Elosztott behatolásérzékelő rendszer lehetőségei, alkalmazása

Gyimesi Judit
V. műszaki informatika

Konzulensek:
Dr. Fehér Gábor, adjunktus, BME-TMIT
Korn András, doktorandusz, BME-TMIT

Networkshop 2005

1 BEVEZETŐ

Cikkemben az elosztott, anomália-detektáló behatolás-érzékelő rendszereket vizsgálom, mint egy lehetséges megoldást a hálózati férgek terjedésének felismerésére. Megemlítek több alkalmazást is, ahol sikeresen működne, mint például a portscan-ek kiszűrése, és az elosztott erőforrás kimerítő támadások (Distributed Denial of Service – DDoS) ellen. Részletesen a féregtámadásokra koncentrálok, ajánlva egy algoritmust, mely – bár teljesen nem képes megakadályozni a férgek terjedését – mégis jelentős eredményeket tud felmutatni. Ennek az algoritmusnak a korlátait, és lehetőségeit analízis keretében mutatom be.

1.1 *Behatolásérzékelő rendszerek (Intrusion Detection Systems – IDS)*

Napjainkban egyre nagyobb hangsúlyt fektetnek a hálózati biztonságra. Egyre több érzékeny adat kerül fel az Internetre, ezenkívül egyre több szolgáltatónak anyagi érdeke, hogy szerverei zavartalan működést tudjanak biztosítani. Ez utóbbi megakadályozására születtek például az erőforrás-kimerítő (Denial of Service – DoS) támadások, amellyel a fenyegetettség lényegesen más jelleget öltött. A védelemre már nem elég egy tűzfal. Olyan eszközre van szükség, amely gyorsan, és a pillanatnyi körülményektől függően tud alkalmazkodni a védelmi igényekhez, akár egy intelligens berendezés formájában, mely méltó ellenfele a felkészült támadóknak.

A behatolás-érzékelő rendszerek (Intrusion Detection Systems – IDS) olyan szoftveres, vagy hardveres rendszerek, melyek automatizálják a hálózatban vagy rendszerben levő események monitorozását, gyanús, támadásra utaló jeleket keresve. Feladatuk a már elkezdett behatolás, támadás felismerése. Más hálózatbiztonsági eszközök kiegészítéseként használják, jellemzően tűzfalakkal együtt. A tűzfalak blokkolni tudnak kimenő, illetve bejövő forgalmat, a behatolás-érzékelő rendszerek viszont valós időben észlelni tudnak egy támadást, és ellenlépésként utasíthatják a tűzfalat egy új szabály felvételére, bizonyos forgalom tiltására. A mai biztonsági rendszerek csak felismerik a támadást, és saját védelmükben intézkednek, esetleg tájékoztatják a hatóságot.

A támadásérzékelőlet két fő csoportba oszthatjuk a detektálási algoritmus szerint. Az első a rendellenesség (anomália) alapú (anomaly-based) felismerés, amely a felhasználók viselkedését monitorozza, és a jellemzőit valamilyen módon tárolja (pl. log). Ezáltal ismeri azok normális viselkedését, az ettől lényegesen eltérő viselkedésminta gyanús. Ilyen lehet a rendszer használata a normális időn kívül, abnormális gyakoriságú használat, abnormális mennyiségű adatok használata, abnormális minták a program-, vagy adathozzáférésben, eszközhasználatban. Az anomália detekciót hálózat szinten általában csomagszűréssel, a protokoll fejlécek alapján végzik. Statisztikai módszereket, vagy neurális hálózatot alkalmaznak a leggyakrabban. A módszer azon a feltételezésen alapul, hogy a támadás során normálistól eltérő viselkedés is tapasztalható. Előnye, hogy nagy rendszernél is autonóm működés valósítható meg, valamint képes lehet még ismeretlen, új támadások felismerésére is. Ezért ismét az érdeklődés középpontjába került [1]. Hátránya viszont, hogy kijátszható: lassan felveszünk plusz tulajdonságokat, hozzászoktatjuk a rendszert, így az nem lesz rendellenes, amikor támadásra használjuk. Ezért ajánlatos nem magában használni, hanem például szabály-alapú IDS-sel együtt.

A másik lehetséges módszer a szabály alapú (rule-based) detekció. Előzetes tudással dolgoznak, felderítéskor már ismert támadások szignatúráit nézi. Lehet alapértelmezésben

elfogadó, tehát amire nincs szabálya, azt nem tekinti támadásnak. Viszont ha a forgalom figyelése során olyan részeseményekből álló viselkedésmintát vesz észre, amely a szabálytárában, mint támadási szignatúra megtalálható, akkor ellenlépéseket tesz. Ennek a detektálási módszernek az előnye, hogy kevés hibás riasztást produkál. Igen hatásosak, ezért szívesen használják kereskedelmi rendszerekben. Hátránya viszont, hogy a felderítői szignatúra elkészítése bonyolult, kézzel megadandó, valamint – és ez a legnagyobb hibája, - nem képes újszerű támadásokat detektálni. Létezik ugyan olyan fajtája, ahol éppen fordítva, nem a támadások szignatúráit, hanem a megengedett tulajdonságokat tárolják. Ez alapértelmezésben elutasító, hiszen minden olyan viselkedésmintát, ami nem található meg a szabályrendszerében, automatikusan gyanúsnak ítél. Ebből fakad az előnye, és hátránya is: minden más módszernél nagyobb védelmet biztosít, ám megengedhetetlenül magas a hibás jelzések aránya.

Az IDS-t két fontos szempont minősíti. Egyrészt a felismert valódi támadások és a téves riasztások aránya [2]. A téves riasztás veszélyesebb, mint elsőre hinnénk. Nemcsak kellemetlen plusz munkát okoz, hanem kihasználható gyenge pontot nyújt azáltal, hogy képes maga DoS forgalmat generálni. Ugyanis a támadó még a valódi támadás előtt működésképtelenné tudja tenni az IDS-t azáltal, hogy olyan csomagokat, csomagszekvenciákat küld, amelyekre a rendszer riaszt. Így megtelhet a log fájl, vagy egyéb erőforrások kimerülhetnek aszerint, hogy milyen feladatokat kell végrehajtania feltételezett támadás esetén, és lebénul a rendszer.

1.2 *Elosztott behatolás-érzékelő rendszerek (D-NIDS)*

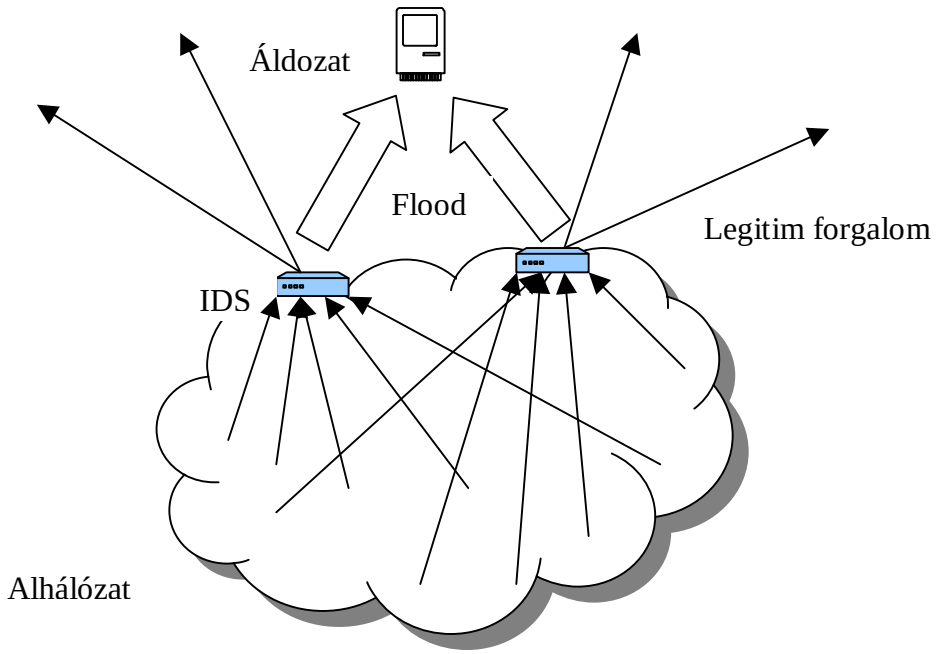
Több IDS együttműködése esetén – mint azt a következőkben bemutatom, – hatékony módszert nyerünk külön-külön nehezen vagy csak nagy bizonytalansággal felismerhető támadások ellen. A szakirodalom teszteredményei azt mutatják, hogy nemcsak a felismeréshez szükséges idő csökken le több behatolás-érézkelő eszköz együttes alkalmazása esetén – természetesen megfelelő kommunikációs algoritmus használata esetén, – hanem a téves riasztások száma is. Az IDS-ek például egy speciális, IDMEF (Intrusion Detection Message Exchange Format) nevű üzenetformátummal kommunikálhatnak [3,4,5].

2 Elosztott architektúra lehetőségei

Az elosztás egyik előnye az, hogy ugyanazt a védelmet tudjuk nyújtani egy alhálózat tagjainak akkor is, ha nem figyeljük mindet egy-egy külön IDS-sel, hanem csak a hálózat bizonyos részeire teszünk behatolás-érzékelőt, így csökkentjük a költségeket. Ezek mind látni fogják a hálózat egy részét, természetesen átfedésekkel. Ez azt fogja eredményezni, hogy egy hoszt forgalma több IDS-en is át fog menni. Eszerint lesz átfedés, vagyis a forgalomnak olyan része, amit több érzékelő is lát, illetve olyan is előfordul majd, hogy az egy hoszthoz menő forgalmat egyik IDS sem látja egészben. Hogy a teljes adatfolyamot elemezhessük minden hoszthoz, az IDS-eknek össze kell beszélniük. Érdemes az IDS-eket úgy elhelyezni, hogy a switchekhez, routerekhez kapcsolódjon egy-egy, ami a kimenő aggregált forgalmat tudja figyelni, illetve az egy egységbeli hosztok közötti forgalomra egyet.

Az elosztás másik előnye az, hogy nemcsak a bejövő forgalmat, de a kimenőt is vizsgálhatjuk. Felismerhetünk így tőlünk kimenő DDoS támadást. Ha egy IDS megnövekedett forgalmat lát egy azonos cél IP címre, azonos vagy különböző forrásokról, akkor lekérdezheti a többi kimenő vonalat figyelő IDS-t, hogy ők hány ilyen csomagot látnak. Tipikusan, ha több kimenő vonal van például az alhálózaton belüli logikai egységeknek megfelelően, úgy

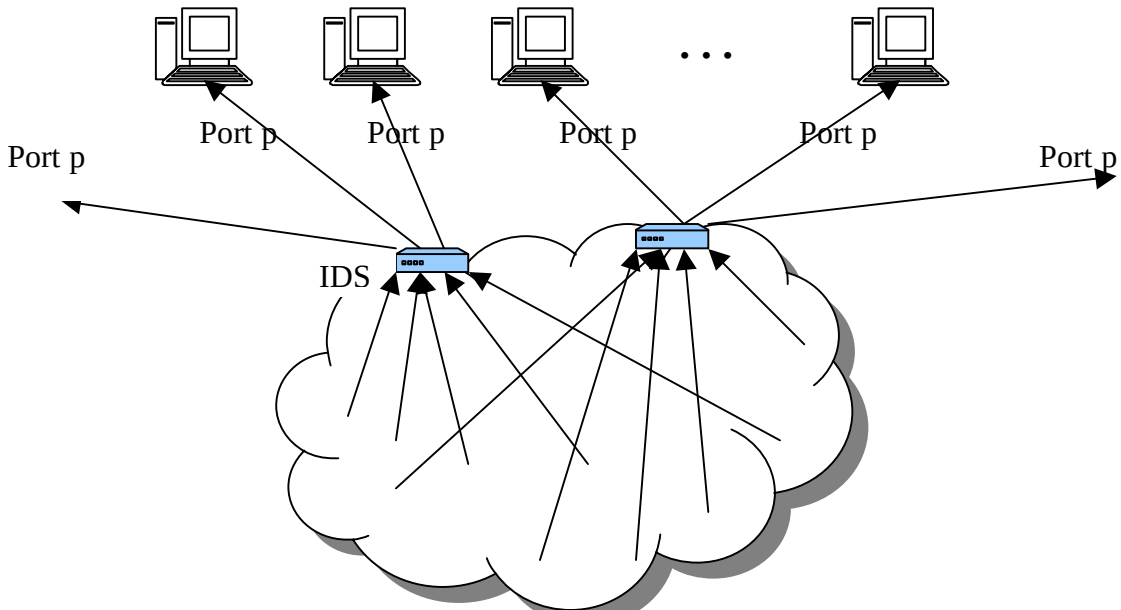
egyetlen IDS az összetett forgalmat nem lenne képes detektálni. De ugyanez, bár feltételezhetően kisebb valószínűséggel, de saját alhálózaton belüli célpontra is irányulhat.



1. ábra-Kimenő DDoS által generált forgalom

A harmadik előny a portscanek felismerése, amikor egy célcím nagyon sok portjára érzékel az IDS forgalmat. Ennek elméletét a férgeknél fejtem ki bővebben.

A negyedik előny pedig a férgek (worms) felismerése, amely alkalmazás alapelvében nem tér el az előzőtől. Itt azonban éppen fordított a helyzet: nem az azonos célcím különböző portjait nézzük, hanem a különböző célcímek azonos portjaira tartó forgalmat. ezzel foglalkozik a következő fejezet.



2. ábra-A férgek által generált forgalom

Tehát összefoglalva a költségcsökkentésen kívüli alkalmazásokat:

Támadás típusa	*Figyelt jellemzők*	*Várt megfigyelések*
(D)DoS	célcím	1 célcím
portscan	célcím, célport	1 célcím, sok célport
férgek	célcím, célport	sok célcím, 1 célport

3 A konkrét felhasználás

3.1 *Férgek (Worms)*

Az ismert férgek, mint például a Code Red, vagy a Morris, igen gyors terjedést produkáltak, akár pár óra, vagy nap elegendő ahhoz, hogy az internet minden táján elterjedjenek. Azonban hipotetikusan ennél sokkal gyorsabban is lehetne teret hódítani [6]. Egy ilyen képzeletbeli féregnek körülbelül 15 perc és egy óra közötti időre lenne csak szüksége ahhoz, hogy az összes sebezhető számítógépet megfertőzze. Ez idő alatt az embereknek nem feltétlenül sikerülhet kiépíteni a védelmet. A férgek sokkal veszélyesebbek lehetnének, mint amilyenre eddig példát láttunk. Érdemes tehát előre felkészülni még ismeretlen férgek támadására.

A legtöbb féreg úgy működik, hogy egy gépről indul, és keres sebezhető célpontokat, általában véletlenszerűen választva az IP térből. A random próbálkozással elejét veszi annak, hogy egy intelligensebb tűzfal felismerje. A keresést gyakran a saját alhálózatuknál, vagy ahhoz logikailag kapcsolódóval kezdik, mivel a sebezhető gépek általában csoportosan fordulnak elő. Egy ilyen alhálózati scan technika beépítése is felelős volt a Code Red II. sikeréért. Amennyiben talál sebezhető gépet, akkor megkísérli a fertőzést, átküldve a saját kódját. Az így létrehozott féreg-gyermek folytatja ugyanezt a tevékenységet, miközben az eredeti is.

Mi látszik ebből a hálózaton? Megpróbálni azt kiszűrni, amikor az alhálózatunk egy, vagy több gépe megfertőződik, szinte lehetetlen lenne, hiszen ez csak néhány csomag fogadását jelenti. Sávszélességet gyakorlatilag nem foglal, mivel a próbálkozás során, amikor a sebezhetőséget vizsgálja, alig néhány byte-ot kell átküldenie, és csak ha már megbizonyosodott a sebezhetőségről, utána küldi át a saját kódját, ami megint csak minimális, körülbelül 100kbyte nagyságú. Ellentétben a levelezési férgekkel (mail worm), amik minden esetben elküldik a kódjukat. Az egyetlen jelentős hálózati hatás a megnövekedett BGP (Border Gateway Protocol) kérések száma [7], amelyeket azáltal okoznak a férgek, hogy újra és újra próbálkoznak a világ összes táján levő számítógépére bejutni. A Code Red okozott némi instabilitást a periférián levő routereken, amit így ki lehet mutatni, hogy miért, arra több magyarázat is lehetséges. Azonban vannak olyan mechanizmusok, amik nem lennének ilyen hatással.

Azonban ha egy gép megfertőződött, akkor rövid időn belül kiküld sok további fertőzést. Ez már nagy, hasonló csomagokból álló forgalmat jelent, amit az elosztott IDS-ek kiszűrhetnek. Mégis, ezt az első hullámot, vagy annak legalább egy részét a felismerésig átengedik. Ez ellen nehéz védekezni, a támadás első hullámában általában nagyon gyorsan megfertőződik nagyon sok gép. Ám némely esetben a következőkben javasolt algoritmus már ebben a szakaszban is noha részlegesen, de eredményeket tud elérni. Ám leginkább a második fázis ellen véd, amikor a fertőzött gépek újra bekapcsoláskor egy újabb fertőzés lökéshullámot küldenek ki. Így az első támadás idején kikapcsolt, vagy szeparált gépeket meg lehet védeni. Jó esetben

ennél többet is tesz az algoritmus: ha a féreg terjedését megelőzi a híre, az IDS-ek közötti kommunikáció hatására, akkor már az első terjedési hullámban is eredményeket érhet el.

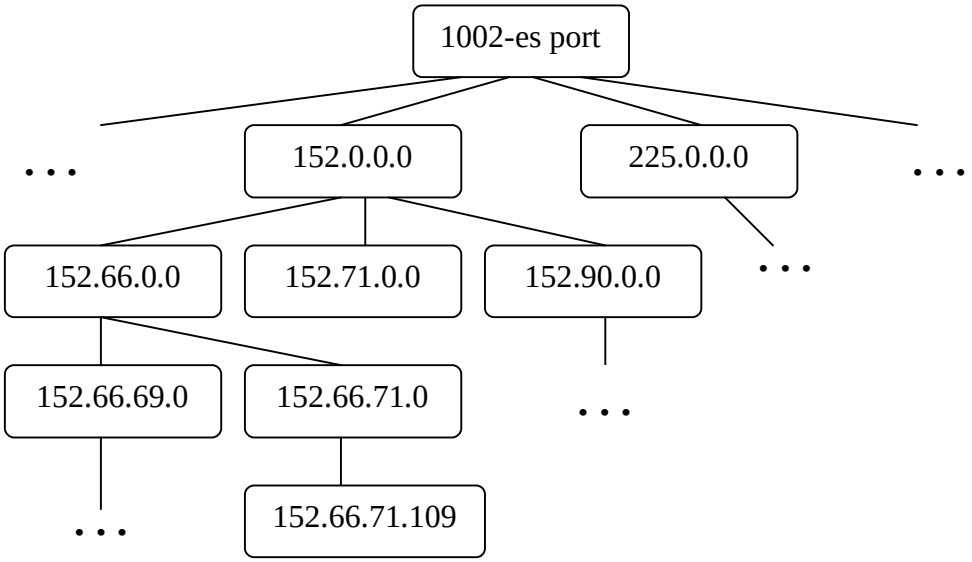
3.2 *Az algoritmus működési elve*

Az algoritmus elosztott anomália-detekciót használ. Azon alapul, hogy az alhálózat már megfertőzött gépei egyszerre, vagy akár időben némileg szeparálva, de nagy mennyiségű csomagot küldenek ki sok különböző gép azonos portjára. Az azonos portot azért feltételezhetjük, mivel a terjedéskor a férgek egy konkrét felismert sebezhetőséget használnak ki. Ha ez a jelenlegi tendencia változna is, adatbányászati módszerekkel esetleg más jellemzők meghatározása is lehetővé válna, amelyekre figyelhet az IDS. Hogy melyik gép fertőzött, azt nehéz lenne megkeresni, hiszen a forráscímek hamisíthatók, illetve a scan-elést gyakran a DDoS támadásokhoz hasonlóan több előre megfertőzött ún. zombiról hajtják végre. Egy IDS-sel nehéz lenne felismerni a férgeket, noha néhány esetben már így is sikereket érhetnénk el. Azonban a döntés biztosságát növeli, ha több IDS is függetlenül érzékel hasonlót. Így a téves riasztások illetve a fel nem ismert támadások tartományát csökkenti.

Az alhálózat forgalmának jellemzésére a csomagok fejlécét vizsgáljuk, annak is a célport, forráscím, célcím mezőit. A forráscím figyelését akár el is hagyhatjuk, mert bizonytalan lenne következtetéseket levonni belőle, hiszen az alhálózaton belül bármilyen IP cím spoof-olható. Normális viselkedési paraméterként minden célporthoz tárol a rendszer egy értéket. Ez lehet az adott célportra menő csomagok normális előfordulási aránya a forgalomban, vagy pedig egy adott időintervallum alatt normálisan előforduló ilyen csomagok darabszáma. Az első megoldást azért nem javasolnám, mivel könnyen kijátszható. Pusztán nagy forgalom generálásával helyreállítható lenne a statisztika, elrejthetők lennének a gyanús csomagok, hiszen az előfordulási arányuk nem változna. Az algoritmus diszkrét voltát ott sem úsznánk meg, mivel túl nagy számítási költség lenne minden megvizsgált csomag után frissíteni az arányt, tehát ezt bizonyos időközönként, vagy csomagmennyiség után tennénk meg. Járható lenne ez az út is azonban, ha egyidejűleg az összes forgalom mennyiségét is figyelnénk. Ez az adat egyébként is a rendszer rendelkezésére áll, hiszen egy általános anomália-detektáló eszköz ezt a vizsgálatot mindenképpen elvégzi. Mégis szemléletesebb, ha – mivel az azonos célportra menő csomagok megnövekedett darabszámának a forgalom mennyiségétől függetlenül gyanúsnak kell lennie – eleve ezt a darabszámot figyeljük. Emellett is jó azért az összes forgalmat is figyelni.

Egy csomagtípust ebben a konkrét alkalmazásban két paraméter jellemez: a célcím és a célport. A hálózat monitorozásakor az IDS minden célporthoz tárol egy listát, ami azokból a célcímekből áll, amely cím azonos portjára küldtek csomagot. Egy új csomagnál a rendszer megnézi ezt a két jellemzőt, megkeresi a csomag célportjának megfelelő listát. Ha nem létezik az adott célcím a listában, akkor felveszi azt, ha létezik, akkor nem tesz semmit. Ez utóbbit az indokolja, hogy nem gyanakodunk féregre akkor, ha csak egyetlen gépnek az azonos portjára megy sok csomag, hanem arra figyelünk, ha gyakorlatilag 1-2 csomag megy csak sok gépnek az azonos portjára. Tehát csak az a tény érdemel figyelmet, hogy hány gépre ment, az viszont, hogy egy gépre hányszor, nem. A célportokhoz tárolt listát rendezve érdemes tárolni, hogy a keresést gyorsítsuk. Kihasználva az IP címek hierarchikus jellegét, építhetünk minden célporthoz egy 4 szintű fát. A gyökér a célportot jellemzi, és egy számlálót tartalmaz az összes leveléről, illetve egy időadatot: az intervallumban elsőnek érkező csomag, ami az adott célport felé tart. Mindkét adat jelentősége később tárgyalásra kerül. Az első szint az első nyolc bit alapján osztályozza a cél IP címeket, vagyis az A osztályú alhálózatokat nézi. Majd minden A osztálybelihez tartozhat maximum 256 db B osztálybeli csomópont, amelyek már az első 16 bitre jellemzően csoportosítanak. És így tovább. Így egy legfeljebb 256-odrendű fát

kapunk, mely dinamikusan bővül a beérkező csomagoknak megfelelően. Így a keresésnél egyetlen összehasonlítással az A szinten akár 256*256 IP címet is átugorhatunk.



3. ábra-az egy porthoz tartozó célcím fa

Természetesen, hogy korlátozzuk a memória-felhasználást, bizonyos időintervallumonként a tárolt adatokat törölnünk kell. A normális adat időintervallumát is érdemes ehhez igazítani. Az intervallum végén, ha nem észlelünk anomáliát, vagyis bizonyos tűréshatáron belül a célportra jellemző normális értéket kaptuk, akkor töröljük a porthoz tartozó fát, és kezdjük elölről a vizsgálódást. Amíg egy IDS normál tevékenységet lát, azaz semekkora mértékben, vagy esetleg egy kis tűréshatáron belül nem haladja meg a tárolt modell alapján normálisnak ítélt viselkedést képviselő paraméterértéket, úgy nem kezdeményez kommunikációt. Következtethet arra, hogy mivel a többi IDS sem vette fel vele a kapcsolatot, így azokban a jellemzőkben, ami őt is érinti, azok is normál viselkedést észlelnek. Ha sokszorosan meghaladta a normális értéket, akár már az intervallum közepén is, vagyis egy előre beállított értéknél nagyobb a megfigyelt és a normális mennyiség aránya, akkor a többi IDS megfigyeléseitől függetlenül közbeavatkozik. Nem kell tehát sem az intervallum végét megvárni, sem a többi érzékelővel való kommunikáció miatti időveszteséget elszenvedni, hanem gyors reakcióra van lehetőség. E két határ között nem cselekszik azonnal az IDS, hanem kérdést intéz a többihez, aminek eredményeként összetett megfigyeléseken alapuló döntést hoz. Ez az eset szintén nem kötődik intervallumhatárhoz, hiszen ha bármikor összegyűlt a gyanússághoz elegendő csomag, akkor kiküldhet lekérdező üzeneteket, miközben tovább figyeli az érkező csomagokat. Ha később az intervallum során meghaladja az azonnali beavatkozás küszöbét is, akkor természetesen aszerint cselekszik, nem várva meg a válaszokat. Ez utóbbi két esetben, vagyis amikor valamennyivel meghaladtuk a normális értéket, az intervallum határán nem törlünk minden adatot. Azért, hogy később felhasználhassuk a detekcióban, megtartjuk a célportot, amire a megnövekedett forgalom ment, így egy későbbi támadási hullámban, akár napokkal, hetekkel később is, amikor már nem élnek a védekező szabályok, hamar újra felismerjük a férget. Ez akár kézzel törölhető is, ha utólag téves riasztásnak bizonyult a jelzés.

A kommunikáció során a gyanús viselkedést észlelő IDS elküldi a többi, a kérdésben kompetens érzékelőnek a megfigyeléseit, tehát a jellemzőt, és a hozzá tartozó mennyiséget.

Ha a többi IDS eddig nem figyelte a kérdéses tulajdonságot, akkor ezzel egyben utasítva is lett, hogy vizsgálja meg, az ő szemszögéből mi látszik. Biztonságos adatátvitelt feltételezünk, ám még ekkor is célszerű az IDS-ek üzenetváltásaira külön csatornákat kiépíteni, hiszen különben egy sávszélesség támadás (bandwidth attack) lehetetlenné tenné az elosztott architektúra működését. Miután egy IDS vagy a saját önálló megfigyelései alapján, vagy egy másik IDS megfigyeléseinek kézhezvétele után úgy dönt, hogy támadás van, és saját részéről intézkedik, ezt tudatja a többi IDS-sel is. Ha azonban az alapján konstatálja a támadást, hogy már egy másik IDS erről értesítő üzenetét vette át, tehát az hamarabb detektálta, akkor már nem kell értesítenie a többit, hiszen ezt már előtte megtették.

3.3 *Analízis*

A vizsgálatokat egyetlen portszámra végzem, ez általánosítható. Jelölje n azt a normális mennyiséget, amit a rendszer a porthoz tárol. Ez azt fejezi ki, hogy a Δt megfigyelési periódusidő alatt átlagosan hány adott célportú csomagot lát a rendszer. Δt-t úgy választjuk meg, hogy ahhoz az időhosszhoz igazodjon, amekkora intervallumonként a megfigyeléseket töröljük. Nem érdemes ezt túl kicsire választani, hiszen ezzel szétszabdaljuk a felismerést, ráadásul téves következtetésre is lehetőséget adnánk. Túl nagyra pedig azért nem jó állítani, mert túl nagy memória-szükséglettel lehetne csak fenntartani a detekciót. Jelölje az észlelt, adott portszámú forgalmat x, ami a legitim forgalom, és a féregterjedés által generált forgalom összege. Az anomália mértékét az x/n arány jelzi majd. Ha ez az arány egy $k1$ tűréshatáron belül marad, akkor legitimnek vesszük a forgalmat. A $k1$ tehát az alsó küszöbérték. Ha $k1$ és $k2$ értékek közé esik, akkor kommunikál a többi IDS-sel. Ha a $k2$-es felső küszöböt is meghaladja, akkor azonnal beavatkozik. Ezt igen magasra kell tenni, hogy ne legyen nagy a téves riasztások száma, ami – mint láttuk – az IDS-ek egyik teljesítményjellemzője. Tegyük fel, hogy a féregterjedés t idővel egy megfigyelési periódus után kezdődik. A fertőzés során rövid időn belül sok csomag kerül kiküldésre, tf idő alatt, φ darabot időintervallumonként. Az analízis során feltételezem, hogy ez egyenletesen megy végbe, ami a valóságban is jó közelítést ad. Azt is felteszem, hogy a különböző IDS-ek felé menő legitim forgalom Poisson eloszlást követ, várható értéke azonos hosszúságú intervallumokra egyenletes. Természetesen az egyes érzékelők által látott csomagmennyiség különbözik. Az egyenletességet az ronthatja el, ha több gép is fertőzéssel próbálkozik, kicsit más kezdeti időponttal. Így a rosszindulatú forgalom ezek összege lesz, ami szintén egyenletes, kivéve az átmeneti időszakokat, amikor egy új gép éppen elkezdi a terjesztést, hiszen akkor az addigi forgalom hirtelen megnő, amint az új fertőzés hozzáadódik. Tehát az eddigiek formálisan:

$$k_1 \le \frac{n+j}{n} \le k_2$$

esetén kommunikálnak az IDS-ek. Feltehetően nagyjából azonos időben éri el több különböző IDS is a $k1$ küszöbértéket, így ilyenkor megnövekedett forgalomra kell számítani.

A hálózat monitorozásakor tehát minden IDS figyeli a beérkezett azonos célportra tartó csomagokat. Nem ismer fel egyáltalán egy férget, ha a kiküldött csomagok száma periódusonként nem elegendő a $k1$ küszöb meghaladásához, vagyis:

$$j \le (k_1 - 1)n$$

Ez a helyzet azonban nem jellemző, hiszen a példa paraméterek szerint a fenti összefüggés azt jelentené, hogy kevesebb, mint a normál forgalom egy ötödét tennék ki a féregtámadásból

származó üzenetek. Ezt valóban nem reális célkitűzés felismerni, szerencsére azonban konkrét esetekben ennél jóval jobb helyzetben leszünk. Szintén nem jeleznek az IDS-ek, ha a felismeréshez szükséges idő előtt a támadás megszűnik. Nem ismeri fel a férget a terjedés kezdetének megfelelő periódusban, ha a *t*-től az intervallum végéig tartó időszakban már nem tud beérkezni megfelelő számú csomag, vagyis:

$$\frac{(\Delta t - t)}{\Delta t} j \leq (k_1 - 1)n$$

Ilyenkor azonban felismeri a támadást, ha az első összefüggés nem teljesül, de csak a második periódus alatt. Mivel a periódusnak nem kell megvárni a végét, ha időközben átlépi a *k1* küszöböt, akkor cselekedhet. Ezért összességében, mivel az intervallumhatár előtt nem érkezhetett *k1*n* db csomag, mert akkor már abban a periódusban gyanússá vált volna a forgalom, a második intervallumban pedig ha lát ennyi darabot, akkor cselekszik, így az átengedett csomagok száma maximum *2k1*n*. Ebből viszont a két periódus alatt *2n* a feltételezett legitim forgalom, így átlagban

$$2n(k_1 - 1)$$

rosszindulatú csomagot enged át az IDS, ennek férgek esetén kevesebb, mint a fele tartalmaz konkrét kódátvitellel járó fertőzést.

Hogy mennyi időt vesz igénybe a felismerés, azt úgy számolhatjuk, hogy legrosszabb esetben szükség van a kezdetin kívül egy második periódusra is, tehát, a terjedés kezdetétől az intervallumhatárig hátralevő idő, plusz ami ahhoz szükséges, hogy *k1*n* darab csomag beérkezzen.

$$t_{gyanú} = (\Delta t - t) + \frac{k_1 n}{j + n} \Delta t$$

Ez összességében kevesebb lesz, mint a második összeadandó tag kétszerese, máskülönben a tevékenység már az előző periódus alatt gyanússá vált volna.

Ez még csak a kommunikáció kezdetét jelenti, ami azonban lényeges késleltetés nélkül kell, hogy végbemenjen. Jelölje *T* azt az időt, amíg a kommunikáció lezajlik. Ezalatt tovább jönnek be a féreg által generált csomagok. Ám számuk nem haladhatja meg periódusonként a normális mennyiség *k2*-szorosát, hiszen akkor azonnal beavatkozik az IDS. Tehát az összes átengedett csomagot úgy határozhatjuk meg, ha nézzük a férget leggyorsabban meggyanító IDS-t, ami elsőként küldi ki üzeneteit a többi érzékelőnek. Ezzel kihasználjuk, hogy egy érzékelő a saját idejénél hamarabb is reagálhat egy támadásra, ha egy másik IDS adja át neki az információt, ami már hamarabb észlelte. A leggyorsabb IDS üzeneteinek kiküldése után *T* idő szükséges, amíg az utolsó IDS is utasítja a tűzfalát, tehát az összes kommunikáció lezajlik. Ez idő alatt már egyre kevesebb csomagot enged át a rendszer globálisan, ahogy a részforgalmak folyamatosan tiltódnak, így ebben is egyenletességet feltételezve, *T* ideig a csomagok felének beengedésével számolok. Tehát a leggyorsabb IDS átenged

$$\frac{(\Delta t - t)}{\Delta t} j_1 + (k_1 - 1)n_1$$

darab csomagot, ahol az indexek azt jelentik, hogy az érték az 1. IDS-re vonatkozik (A *k1*-en kívül, ott az index azt jelzi, hogy ez a gyanú alsó korlátja). Minden további IDS pedig annyit, enged be, amennyit a leggyorsabb IDS gyanússági ideje alatt lát.

$$((\Delta t - t) + \frac{k_1 n_1}{j_1 + n_1}\Delta t)\frac{j_2}{\Delta t}$$

Efölött még összesen, a T kommunikációs idő alatt átjut az IDS-eken

$$T\frac{j}{2\Delta t}$$

ahol φ az összes, terjesztő gépek által kibocsátott féreg generálta csomagszám Δt idő alatt. Ezeket az eredményeket minden IDS-re összegezve, az összes átengedett csomag:

$$\frac{(\Delta t - t)}{\Delta t} j_1 + (k_1 - 1)n_1 + \sum_{i=1}^{j}((\Delta t - t) + \frac{k_1 n_1}{j_1 + n_1}\Delta t)\frac{j_2}{\Delta t} + T\frac{j}{2\Delta t}$$

ha *j* darab IDS van a hálózatban. Vagyis:

$$\frac{(\Delta t - t)}{\Delta t} j_1 + (k_1 - 1)n_1 + ((\Delta t - t) + \frac{k_1 n_1}{j_1 + n_1}\Delta t)\frac{j}{\Delta t} + T\frac{j}{2\Delta t}$$

Feltételezhető azonban, hogy sok csomagot többször is számoltunk, mivel az elosztott architektúrában átfedések vannak az érzékelők által figyelt hálózatrészek között. Ilyenkor, mivel mindegyik a terjedés első valahány csomagját engedi át, ugyanazokról a csomagokról van szó. Egy átlagos felépítésben egyetlen üzenet 2-3 IDS-en is át fog haladni, ezért az előző eredménynek csak az arányos része reprezentál különböző csomagokat.

A konkrét számítási paraméterek esetében ez hány csomagot fog jelenteni? Legyen Δt=15 perc φ=800 csomag/900s, T=2s, és t=850s, a leggyorsabb IDS-re pedig φ1=200 csomag/900s, n1=50 csomag/900s, k11=1,2 és k21=10. A becslés szerint 64 rosszindulatú csomag fog észrevétlenül átmenni az IDS-eken.

Már ez az eredmény is biztató, és megoldást ígér a kikapcsolt és szeparált hosztok védelmére. Ha az első hullámbeli támadás hatását is csökkenteni akarjuk, természetesen a nagyobb műveletigények terhére, akkor lehetőség van további gyorsításokra. Javulhat a felismerési idő a paraméterek változtatásával. Például nagyobb átfedéssel, ez azonban többletköltséget jelent a hálózat kialakításakor. Másrészt ahogy az összesített képlet is mutatja, a periódusidő növelésével is gyorsul a detekció, valamint nagyobb intenzitású féregterjedésnél is. A küszöbérték kisebbre állításával is javulhatna a reakcióidő, ám ez túl sok téves riasztást eredményezne.

Jelentős javulást okozhat, ha figyelhetjük nemcsak a terjedést, hanem a férgek közötti esetleges kommunikációt is. Összehangolt támadások ugyanis gyakran a fertőzött gépek egymás közötti egyezkedésével valósulhatnak meg. Ilyenkor különösen jó eredményeket érhet el az architektúra olyan portok esetében, amiket normális forgalom esetén nem, vagy csak ritkán használnak. Az átengedett csomagok száma

$$j + T\frac{j}{2\Delta t}$$

ami jelen esetben, 9 IDS-sel számolva: összesen 10 csomagot jelent majd. Ez 2 átfedést feltételezve 5 csomag, ezenkívül továbbra is feltételezve a féregterjedések tulajdonságaiból, ennek csak a fele lesz tényleges kódátvitel, így csupán 2 új gép fertőződik majd meg az

alhálózatunkban. Ilyenkor nemcsak a még kikapcsolt vagy szeparált gépeket érinti a védelem, hanem a terjedés első hullámán is jelentősen gyengít.

3.4 *Értékelés*

Az előzőekben egy újfajta védekezést mutattam be a féregtámadások ellen. Körüljártam a lehetőségeket, a viselkedési jellemzőket, amire egy algoritmust alapozhatunk. Megegyeztünk, hogy a féreg alhálózatunkba való bekerülését nagyon nehéz lenne detektálni, így vizsgálódásaim a továbbfertőzés szakaszára koncentrálnak. Beszéltem az algoritmus alapelveiről, a megvalósított kommunikációról, illetve az egyes IDS-ek döntési mechanizmusairól. Végül pedig matematikai analízissel támasztottam alá a hatékonyságot.

Összességében tehát egy értékes eszközt nyertünk, amelynek azonban a férgek elleni védelem csak egy praktikus felhasználása a sok közül. Hogy milyen helyzetekben lehet előnyös használni, arra is hoztam néhány példát, a teljesség igénye nélkül.

Mint anomália-detektáló eszköz, nagy előnye, hogy új, még ismeretlen támadásokkal is hatásosan veszi fel a harcot. Kijátszhatósága miatt azonban érdemes más biztonsági eszközökkel együtt használni, mint például szabály-alapú IDS-sel.

4 Irodalomjegyzék

[1] The History and Evolution of Intrusion Detection, Guy Bruneau

[2]Characterizing the Performance of Network Intrusion Detection Sensors: Lambert Schaelicke, Thomas Slabach, Branden Moore, Curt Freeland 2003 http://www.cse.nd.edu/~spanids/papers/nids_perf_raid03.pdf

[3] IDS Interoperability and Correlation Using IDMEF and Commodity Systems: Nathan Carey, Andrew Clark, George Mohay 2002

[4] Intrusion Detection Message Exchange Requirements: M. Wood, M. Erlinger 2002

[5] Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition: D. Curry, H. Debar 2002

[6] Warhol Worms: The Potential for Very Fast Internet Plagues by Nicholas C Weaver http://www.cs.berkeley.edu/~nweaver/warhol.html

[7] Cowie, J., Ogielski, A., Premore, B., and Yuan, Y. (2001) "Global Routing Instabilities during Code Red II and Nimda Worm Propagation." http://www.renesys.com/projects/bgp_instability, September 2001.