá tételére pedig az `x` miatt került sor, hiszen viszonylag kevés támadás jött ritkásan, így azoknak a valószínűsége lineáris skálán 0 körüli lenne, ami megint csak nem sokat mutatna a konkrét értékekből. Látható, hogy a tudatos támadó esetében a támadások alig 10% van 10 másodpercenként ritkábban. A vírussal fertőzött gép esetén a nem túl meredek cdf mutatja, hogy nagyon sok fajta támadási időköz előfordul. A védett rendszer cdf-én mindkét időszakra azt kapjuk, hogy az állandó támadás miatt nagyon kis időközönként érkeztek támadó levelek.

Egyik érdekes statisztikai tulajdonság, ami főleg a támadók modellezésénél lehet érdekes a nehézfarkúság tulajdonsága. Például a nagysebességű telekommunikációs forgalommal kapcsolatban lehetett megfigyelni a nehézfarkúság tulajdonságát, többek között a web szerverek esetében a csomagméret eloszlását találták nehézfarkúnak. Intuitívan a nehézfarkúság úgy fogalmazható meg, hogy a nehézfarkú véletlen változókat 1 nagy minta dominálja, a többi minta pedig elhanyagolhatóan kicsi ehhez az egy mintához képest az adathalmaz nagy

értékeit tekintve. (Részletesen lásd. [8]). Helyhiány miatt ezt a tulajdonságot és a vizsgálati módszereket nem részletezem, de a kapott eredmény arra utalt, hogy a támadási intenzitásokra nem jellemző a nehézfarkúság.

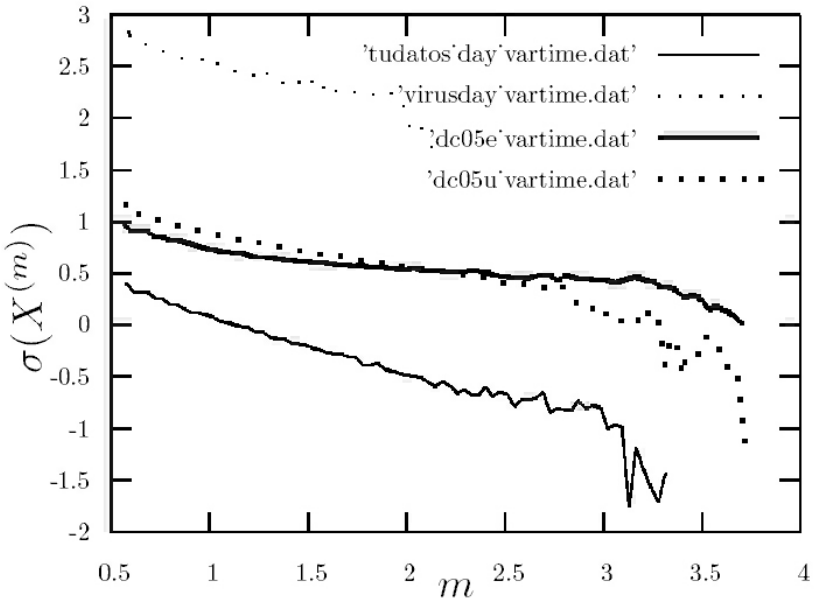


4. ábra A támadók adatsorainak kumulatív eloszlás függvényei

Több cikk is megjelent az ethernet csomagok érkezési időköz eloszlásával kapcsolatban. Némely statisztikai jellemzőjük eltérő volt a Poisson-folyamatnál megszokottaktól. Ha a kapott csomagok számát kirajzoljuk egy adott időtartományra az idő függvényében, majd ezt megismételjük különböző időskálákat választva, akkor a kapott diagramok hasonlóak lesznek, akármekkora periódust veszünk is alapul. Ugyanez igaz az adatcsomagok hosszára is adott időintervallumokat tekintve. Ez a fraktálokhoz hasonló viselkedés, azaz mindig ugyanolyan mintákat lehet látni az időtől függetlenül. Azaz a burstösség különböző időskáláknál megjelenik; a burstöknek nincs egy természetes, átlagos mérete; és a Poisson-forgalommal ellentétben nem lesz egyenletesebb a forgalom, ha aggregáljuk. Ez a viselkedés a hosszútávú összefüggőség.

A hosszú távú összefüggőség meghatározásának egyik legjellemzőbb módja a Hurst-paraméter értékének becslése, más néven az önhasonlóság fokának megállapítása. Erre az első bemutatott módszer a *Variancia-idő diagram* számítása. A variancia-idő diagram számításánál és rajzolásánál, arra a tényre alapozunk, hogy az önhasonló idő-sorozatokat szórását ábrázolva egyre több csomagot aggregálva azt tapasztalhatjuk, hogy egy exponenciális görbére illeszkednek az így kapott értékek, amely kitevőjének az ellentettjéből a Hurst-paraméter meghatározható. (Formálisan: $\sigma\left(X^{(m)}\right) \sim m^{-\beta}$, ahogy $m \to \infty \Rightarrow 0 < \beta < 1$)

A variancia-idő diagramon a $\log\left(\sigma\left(X^{(m)}\right)\right)$ ábrázoljuk a $\log(m)$ függvényében. Pszeudo önhasonló idősorozatokra a kapott egyenes meredeksége $-\beta$, ahol $-\beta > -1$. A Hurst-paraméter pedig könnyedén kiszámítható: $H = 1 - \frac{\beta}{2}$. Egy folyamatot akkor mondunk pszeudo önhasonlónak, ha az így kapott H érték 0,5 és 1 között van. Ahogy egyre nagyobb csoportokat képzünk úgy csökken a szórása ezen újonnan képzett adathalmazoknak. A logaritmikus skálán rajzolt értékekre egy egyenest lehet illeszteni, ennek az egyenesnek a meredeksége a keresett β paraméter. A nem önhasonló adathalmazokra ennek az egyenesnek a meredeksége -1 körül van, az önhasonló adathalmazokra egy sokkal kevésbé meredek egyenest kapunk. A β paraméterből számolt Hurst-paraméter értéke hosszútávon összefüggő adathalmazok esetében 0.5 és 1 közötti érték.



5. ábra A támadók adatsorainak variancia-idő diagramjai

Az 5. ábrán látható, hogy a hosszú távú összefüggőség

tulajdonságával szinte mindegyik adatsor rendelkezik (a vírusos adatsorra a legkevésbé érvényes ez), mivel a görbére illeszthető egyenesek meredeksége beleesik a -0,5 és -1 közötti intervallumban. A legérdekesebb a védett rendszer aggregált statisztikája, amelynek meredeksége nagyon kicsi, így a Hurst paraméter, azaz az önhasonlóság foka nagyon magas. Ebből azt a következtetést lehet levonni, hogy a adatsorban csak a szerver burstössége jelenik meg, a támadóké nem. Azaz pl. csak a szerver terheltsége volt a bottleneck a rendszerben, ami a bustösséget okozta, nem a támadók saját rendszerei.
Ezt a tapasztalatot felhasználhatnánk egyik állandó probléma megközelítése: a vállalati proxyk és e-mail szervereket üzemeltető vállalatok mögött levő fertőzött kliensek miatt, mivel forráscímként a levelező-szerver IP-címe jelenik meg a rendszernaplóban, így a szerver kitiltódik, de ezzel az általa karbantartott nem támadó felhasználóktól sem fogunk levelet elfogadni. Az adatsorok hosszú távú összefüggőségének vizsgálatával képet kaphatunk az adott IP-cím mögött megbúvó támadókról:

- Ha az adatsor erősen összefüggő, azaz a szerver burstössége mellett nem jelenik meg másik időskálán a burstösség, akkor feltételezhető, hogy csak a szerver fertőzött.
- Gyengén összefüggőség azt jelenti, hogy sok fertőzött kliens van a szerver mögött, ráadásul különböző időskálákon okoznak burstösséget, azaz a különböző támadóknak különböző erőforrások is állnak rendelkezésére.

8. Összefoglaló

A cikkben megvizsgáltuk a DHA támadásokat. A DHA támadás egy brute-force jellegű támadás egy levelező-szerver által karbantartott e-mail címek kinyeréséért. Bemutattuk a lehetséges védekezési technikákat. A javasolt hálózaton alapuló megoldásnál a rendszer a hálózat egyéb résztvevőivel együttműködve próbál védekezni a DHA támadás ellen. Ez tipikusan egy központi szerver által karbantartott RBL-lista kérdezésével és feltöltésével működik. Ezek figyelembevételével bemutattuk az általunk kialakított komponensekből felépülő védelmi mechanizmust, és elemeztük ezt. A rendszerünk felépítése a következő: egyrészt áll egy rendszernapló elemzőből, és egy szűrő komponensből a kliens oldalon. Másik része egy szerver alkalmazás, ami a kliens által szolgáltatott eredményeket központi nyilvántartásban összegzi, azaz nyilvántartja azokat a gépeket, amelyek DHA támadásban érintettek. A rendszer által szolgáltatott adatokat alapul véve több szempontból is csoportosítottuk a támadókat és a fenyegetettségeket amit jelentenek a levelező szerverekre, ezek statisztikai tulajdonságait pedig megvizsgáltuk.

Irodalomjegyzék

1. J. Klensin: *Simple Mail Transfer Protocol* 2001. RFC 2821
2. Jaeyon Jung, Emil Sit: *An Emprical Study of Spam Traffic and the Use of DNS Black Lists* 2004.
3. Jaeyon Jung, Emil Sit, Hari Balakrishnan, Robert Morris: *DNS performance and the effectiveness of caching* IEEE/ACM Transactions on Networking, 10(5), October 2002.
4. Bencsáth Boldizsár, Vajda István: *Ados frontend* /2005. január/
5. András Horváth, Miklós Telek: *Markovian modeling of real data traffic: Heuristic phase type and MAP fitting of heavy tailed and fractal like samples*
6. Distributed Sender Blackhole List, http://dsbl.org
7. Mailinator, http://www.mailinator.com/mailinator/index.jsp
8. The Apache SpamAssassin Project, http://spamassassin.apache.org/
9. Clam AntiVirus, http://www.clamav.net/
10. Vipul's Razor, http://razor.sourceforge.net/
11. Distributed Checksum Clearinghouse, http://www.rhyolite.com/anti-spam/dcc/
12. Styx Mail Filter - vállalati levelező szerverek védelmére kifejlesztett integrált hardver- és szoftver megoldás, http://www.albacomp.hu/sajtokozlemeny.asp?szam=25 /2005.január/
13. eSafe Advanced Anti-spam Software, ftp://ftp.ealaddin.com/pub/Marketing/eSafe/White_paper%/WP_eSafe_Anti_Spam/esafe_antispam_whitepaper.pdf
14. Kerio MailServer, http://www.kerio.com/kms_antispam.html
15. Secluda Inboxmaster, http://press.arrivenet.com/tec/article.php/308157.html
16. VIRUSFLAGS (A rendszerem működő prototípusa), http://www.virusflags.org