



Budapesti Műszaki és Gazdaságtudományi Egyetem

# Új módszerek hálózati szolgáltatás-megtagadásos problémák kezelésére

Boldizsár BENCSÁTH

PhD értekezés tézisei

Témavezető:

Dr. Vajda István

Híradástechnikai Tanszék

2009

# 1. Bevezetés

Az internetes kommunikáció rengeteget fejlődött az elmúlt évek során. Az internet tervezésének kezdeti célkitűzése a külső támadók elleni védettség megteremtése volt, amit a centralizáció elkerülésével kívántak megteremteni. Nem volt viszont tervezési cél a belső, internetes támadások elleni védelem.

Napjainkban ezzel szemben az internetet állandó hadszíntérnek tekinthetjük. Folyamatosak a támadások, amelyek olyan mértékben veszélyeztetik a hálózat integritását, hogy szakértők egyre nagyobb része gondolkodik alapvető protokolloknak és magának az architektúrának az áttervezésén a veszélyek figyelembevételével.

Az internetes támadások is sokat fejlődtek az elmúlt években. Kezdetben öncélú vírusokat és elterjedt hibákat kihasználó egyedi eszközöket láthattunk, ma a leggyakoribb támadások vírusok, trójai és spyware programok, melyeket spamküldésre, információ gyűjtésre vagy éppen phishing célra használnak. Az egyedi kisstílusú támadót leváltották a számítógépek ezreit végigkereső pénzéhes támadócsoportok, céljuk a behatoláson keresztül a gépek vezérlésének megszerzése, a gépek ún. zombivá alakítása.

A szolgáltatás-megtagadásos támadások, azaz Denial-of-Service (DoS) támadások régóta ismertek. Az évek során számos típusát és esetét ismertük meg, a legegyszerűbb és legelterjedtebb típusai a programok hibáit kihasználó egyszerű támadások (pl. ping of death) vagy a sávszélességet teljesen felemésztő támadások. Az évek során a szolgáltatás-megtagadásos támadások elleni védekezés csak igen szűk körben fejlődött. Ez azért van így, mert az általános szolgáltatás-megtagadásos támadások elleni legjobb védekezés a megfelelő architektúra, a jelenlegi környezet pedig nem úgy jött létre, hogy komolyan figyelembe vette volna ezt a problémát. Adott tehát egy olyan terület, amely régóta ismert, de korántsem megoldott, sőt valódi megoldást találni a jelenlegi környezetben igen nehéz. Ezzel a területtel, azaz a jelenlegi internetes infrastruktúra melletti DoS védelmi lehetőségekkel foglalkozik kutatásom.

Számos cikk és publikáció jelent meg már felmerült szolgáltatás-megtagadásos támadások lehetőségéről. Ez jelenthetné azt, hogy ezek a támadások jelenleg igen elterjedtek, de tévednék. A szolgáltatás-megtagadásos támadások potenciálisan sokkal súlyosabb problémát jelenthetnek, mint a jelenleg megfigyelt helyzet, mert könnyen kivitelezhetőek, de nehezen lehet ellenük védekezni. Jelenleg ritkák az ilyen támadások, de bármikor megsokszorozódhat a támadások száma. Éppen ezért fontos sokkal komolyabban venni a problémát: noha jelenleg szerencsére még ritka a kifejezetten összeomlást előidéző szándékos támadás, de bármikor bekövetkezhet egy fordulat, amire nem vagyunk felkészülve.

Továbbá azt is elmondhatjuk, hogy a leggyakoribb szolgáltatás-megtagadásos támadások jelenleg észrevétlenek, a felhasználók általában az általuk használt rendszer ilyen-olyan összeomlásán keresztül veszik észre. A rendszergazdákat megkérdezve arról, hogy mi okozta az összeomlást, fel tudnak hozni egy okot, ami az összeomlást okozta, még akkor is, ha az nem a valós indok, azaz például egy DoS támadás. Megfigyelhető azonban a kapcsolat a tényleges ok és az eredmény között, de egy modern hálózati infrastruktúrában olyan sok és bonyolult eszköz kombinációja van jelen, hogy az egyes okok azonosítása nehéz. A probléma okának azonosítása (DoS támadás) néha egyike a legbonyolultabb feladatoknak, viszont az azonosítás után már a

védekezés nem is jelent gondot.

Példaként említhetjük a detekció nehézségére a kéréstelen reklámlevelek, a spamek okozta gondokat. A hálózati e-mail forgalom nagyobbik hányada ma már kéréstelen levél. Ezeket speciális célszoftverekkel szűrjük ki, mely szoftverek nagy erőforrásigényűek. Ha sok e-mail érkezik, úgy a rendszerben nagyobb az erőforrásigény, ami egyre könnyebben okoz szolgáltatás-megtagadásos jellegű problémát. A spam problémát nem egy forrás okozza és az egyes forrásoknak nem is támadás a célja. A sok elküldött levél eredménye viszont egy összeomlás, és ezen összeomlást nehéz vizsgálni a szokásos eszközökkel, mivel a küldött levelek nem valós kiszolgáláshoz kötődnek, hanem főként reklámtevékenységhez. A szokásos megbízhatósági- és teljesítménytervezés ellenben főként a valós szolgáltatási igényekkel foglalkozik, nem olyan, ettől jelentősen eltérő problémával, mint a spam kérdése.

Kutatásom során ezért olyan területekkel foglalkoztam (pl. DoS támadások modellezése, védelmi módszerek tervezése, problémák formális leírása, védelmi algoritmusok analízisa), amelyekkel eddig mostohán bántak: egyesek úgy gondolják, hogy ezek a területek megoldottak, mások egyszerűen nem ismerték még fel a problématerület fontosságát, míg sokan gondolkodtak már a megoldáson, de nem oldották meg véglegesen a problémát.

## 2. Kutatási célkitűzések

Célul tűztem ki, hogy olyan megoldásokat nyújtsak a DoS területén, amelyek előremutatóak az internet biztonságosabbá tétele tekintetében. Tevékenységemmel azt céloztam meg, hogy modelleket, tudományos igényességű módszereket, analízist, adatokat adjak a problémák kezeléséhez.

Az internetes támadások vizsgálata során kétféle szemléletmódot különböztethetünk meg. A tudományos élet a problémák modellezésén, formális módszereken keresztül, egyszerűsítések révén próbál előrelépni a problémák kezelésében. Ez a megközelítés esetenként korlátos hatókörű lehet, mivel az eredmény túlságosan leegyszerűsített lehet és gyakorlatban nehezen alkalmazható. A másik szemléletmód a gyakorlatias szemlélet, amikor tudományos igényesség nélkül, többnyire ad-hoc módszerek keretében valamilyen elképzelt terv mentén kidolgoznak egy védelmi megoldást, vagy hipotézist. Az ilyen megoldások gyakran működőképesek, mivel a gyakorlati tapasztalatokból indulnak ki, ám számos esetben nehéz általánosításuk, nem bizonyított eredményességük és nincs elemezve hatékonyságuk.

Célom az volt, hogy a két megközelítési módot vegyítve, egyszerre kezeljek gyakorlati problémákat az alkalmazhatóság figyelembe vétele mellett, mindeközben tudományos módszertant, analízist, formalizmust adjak a módszerekhez. Ez igen nehéz ezen a problématerületen, mert a támadások jelentős része nem könnyen kezelhető a jelenlegi tudományos módszerekkel, miközben bizonyítottan működőképes gyakorlati megoldások is csak egy-egy területen léteznek.

Kutatásaim során egyre jobban körvonalazódott, hogy a megoldatlan internetes problémák között a szolgáltatás-megtagadásos támadások igen különleges helyet foglalnak el. Ez a terület ugyanis összefüggésben van rengeteg más támadásfajtaival. Lehetőség szerint megoldást keresek általános protokollok védelmére, azonban nem mindig adható általános megoldás, így

figyelembe kell venni azt is, hogy mélyebb ismeretek, pl. a protokoll specifikálása mellett milyen többletinformáció áll rendelkezésre a védelem javítása érdekében.

Vizsgálataim célja tehát az volt, hogy új megoldásokat nyújtsak a DoS védelem erősítésére, miközben modelleket és módszertant is alkotok a védelem tudományos megalapozására.

### **3. Alkalmazott kutatási módszerek**

Kutatásaim során kezdetben mindig konkrét tapasztalatokat kerestem internetes biztonsági problémák tekintetében. Tudományos és ipari tevékenységem során olyan problémákkal szembesültem, amelyekről nem találtam elég információt és nem voltak kielégítő megoldások sem ellenük. Ezeket a konkrét problémákat modelleztem, illetve az ellenük tervezett védekezési módszereket analitikus módszerekkel is elemeztem (pl. az internetes biztonság területén nem elterjedt játékelméleti módszertan segítségével). Az elméleti eredményeket szimuláció segítségével támasztottam alá, hiszen ez a módszer könnyen kijelozheti az elméletben levő hibákat.

Az eredmények megfelelő rendelkezésre állása után visszatértem a gyakorlati alkalmazhatóság kérdéséhez. A javasolt megoldásokat megpróbáltam a gyakorlatban is alkalmazható prototípusok segítségével alátámasztani, alkalmazhatóságukat bizonyítani.

### **4. A szolgáltatás-megtagadásos problémák definíciója**

A szolgáltatás-megtagadásos állapotot (Denial-of-Service, azaz DoS) a következő módon definiálom: *Amennyiben a informatikai rendszer vagy a hálózati szolgáltatás működőképessége külső, nem fizikai hatásra olyan mértékben csökken, hogy az a szolgáltatást igénybe vevő kliensek számára elfogadhatatlan, legyenek a kliensek valós személyek vagy akár szoftverkomponensek, úgy a rendszer szolgáltatás-megtagadásos állapotba került.*

A szolgáltatás-megtagadásos támadást (DoS támadás) pedig a fentiek alapján a következő módon tudom definiálni: *Amennyiben egy informatikai rendszer vagy hálózati szolgáltatás működését külső szereplő akaratlagosan és szándékosan szolgáltatás-megtagadásos állapotba viszi, úgy szolgáltatás-megtagadásos támadásról beszélhetünk.*

A DoS támadás mindig akaratlagos, valós támadót feltételez. Ezzel szemben a DoS állapotot előidézheti más körülmény is: Egy támadónak nem volt célja a DoS állapot elérése, viszont cselekedetei, támadása folytán a rendszer DoS állapotba kerül. Példának mondhatjuk, ha egy támadó kéretlen reklámleveleket küld, és ezzel a levelezőrendszert működésre képtelen állapotba hozza, akkor DoS valósul meg, ugyanakkor nem igazán precíz DoS támadásról beszélni. Fontos még megjegyezni, hogy a DoS támadások kérdésköre közel esik a megbízhatóság, hibátűrés témakörhöz, azonban indíttatása lényegesen eltér: itt egy adatbiztonsági, biztonságtechnikai problémáról van szó, így eszköztára, problémái jelentősen eltérnek a megbízhatóság általános növelését célzó informatikai területektől.

Disszertációm során, illetve a DoS elleni védekezés területén általában nem kritikusan fontos annak megkülönböztetése, hogy egy valós DoS támadásról, vagy egy kialakuló, nem teljesen vagy egyáltalán nem akaratlagos DoS állapot megelőzéséről van szó. A megoldások, kezelési módszerek általában mindenfajta DoS állapot megelőzését célozzák. Ahol kiemelten fontos a két terület megkülönböztetése, ott azt egyértelműen jelezem.

## 5. Új kutatási eredmények

Kutatásaimat három fő tézisbe foglaltam.

Kezdetben a szolgáltatás-megtagadásos támadások általános sémájával foglalkoztam, amelyben általános alkalmazás szintű protokollok DoS védettségét elemeztem és növeltem. Kutatásaim azt igazolták, hogy az általános megoldások hatóköre szűk, és noha egyes esetekben megoldást jelenthetnek, alkalmazásuknak számos gátja van.

Második tézisemben a internetes e-mail szerverek szolgáltatás-megtagadásos támadásaival foglalkozom. A specifikus támadásoktól eltekintve itt is szűk a mozgástér: a vizsgálataimat a hálózati forgalom analízisén alapuló módszerekre koncentráltam. Ez az SMTP szerverek esetében jelenthet általánosabb megoldást a problémákra.

Bármilyen általános megoldással szemben sokkal hatásosabbak lehetnek (és néha csak ezek lehetnek hatásosak) azok a megoldások, ahol a konkrét támadást, támadó szándékot, célt és módszert is figyelembe vesszük a védekezés során. Harmadik tézisemben DoS támadások előkészítése használt Directory Harvest Attack (DHA), azaz SMTP szerverek címkigyűjtést célzó támadásaival foglalkozom.

### 1. Tézis – Szolgáltatás-megtagadásos támadások (DoS) elleni védekezés client-side puzzle technika és játékelméleti módszerek együttes alkalmazásával

Publikációk: [J1], [C2], [C3]

**Újszerű, játékelméleti módszereket felhasználó megközelítést javaslok a DoS védelmet szolgáló client-side puzzle megoldások vizsgálatára és javítására. Megállapítom a szerver és támadó között levő játék optimumát a költségparaméterek függvényében. Új client-side puzzle rejtvényt mutatok be, amely prímszámok szorzatán alapul. A javasolt algoritmus előnye, hogy számításigénye pontosabban meghatározható, mivel egyszerű műveletekre épül.**

A DoS támadások elleni védekezés céljára olyan megoldásra teszek javaslatot, amely ötvözi a client-side puzzle technikát a játékelméleti analízis lehetőségeivel. Ehhez részletesen analizáltam és bemutattam, hogyan lehet a meglévő internetes protokollokat módosítani és védettebbé tenni a DoS támadásokkal szemben a client-side puzzle technika felhasználásával. Részletesen leírtam, hogyan írható le a támadó és a védekező fél magatartása játékelméleti módon és megmutattam, hogy kevert stratégiák alkalmazásával hogyan javítható a védekezés DoS támadásokkal szemben.

Javasolt megoldásom előnye, hogy eredményeim felhasználása segíthet a DoS támadások analízisében és így elősegítheti jobb védelmi módszerek tervezését.

Olyan skálázható client-side puzzle megoldásra teszek továbbá javaslatot, amely prímszámok szorzatán alapul. Nem egyszerű feladat megfelelő client-side puzzle rejtvény kivitelezése, mert számos tulajdonságot kell ötvöznie egy ilyen rejtvénynek. A rejtvény generálásának gyorsnak kell lennie, míg az ellenőrzésnek egyszerűnek. A rejtvényvel kapcsolatos tárolási erőforrások korlátosak lehetnek, így a rejtvényhez kevés vagy semmilyen adatot nem szabad tárolni. A rejtvény megfejtéséhez szükséges számítási kapacitást megfelelő pontossággal be kell tudni állítani. A javasolt client-side puzzle megoldásom számos jó tulajdonsággal rendelkezik, könnyen érthető és jól implementálható. Javasolt megoldásom legnagyobb előnye a hash-alapú rejtvényekkel szemben, hogy olyan alapvető számítási műveletre épül, mint a szorzás, így a rejtvény megfejtéséhez szükséges erőforrásigény sokkal pontosabban meghatározható. Annak érdekében, hogy a javasolt rejtvény a valóságban is könnyen alkalmazható legyen, megadom a szükséges paraméterek kiszámításának módját a számításigényt bemutató analízis segítségével.

## **A DoS elleni védelem modellezése**

A kriptográfiai protokollok (melyek alatt most olyan internetes kommunikációs protokollokat értünk, amelyek kriptográfiai műveleteket alkalmaznak) többnyire igen sérülékenyek DoS támadásokra, mivel komplex számításokat hajtanak végre, mint pl. a nyilvános kulcsú rejtjelezés. A DoS támadást végrehajtó támadó lehetőségei és céljai lényegesen eltérnek egy megszokott támadástól, amikor a támadás kriptográfiai protokollt céloz meg. A DoS támadó arra próbálja rábírní az áldozatot, hogy az sok műveletet végezzen el feleslegesen egy viszonylag rövidnek tekinthető idő alatt, és ezen műveletek lehetőség szerint minél nagyobb számítási igényt jelentsenek a kiszolgáló számára. Természetesen a kiszolgáló célja ekkor az lehet, hogy minimalizálja az így felmerült számítási költségeket. Mind a kliens, mind a szerver optimálisan próbálja felhasználni erőforrásait, valamilyen szempontból.

A kiszolgáló erőforrás-vesztéseinek csökkentésére több megoldást javasoltak már (pl. [1], [2]), de ezek hatóköre korlátos. Általánosabb védekezési megoldásként javasolták a client-side puzzle technikát (pl. [3],[4]), mellyel több probléma is kezelhető, mint a DoS támadások vagy a spam küldés. Kevesen vizsgálták azonban, hogy miként használható a client-side puzzle technika több forrású (distributed) DoS támadások ellen, továbbá az irodalom igen hiányos a pontos alkalmazástechnikát illetően. Nehéz feladat a modellalkotás és analízis egy értelmes támadót feltételező esetben. Munkám során értelmes támadót feltételezve, játékelméleti módszereket alkalmazva kívántam megvizsgálni a client-side puzzle technika gyakorlati megvalósítási lehetőségét, védelmi módszert adva a DoS egyes eseteire.

A client-side puzzle technika alkalmazása esetén egy kihívás-válasz jellegű protokoll-előttel egészíthetjük ki a kommunikációs folyamatot, amely védelmet jelenthet a protokoll ezután következő részei számára.

A DoS elleni védekezésben a client-side puzzle használatát játékelméleti módszerek bevezetésével, alkalmazásával kívánom hatékonyabbá tenni. A játékelméleti megközelítést két fél között

definiáltam. A kihívás-válasz alapú kézfogás (handshake) a kiszolgáló (S) és a kliens (C) között zajlik. A szerver a szolgáltatáskérésre elsőként egy rejtvényt (challenge) készít, és elküldi ezt a kliensnek. A kliens megoldja a rejtvényt, elküldi az eredményt. A kiszolgáló ellenőrzi a választ és ha az helyes, végrehajtja a kiszolgálást.

A támadó 3 fő lehetséges stratégia közül választhat. Az első támadási stratégia esetén a támadó csak a protokoll nyitólépését, a szolgáltatás kérését hajtja végre, a további protokolladatokra választ nem küld. Ennek a stratégiának a lényege, hogy a támadó nem akarja elpazarolni erőforrásait a megfelelő válasz kiszámítására, de az is elképzelhető, hogy nem képes kétirányú csatorna nyitására a kiszolgáló irányában.

Ha a támadó a második stratégiát választja, úgy egy hamis, nem kiszámolt, véletlenszerű megfejtést küld a kiszolgálónak annak érdekében, hogy rávegye a kiszolgálót a válasz ellenőrzésére és így az előző stratégiához képest további számításokra.

A harmadik stratégia esetén a támadó megfelelően végrehajtja a kézfogás összes lépését. A harmadik stratégiát kell alkalmazni a támadónak akkor, ha el akarja érni, hogy a kézfogás utáni költséges részeket (például a digitális aláírás készítése) is végrehajtsa a szerver.

A kliens oldali rejtvényt, DoS védelmi részprotokoll célját úgy is megfogalmazhatjuk, hogy azt kívánjuk elérni, hogy a támadó ritkán válassza ezt a stratégiát a támadás alatt, azaz a költséges rész ritkán kerüljön végrehajtásra. A támadó fenti stratégiáit jelöljük rendre  $A_1, A_2$  és  $A_3$ -mal.

Tételezzük fel, hogy a szerver védekezési stratégiái a feladott rejtvényt nehézségével, komplexitásával kerülnek megadásra. A kiszolgáló a rejtvényt algoritmikus bonyolultságot különböző szintekre tudja beállítani. Az egyszerűség kedvéért tételezzünk fel két kiszolgálós stratégiát, jelölje  $S_1$  egy kisebb bonyolultságú rejtvényt, míg  $S_2$  egy nagyobb bonyolultságú rejtvényt feladását.

A fentiek következtében a  $G(A_j, S_k)$  egy olyan játékot jelöl, ahol a támadó az  $A_j$ , míg a kiszolgáló az  $S_k$  stratégiát választja.

Már ez az leegyszerűsített játékelméleti modell is segíthet olyan protokollok megvalósításában, amelyek védettebbek DoS támadásokkal szemben, mint a jelenleg javasolt egyéb megoldások.

Költségelemek (támadó szempontjából) definiálásával felírható a játék mátrixa (1. táblázat).

		$S_1$	$S_2$
		$x_1$	$x_2$
$A_1$	$y_1$	$M_{11} = c_c(1) \cdot \frac{R}{c_r}$	$M_{12} = c_c(2) \cdot \frac{R}{c_r}$
$A_2$	$y_2$	$M_{21} = (c_c(1) + c_v(1)) \cdot \frac{R}{c_r + c_g}$	$M_{22} = (c_c(2) + c_v(2)) \cdot \frac{R}{c_r + c_g}$
$A_3$	$y_3$	$M_{31} = (c_c(1) + c_v(1) + c_e) \cdot \frac{R}{c_r + c_a(1) + c_p}$	$M_{32} = (c_c(2) + c_v(2) + c_e) \cdot \frac{R}{c_r + c_a(2) + c_p}$

R: rendelkezése álló erőforrások; költségek:  $c_r$ : szolgáltatás kérés;  $c_c(k)$ :  $k$  erősségű rejtvényt készítése;  $c_g$ : hibás válasz;  $c_a(k)$ :  $k$  erősségű rejtvényt helyes válasz;  $c_v(k)$ : válasz ellenőrzése;  $c_p$ : szolgáltatás kérés;  $c_e$ : szolgáltatás kiszolgálás

1. táblázat. A játék mátrixa

A játék megoldásaként tiszta és kevert stratégiák használatát választhatják a szereplők. Kevert

stratégiák esetében a játékosok saját stratégiájukat egy valószínűség-eloszlással adják meg, melyet a kiszolgálónál  $X = \{x_1, x_2\}$ , míg a támadónál  $Y = \{y_1, y_2, y_3\}$  segítségével jelölünk.

Az általam javasolt játékelméleti és client-side puzzle módszereket kombináló módszer segítségével tökéletesíthető a DoS elleni védekezés. Javasolt kombinált módszerem azért jelentős, mert a két módszer ötvözésével sokkal hatékonyabban elemezhetővé vált a DoS elleni védekezés és jobb megoldások is nyújthatóak a védekezés céljából.

A disszertációban bemutatom egyes tipikus esetekben a játék megoldásának számítási lehetőségeit, illetve azt, hogy ha a kiszolgáló tiszta stratégiát folytat, úgy a támadó is tiszta stratégiát fog követni, a nyereségmátrix ilyen felállásai esetén a kevert stratégia elfajul és tiszta stratégiává válik. A DoS védelmi módszerek területén a modellalkotás és analízis különösen nehéz feladat, ezt jelentősen segítettem munkámmal, amely előremozdíthatja a hasonlóan komplex védelmi intézkedések bevezetését.

### **Client-side puzzle javasolt rejtvény DoS támadás ellen**

Legyen  $T = \{p_1, p_2, \dots, p_N\}$  egy minden fél által ismert halmaza  $N$  prímszámnak. A kiszolgáló ebből a halmazból kiválaszt egy  $S$  részhalmazt, mely  $k$  prímet tartalmaz. A kiválasztás visszatétel nélkül történik, de a megoldás kiterjeszhető visszatételes esetre is. A kiszolgáló kiszámolja a kiválasztott elemek szorzatát, jelöljük ezt  $m$ -mel.

$$m = p_{i_1} \cdot p_{i_2} \cdot \dots \cdot p_{i_k}$$

A DoS ellen alkalmazható client-side puzzle rejtvénynek a következő új feladatokat javaslom:

- *1. rejtvény:* A rejtvény tartalma  $m$  értéke.
- *2. rejtvény:* Legyen  $m'$  az  $m$  szorzat módosítása oly módon, hogy  $\ell$  egymás után következő bitet, azaz  $m_r, m_{r+1}, \dots, m_{r+\ell-1}$  biteket 0-ra cseréljük  $m$  bináris reprezentációjában. A rejtvényt ez esetben a kapott  $m'$  és a hozzá tartozó  $r$  pozícióval adjuk meg.

Minkét rejtvény esetén a feladat az  $m$  (vagy  $m'$ ) prímfaktorainak meghatározása, és a faktorok megfelelő indexeinek elküldése a szerver részére. (Feltételezzük, hogy  $T$  rendezett és eszerint indexelünk.)

Nagyon nehéz feladat olyan client-side puzzle rejtvény definiálása, amely teljesíti mindazon követelményeket, ami a DoS elleni védekezés során szükséges. A rejtvény generálásának gyorsnak kell lennie, az ellenőrzésnek hatékonynak. A rejtvény kliens oldali megfejtése lehetőleg ne legyen párhuzamosítható, mindeközben lehetőleg legyen bizonyított, hogy a kliensnek nincs módja hatékonyabb megfejtő algoritmust létrehozni a tervezett feladatnál. Az irodalomban igen kevés konkrét rejtvény van, így az általam javasolt megoldás (mely igen jól kezelhető és a kritériumok szempontjából is fejlett) előrelépést jelenthet a client-side puzzle technika gyakorlati alkalmazása szempontjából is.

## A Javasolt rejtvény analízise

A javasolt rejtvény alkalmazhatóságának egyik fontos feltétele, hogy elemezve legyen teljesítményigénye, be legyen mutatva, hogy paramétereit hogyan kell beállítani. Elemzésemmel megmutatom, hogy miként lehet kiszámolni a kliens oldalon szükséges erőforrásokat, ezáltal lehetőséget nyújtok arra, hogy a gyakorlatban is implementálható legyen a javasolt client-side puzzle rejtvényem.

Az 1. rejtvény esetén a kliens a következő módon tudja kiszámítani a választ: legyen  $\mu$  egy változó, melyet minden számítási lépésben frissítünk. Legyen  $\mu = m$  a kezdeti érték. Az  $i$ . lépésben a kliens ellenőrzi, hogy  $p_i$  faktora-e  $\mu$ -nek (és így  $m$ -nek is.). Ha igen, úgy  $\mu$  módosításra kerül, értéke  $\frac{\mu}{p_i}$  lesz, a megtalált indexet eltároljuk; máskülönben  $\mu$  értéke nem változik. Ez az eljárás ismétlődő, míg  $m$  minden faktorát meg nem találjuk (azaz  $\mu$  eléri az 1 értéket).

A fenti számítás esetén  $k - 1$  darab osztás kerül elvégzésre<sup>1</sup>, az osztók  $n$  bit méretű prímszámok, az osztandó pedig fokozatosan csökken  $k n$  méretről  $2n$  méretig. Az osztások átlagos számára a következő képlet írható fel

$$D(N, k) = \sum_{i=k}^N q_i \cdot (i - 1) \quad (1)$$

ahol

$$q_i = \frac{\binom{i-1}{k-1}}{\binom{N}{k}} \quad (2)$$

A 2. rejtvény esetén a klienst arra kényszerítjük, hogy több számítást végezzen el. A kliens a következő két számítási eljárás közül tud választani:

- A kliens kipróbálja a kitörölt bitek összes helyettesítését. Ha egy helyettesítés nem megfelelő, úgy valószínűleg olyan prímfaktorokat fog kapni, melyek nem  $T$  részei. Ilyen esetben a kliens egy újabb helyettesítést tud elvégezni. Az osztások átlagos száma kb.  $N2^{\ell-1}$ , tehát a rejtvény megfejtésének nehézsége átlagosan  $2^{\ell-1}$  faktorialis növekedett.
- A kliens úgy is eljárhat, hogy kiszámolja  $T$  halmaznak  $k$  véletlenszerűen kiválasztott elemét és ellenőrzi szorzatukat, hogy az  $m'$ -et adja-e. Ezt  $m'$  megkapásáig folytatja. A szükséges szorzások átlagos száma

$$\frac{1}{2} \binom{N}{k} (k - 1)$$

---

<sup>1</sup>Az utolsó prímfaktort már nem kell ellenőrizni osztással.

A javasolt client-side rejtvényem könnyen implementálható, átlátható, emellett a bevezetéséhez szükséges paraméterszámításokat is megadom az erőforrásigények elemzésén keresztül. A disszertációban analizálom a javasolt client-side puzzle megoldást a mások által javasolt hashfüggvény alapú puzzle megoldásokkal. Az eredeti javaslatom publikálása után egy lehetséges támadást is felfedeztem az általam javasolt rejtvény ellen. A disszertációban ismertetem a támadást és a lehetséges védekezést is.

## **2. Tézis – Szolgáltatás-megtagadásos támadások elleni védekezés forgalom analízis segítségével internetes levelezési környezetben**

**Publikációk:** [C1], [C4], [C6]

**Új, a hálózat forgalomanalízisen alapuló védelmi módszert javaslok a DoS támadások kezelésére. A javasolt módszer nem teszi szükségessé a megtámadott szervereken kívüli hálózati elemek módosítását és minimalizálja a hibásan kiszűrt legális felhasználók számát.**

Mérések segítségével megmutattam, hogy sikeres Denial-of-Service támadás hajtható végre SMTP szerverek ellen alacsony sávszélesség felhasználása mellett, és a támadási lehetőség még valószínűbb, ha a szerveren tartalomszűrés is aktív. Méréseim a támadhatóság mértékét konkrét számokkal is alátámasztják.

Új védelmi módszert terveztem a DoS elleni védekezésre, amely hálózati forgalmi analízisen alapul. Egy ún. front-end bemeneti modul érzékeli a támadásokat és végzi el a támadók kiszűrését. Bemutatom a megoldásomban várható hamis pozitív és hamis negatív hibaválósínűségek kiszámításának módját is.

Felső becslést adtam a detekciós algoritmus hibaválósínűségére és a hibás azonosítás valószínűségére. Ezek a számítások lehetővé teszik az algoritmus valós körülmények közötti alkalmazásának paraméterezését. Szimulációk is megerősítették az analitikus eredményeket és információval szolgálnak a paraméterekre való érzékenység tekintetében.

Architektúrát terveztem, hogy rendszerem valós körülmények között is beágyazható legyen egy SMTP környezetbe. A tervezett architektúrára alapozva prototípus formában is megvalósítottam a rendszert.

**Motiváció: SMTP tartalomszűrők teljesítménytesztelése a DoS támadás lehetőségének felmérésére**

A szolgáltatás-megtagadásos támadások (DoS) ma legismertebb esetei a következők:

- A támadó egy speciális hibát kihasználva működésképtelenné teszi a kiszolgáló szolgáltatást (pl. ping of death, stb.).
- A támadó adatcsomagokkal árasztja el a szerveret, amely nem képes azt feldolgozni, sőt, gyakran a felhasználható sávszélesség is elfogy

Ez a két támadási forma ma tipikus, azonban a jövőben, kifinomultabb támadási formák esetén jelentősen változhat a támadási magatartás. A DoS támadás ugyanis könnyen, de legalábbis elérhető távolságban kivédhetővé válik abban az esetben, ha a támadó magatartása egyértelműen megkülönböztethető a legitim felhasználók tevékenységétől. A ma divatos SYN elárasztás (IP hamisítás nélkül) például tipikusan olyan tevékenység, amelyik statisztikai módon könnyen felderíthető, és a forrásgépek kiszűrésével ellene védekezési módszer hozható létre.

Noha az ilyen védekezési módok létrehozása is jelentős probléma, főként szervezési, koordinációs szempontok miatt, sokkal bonyolultabb kérdés az olyan támadások elleni védekezés, ahol a támadó nehezen, vagy szinte egyáltalán nem különböztethető meg a legitim felhasználóktól. Az ilyen támadók azt használják ki, hogy egy kérés sokkal jelentősebb erőforrásfelhasználást eredményezhet, mint a kérés erőforrásigénye annak létrehozójánál, kiváltképp az alkalmazási protokoll rétegben. Ezzel a módszerrel a támadó akár néhány adatcsomaggal olyan kéréseket intézhet a szerverhez, amely hosszú időre leköti annak teljesítményét.

Egy tipikus támadási lehetőség, ha egy e-mail kiszolgálónak (SMTP szervernek) leveleket küld a támadó annak reményében, hogy a levél feldolgozása jelentős erőforrásokat vesz igénybe, miközben alacsony sávszélesség-felhasználás történik, és így a támadó statisztikailag a hálózati rétegben nehezen megkülönböztethető a legitim felhasználóktól.

Vizsgálatokat végeztem annak érdekében, hogy megállapítsam, az SMTP szerverek - különös tekintettel a spam- és víruszűrésre mennyire védettek a DoS támadásokkal szemben, tehát mekkora erőforrást emészt fel egy-egy e-mail kézbesítése. Nem volt ugyanis ismert az, hogy mennyire ellenállóak az SMTP szerverek DoS támadási kísérleteknek, vagy más forrásból (pl. spam) eredő DoS állapotoknak. Különösen az a kérdés érdekelt, hogyan befolyásolja a kiszolgálási kapacitásokat a tartalomszűrés, azaz a spam- és víruszűrés. A tapasztalatok szerint sűrűn előfordul, hogy a levelező kiszolgáló nagy mennyiségű levél kézbesítése során órákra lelassul, de az irodalom igen szűkszavú azzal kapcsolatban, hogy mi várható el egy átlagos szervertől.

A mérések elvégzése nem egyszerű feladat, nem véletlen, hogy igen kevés megalapozott mérés áll rendelkezésre a szakirodalomban. A mérések megtervezését tovább nehezítette, hogy követniük kellett a kiszolgálók speciális működését, ahol a kiszolgálók a kézbesítést több jól elkülöníthető szakaszra bontják. A különböző leveleken végzett műveletek átlapolódnak, egyes szakaszok későbbi időpontra halaszthatóak, ráadásul magától a levélről is igen eltérő viselkedést tapasztalhatunk a kiszolgálónál. Mindemellett a levelező kiszolgáló óriási mennyiségű paraméterrel rendelkezik, sőt, egyes esetekben még további kommunikációt is végeznek harmadik felekkel, amely hálózati erőforrásokat is igényel. A levelezési kiszolgálás tehát igen komplex, különböző és nehezen vizsgálható erőforrást felhasználó feladat. Külön probléma a teljesítménytesztelés során, hogy a tesztként beküldött levelek előállítására is relatíve erőforrásigényes, ami befolyásolhatja a mérés pontosságát. Nehéz és körültekintő munkát igénylő feladat volt tehát olyan standard környezet kialakítása, ahol legalább részben összehasonlítható és jellemző adatot kapunk.

Méréseim során 4kb-os mintalevelek kerültek kiküldésre speciális környezetben több forrásból. Ehhez a tesztelt szervereket és tartalomszűrő megoldásokat a legismertebb és leggyakrabban használt megoldásokból választottam ki. Összegeztem az egyes levelezési kiszolgálók

vonatkozásában mért kézbesítési teljesítményt, miközben a kiszolgálón nem alkalmaztam tartalomszűrést (spam- vagy vírusszűrést). Az eredmények szerint a kézbesítési sebesség ilyen esetben 17-64 e-mail/sec között mozgott.

A mérés második fázisában megmutattam, hogy egy egyszerű vírusszűrés (jelen esetben Exim MTA, amavisd és ClamAV mellett) milyen jelentős mértékben csökkenti az átbocsátást. Az eredeti kb. 30 levél/s kézbesítési sebesség, 6,81 levél/s kiszolgálási sebességre csökkent, amely igen jelentős változás, és ez a változás igen komolyan befolyásolja a szolgáltatás DoS ellenállóságát is.

A 2. táblázat végüli bemutatja, hogy amennyiben a vírusvédelem mellett spam szűrést is alkalmazunk, az a spam szűréshez alkalmazott speciális megoldások, modulok függvényében még jelentősebben csökkenti az átvitelt.

2. táblázat. Csomagszűrés e-mail átviteli teljesítménye

haszált SpamAssassin modulok és üzenet	Átvitelhez szükséges átlagos idő	szórás	Átlagos kézbesítési sebesség (e-mail/sec)	szórás
Razor, bayes, fix üzenet	294.0	14.1	5.11	0.2
Razor, bayes véletlen üzenet	945.0	7.0	1.59	0.1
Local_tests_only, véletlen üzenet	448.0	15.5	3.38	0.2
Razor nélkül, bayes, véletlen üzenet	458.0	4.2	3.27	0.1
Razor, bayes nélkül, véletlen üzenet	975.1	7.4	1.54	0.1
Razor, mysql-bayes, véletlen üzenet	789.5	9.6	1.90	0.1

Összességében megfigyelhető, hogy egy átlagos szerver adatátviteli hálózati sebességhez képest meglepően kicsi kb. 30 levél/sec kiszolgálási sebessége a spam- és tartalomszűrés által akár 2 levél/sec alá is csökkenhet. Megmutattam, hogy egy átlagos levelezési kiszolgáló ellen van esélye olyan DoS támadás indításának, amely nem emészti fel a rendelkezésre álló kommunikációs vonal teljes kapacitását, a kiszolgálót mégis olyan mértékben terheli le, hogy az nem képes megfelelő működésre. A szükséges sávszélesség természetesen a konkrét paraméterektől függ, a fenti esetben kb. 1 Mbit/sec. A mérés megmutatja továbbá, hogy a tartalomszűrés alkalmazása igen nagy mértékben csökkenti a teljesítményt, és így a kiszolgálót jelentősen sebezhetőbbé teszi a DoS támadások ellen. A fenti példa esetében a támadáshoz szükséges sávszélesség kb. 64 kbit/sec. Ezek az eredményeim mutatják be annak fontosságát, hogy az SMTP szerverek DoS támadhatóságának kérdése igen fontos kutatási terület, noha kevesen foglalkoztak vele.

Az eredményeim referenciaként szolgálhatnak az SMTP szerverek támadhatóságának vizsgálata szempontjából, de segíthetik a levelező szerverek teljesítménymérésének jobb kidolgozását is, amelyre jelenleg nincsenek bevált, elterjedt standard módszerek. Méréseim természetesen csak megoldások szűk halmazát vizsgálták, természetesen a kidolgozott mérési elvek lehetőségét

nyitnak ennek bővítésére és nagyobb méretű adatgyűjtés után a teljesítmény-összetevők mélyebb elemzésére is.

### **Forgalmi méréseken alapuló védelmi modul**

Az SMTP protokollt alapul véve olyan megoldást javaslok, ahol a támadót forgalmi analízist használó módszer segítségével különböztetem meg a legitim felhasználóktól. A DoS probléma azon esetét próbáltam kezelni, ahol a protokoll átalakítása nem lehetséges, így nem vezethető be pl. client-side puzzle alkalmazása. A támadás során a támadó egyszerű szolgáltatáskérést hajt végre, amire a szerver nagyobb erőforrásokat köt le, azaz a támadó és a kiszolgáló erőforrásigénye jelentősen aránytalan. Másik lehetőség az elosztott (distributed) DoS támadás [5] esete, amikor a szükséges erőforrások a két oldalon nagyjából hasonlóak, de a támadó több erőforrással rendelkezik (pl. ún. zombi gépek révén). Ilyen esetben a védekezés lehetősége azon dőlhet el, hogy a támadó statisztikailag megkülönböztethető-e a legitim kliensektől. Amennyiben nem megkülönböztethetőek a támadók, jól látható, hogy nincs esélye hatékony védekezésnek. Ellenben, ha a támadó megkülönböztethető, márpedig a tapasztalatok szerint általában könnyen megkülönböztethetőek, akkor mód van védekezési lehetőségre forgalmi mérések felhasználásával és a támadók azonosításával. Ezt a lehetőséget megfelelő részletességgel az irodalomban nem vizsgálták, ezért javaslom a módszert a DoS elleni védekezésre és elemzem a matematikai hátterét is.

A legfontosabb megállapítás a támadó modellezése kapcsán az, hogy a támadó szempontjából kiemelkedő szerepe van annak, hogy minél több gépről indítsa a támadást, és így statisztikailag minél jobban megkülönböztethetetlen legyen a legitim felhasználóktól. Ha viszont minél több gépet von hatáskörébe és használja azokat támadásra, annál nagyobb a valódi támadó felderítésének és jogi felelősségrevonásának esélye. A két paraméter közötti trade-off az, ami meghatározza a támadó által használt optimális gépállományt és ami lehetőséget ad arra, hogy a védelem működőképes maradjon.

A védelmet képviselő algoritmuscsoportot négy részre bontottam és definiáltam:

- Támadást felismerő algoritmus, amely csúszóablakos statisztika segítségével forgalmi ugrások alapján (A1 algoritmus), illetve a sorhossz túlzott megnövekedésének érzékelésével (A2 algoritmus) ismeri fel a támadásokat.
- Támadókat azonosító algoritmus, amely a támadás mértékének becslésére és a magas forgalmat generáló kliensek azonosítására épül
- Támadás elnyomó algoritmus, amely a vélelmezett támadók kiszűrésén alapul
- A támadás-elnyomás sikerét vizsgáló algoritmus, mely a hálózati forgalom nyugalmi állapotba való visszatérését ellenőrzi

#### *A.) Támadás felismerése*

##### *A1. Algoritmus*

A támadás időpontját  $\hat{t}$  időpontban határozzuk meg, ha a következő esemény következik be:

1. Esemény: Ha a puffer mérete túllépi az előzőleg beállított  $L_1$  paraméter értékét

*A2. Algoritmus*

A  $\hat{t}$  időpontot tekintjük a támadás kezdetének, ha a következő esemény következik be:

2. Esemény:

$$\hat{\lambda}(\hat{t}) > (1 + r)\bar{\lambda}(\hat{t}) \quad (3)$$

ahol  $r$  ( $r > 0$ ) egy előre beállított paraméter,  $\bar{\lambda}(t)$  a hosszú távon mért forgalomszint, míg  $\hat{\lambda}(t)$  a rövid távon mért forgalmi szint.

*A3. Algoritmus*

A támadás időpontját abban a  $\hat{t}$  időpontban határozzuk meg, amikor az 1. Esemény illetve 2. Esemény közül valamelyik leghamarabb bekövetkezik.

*B.) Támadókat azonosító algoritmus*

Az előzőekben azonosított  $\hat{t}$  időponttól kezdődően megfigyeljük a kliensektől jövő összes és egyedi forgalmi mennyiségeket. Ha a támadást sikeresen azonosítottuk, azaz  $t^* < \hat{t} < t^* + \delta$ , úgy a méréseket a  $(t^* + \delta - \hat{t})$  időtartamban tudjuk elvégezni.

A mért eredményeket az összes forgalom tekintetében  $\hat{\lambda}_r(t^* + \delta)$  jelöléssel, míg az egyedi forgalmi szintekre  $\hat{\lambda}^{(i)}(t^* + \delta)$  jelöléssel kezeljük, ahol  $i$  a forrás számát jelöli.

Mivel nem tudjuk megállapítani a pontos forgalmi szintet, amelyet a legális források okoznak, a  $\bar{\lambda}(\hat{t} - c)$  (ahol  $c > 0$ ) mennyiséggel becsüljük az átlagos legális forgalmi szintet a  $[t^*, t^* + \delta]$  időszakban. Így a következő becsléshez jutunk a támadó források átlagos összeforgalmára vonatkoztatva:

$$\hat{\lambda}_a = \hat{\lambda}_r(t^* + \delta) - \bar{\lambda}(\hat{t} - c) \quad (4)$$

Az aktív forgalmi források  $Z$  halmazát a következő részekre bonthatjuk:

$$Z = Z_n \cup Z_a \quad (Z_n \cap Z_a = \emptyset) \quad (5)$$

ahol a  $Z_n$  és a  $Z_a$  a legális és támadó források halmazát jelöli. Az azonosító algoritmus  $Z$  halmaz egy  $Z_a^*$  részhalmazát válogatja ki. Ennek a halmaznak kell minél jobban  $Z_a$  közelében lenni.

A támadó forrásokat a következő algoritmus azonosítja:

*B1. Algoritmus*

Keressük meg azt a  $Z_a^* = \{i_1, i_2, \dots, i_v\}$  maximális méretű részhalmazát  $Z$ -nek, amely a legnagyobb mért forgalommal rendelkező forrásokat tartalmazza oly módon, hogy

$$\sum_{j=1}^v \hat{\lambda}^{(i_j)}(t^* + \delta) \leq \hat{\lambda}_a \quad (6)$$

*B2. Algoritmus*

Ne törődjük azokkal a forrásokkal, amelyek a  $Z$  halmazból már  $(\hat{t} - c)$ ,  $c > 0$  időszakban is aktívak voltak és használjuk a B1. Algoritmust.

C.) A támadó forgalmat elnyomó algoritmus

C. Algoritmus

Dobjunk el minden érkező adatot azoktól a forrásoktól, melyek a  $Z_a^*$  halmazban szerepelnek.

D.) Támadás elnyomás sikerességének ellenőrzése

D. Algoritmus

Amennyiben a C. Algoritmust is sikeresen hajtottuk végre, úgy a puffer felhasznált hosszúságának  $L_1$  alá kell kerülnie egy  $t_{out}$  timeout perióduson belül. Amennyiben ez nem történik meg, úgy további forrásokból érkező forgalmat kell eldobnunk, mégpedig oly módon, hogy azok a legnagyobb forgalmat kibocsájtó forrásokhoz tartozzanak. A szűrés sikerét ezután újraellenőrizzük és a két lépést addig folytatjuk, míg az elégséges felhasznált pufferhosszt el nem érjük.

### Hálózati forgalmi mérésen alapuló védelmi modul analízise

A disszertációban analizáltam a javasolt megoldás egyes tulajdonságait, mint a hibás detekció valószínűségét. Kétfajta hibás detekció történhet: I. típusú hiba az, amikor valójában támadás van, ám azt az algoritmusunk nem ismeri fel, míg II. típusú hiba, ha tévesen támadást detektál az algoritmusunk, miközben valójában nincs támadás. Két fajta hipotézis létezik a  $t$  időpontban:

$H_0$ : nincs támadás állapot

$H_1$ : támadás van állapot

A definiált hibák a következők:

$$P_I = P(\{H_0 \text{ hipotézisre döntés } t \text{ időpontban}\} | H_1) (= 0), \quad (7)$$

$$P_{II} = P(\{H_1 \text{ hipotézisre döntés } t \text{ időpontban}\} | H_0). \quad (8)$$

Feltételezzük, hogy a rendszer normál állapotban van hosszú ideig, amikor ez megváltozik. Ekkor  $P_{II}$  értékére a következő felső becslés adható:

$$P_{II} = P(\{\text{várakozási sor hossza} \geq L_1\} \cup \quad (9)$$

$$\{\hat{\lambda}(\hat{t}) > (1 + r) \cdot \bar{\lambda}(\hat{t})\} | H_0)$$

$$\leq 2 \cdot \max\{P_{II,A1}, P_{II,A2}\} \quad (10)$$

melyben az  $A1$  és  $A2$  a két támadást detektáló algoritmust jelzi. ( $L_1$ : maximális sorhossz,  $r$ : forgalom-ugrási érzékenységi paraméter,  $\hat{\lambda}$  rövid távon mért forgalomsebesség  $\bar{\lambda}$  hosszú távú forgalom,  $\hat{t}$ : támadás felismerésének ideje )

Melyben a két algoritmusra vonatkozó egyedi részek:

$$P_{II,A1} = P(\{\text{várakozási sor hossza} \geq L_1\} | H_0) \quad (11)$$

$$P_{II,A2} = P(\{\hat{\lambda}(\hat{t}) > (1+r) \cdot \bar{\lambda}(\hat{t})\} | H_0) \quad (12)$$

A  $P_{II,A1}$  standard buffertervezéssel kalkulálható, míg  $P_{II,A2}$  vonatkozásában felső becslést adok meg.

$$P\{\hat{\lambda}(\hat{t}) > (1+r) \cdot \bar{\lambda}(\hat{t})\} \leq \frac{1}{w_s r^2} \left( \frac{\sigma_n}{\lambda_n} \right)^2 \quad (13)$$

( $w_s$  a csúszóablak mérete,  $\lambda_n$ : támadás nélküli (normál) állapot forgalomsebessége,  $\sigma_n$ : támadás nélküli állapot sebességének szórása)

A támadás felismerés sikerességének formális vizsgálata mellett az egyedi támadók felismerésének hibalehetőségét is analízálom. Ennek eredményeképpen bemutatom, hogy a támadó szempontjából a felismerhetőség ellen célszerű, ha az egyes támadási forrásokból egyenletesen, azonos mennyiségben indít támadást a kiszolgálóval szemben. A disszertációban megvizsgálom továbbá az algoritmusaimban használt egyes paraméterek szerepét és fontosságát, melyek vizsgálatához szimulációs eredményeket is felhasználok. Javasolt védelmi módszerem alkalmazhatósága jelentősen függ a védett szolgáltatástól ill. protokolltól. Módszerem és elemzésem azonban segíti azt, hogy specializáltabb megoldások szülessenek.

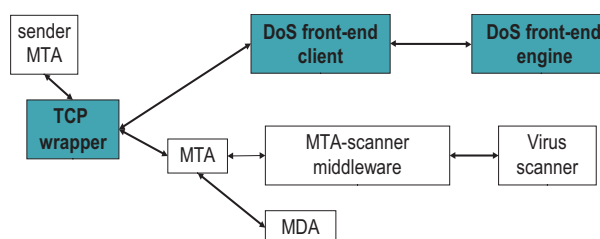
### **Prototípus SMTP DoS támadás elleni védelemre hálózati analízissel**

A javasolt eljárás működőképességének bemutatását prototípus létrehozásával végeztem el. A prototípus alapját egy SMTP kiszolgáló jelenti valós idejű vírusvédelemmel kiegészítve. A prototípus megvalósítása nehéz feladat, mert az SMTP kiszolgálók komplikált, bonyolult szoftvereszközök, módosításuk nem triviális. Módszerem implementálásához biztosítani kellett a mérés lehetőségét, létre kellett hozni és tesztekkel ellenőrizni a statisztikai módszereket és integrálni kellett a támadók kiszűrését is a meglévő infrastruktúrába. A létrehozott megoldás tesztelése is igen sok munkát igényelt, hiszen a statisztikai módszerek működőképességét is ellenőrizni kellett.

A kiinduló rendszer a következő modulokból áll: Exim MTA (SMTP szerver), Amavisd-new tartalomszűrő middleware, és vírusszűrő program, amelyet szükség esetén az Amavisd hív meg. Ez a rendszer ki van téve DoS támadásoknak, mivel a vírusok kiszűrése jelentős erőforrásokat emészt fel, és így egy támadó megfelelő, nem túl nagy sávszélesség mellett (a mai kommunikációs vonalak sebességéhez képest) képes a szervert teljesen leterhelni.

Az általam javasolt rendszer felépítését komponens szinten a 1. ábra mutatja. Az ábrán kiemeltem az általam megvalósított szoftverkomponenseket, melyeket egy nyílt forráskódú szoftverekből álló alaprendszerbe helyeztem bele. A kiinduló rendszert a védelem biztosítása céljából kiegészítettük egy ún. TCP wrapperrel, egy DoS front-end klienssel és egy DoS front-end engine központi elemmel. A TCP wrapper egy olyan egyszerű program, amely első

lépésben csatlakozik a DoS front-end klienshez, és lekérdezi, hogy a másik fél jogosult-e e-mailt küldeni. Megfelelő válasz esetén a wrapper a kapcsolat adatait továbbítja a levelező kiszolgáló (mail transport agent-MTA) felé. A DoS front-end kliens egy egyszerű alkalmazás, melynek célja, hogy kapcsolatba lépjen a front-end engine központi elemmel és a központi elemek válaszát megfelelő módon továbbítsa a wrapper számára. A front-end engine több részből áll. Egyrészt tartalmaz egy statisztikai magot, amely állapotváltozók karbantartásával méri és kíséri figyelemmel a hálózati forgalmat (a kliensek csatlakozásai alapján), továbbá tartalmaz egy döntőegységet, amely az előzőekben definiált algoritmusok alapján eldönti, hogy a másik féltől érkező levelet ki kell-e szűrni.



1. ábra. Prototípus felépítése

Prototípusom jól bemutatja módszerem működőképességét, de valójában olyan konkrét, termékszintű, működőképes rendszert építettem, amely tényleges védelmet tud biztosítani a DoS támadás ellen, legyen az akaratlagos vagy indirekt támadás (pl. vírusfertőzött gép elárasztja a szerveret). Egyes kutatók más, speciális hálózati megoldásokat javasoltak a DoS támadások elleni védekezésre, de ezek többnyire csak akkor működőképesek, ha az egész világon elterjesztik használatukat (pl. [6],[7]), míg az általam javasolt technikánál erre nincs szükség. A javasolt megoldás gyakorlati alkalmazása közben természetesen minden olyan módszert körültekintően lehet csak alkalmazni, amely szűréseket használ, így az általam készített prototípusban is felmerül a téves pozitív esélye, ezért használata során megfelelő felügyeletet igényel.

### 3. Tézis – DoS előkészítő támadások elleni védekezési módszerek

Publikációk: [J2], [J3], [C7], [C8], [C5], [O13], [C9]

**A 3. tézisben új megoldásokat javasolok a DoS előkészítő támadások elleni védekezésre. Az ún. Directory Harvest Attack (DHA, címkinyerést célzó támadás) elleni védekezésre új módszert és architektúrát javasolok és megadom az optimális DHA támadás algoritmusát is. A javasolt védelmi módszer prototípusban is megvalósítottam és szimulációkkal is elemeztem.**

A tézisben analizálom az ún. Directory Harvest Attack (DHA) e-mail címkigyűjtő támadásokat. A DHA tipikus célja e-mail címek összegyűjtése, de maga a DHA támadás akár DoS helyzet kialakulását is okozhatja. Megterveztem a DHA támadás egy optimális formáját, amely az e-mail címek felhasználóneveinek, az ún. local partok valószínűségi eloszlását és a célrendszerben feltételezett felhasználószámot használja ki az optimalizáláshoz.

Képletet adok meg a sikeresen detektálható címek számának várható értékének kiszámításához. Bizonyítom, hogy modellemben a támadás optimális, azaz a sikeresen összegyűjtött e-mail címek száma adott erőforrásfelhasználás mellett maximális. Szimulációval is bemutatom, hogy az optimális támadás hatékonyabb a megfigyelt támadási formáknál. A szimuláció valós adatokon alapul. Mindezek bemutatásához megterveztem egy központosított, RBL (real-time blacklist) alapú architektúrát a DHA támadások elleni védekezésre. A tervezett rendszert prototípus szinten is megvalósítottam, hogy alátámasszam állításaimat.

### **Optimalizált Directory Harvest Támadás (DHA)**

Az e-mail Directory Harvest Attack (DHA), azaz címkinyerést célzó támadás a brute-force, azaz nyers erőt használó támadások speciális típusa. A támadó célja, hogy a megcélzott domainben e-mail címek létezéséről szerezzen információt, még hozzá speciális módon. A támadó próbalevelet küld a célzott címekre, és amennyiben a kézbesítés közben hibüzenetet kap, úgy feltételezi, hogy a célcím nem létezik. Amennyiben azonban nem kap hibüzenetet, mint potenciálisan létező e-mail címet listára veszi azt. A DHA támadások tekintetében a szakirodalom hiányos, sem analízist sem védelmi megoldást nem javasolnak. Egyes kereskedelmi szoftverek védelmet ígérnek a problémára, de a pontos védelmi módszert sem jelölik. ([8],[9])

A DHA támadókat két fő kategóriába sorolom:

- A támadó kipróbál minden lehetséges karakterkombinációt 1..N karakter hosszúságig, így keresi meg a létező e-mail címeket. A véletlen karaktersorozatokról esetenként a támadó csak a szavaknak látszó karaktersorozatokat próbálja ki.
- A támadó egy szótárt használ a támadáshoz, amely szótár a leggyakrabban használt felhasználóneveket (ún. e-mail local part részeket) tartalmazza. A szótárt többnyire ismert e-mail címek alapján hozzák létre.

Tapasztalataim szerint a támadók többnyire egy nagyjából fix méretű szótárt próbálnak felhasználni a DHA során néhány kiválasztott domain ellen. A disszertációban bemutatom, hogy a támadás azon módszere, hogy a kiválasztott domainek egy részét azonos méretű szótár segítségével támadja a támadó nem optimális. Bemutatom, hogy bizonyos plusz információk birtokában (e-mail local-part részek régióban ismert eloszlása, e-mail címek becsült száma az adott domain alatt) a fix erőforrásokkal rendelkező támadó bizonyos szempontból optimális támadást tud tervezni. Az optimális DHA támadást úgy definiálom, hogy a támadás során a támadó fix erőforrásfelhasználás mellett a legtöbb e-mail címet gyűjti össze sikeresen, azaz maximális a várható értéke a begyűjtött címek számának az előzetes információk birtokában. Az optimális támadás feltárása azért szükséges és hasznos, mert várható, hogy a támadók hosszútávon optimalizálják támadásaikat, így a védelmi módszereknek az optimális támadásra kell felkészülniük, amit más még nem adott meg.

A támadó az optimális esetben kapacitását úgy osztja fel, hogy a nagyobb becsült e-mail címkészlettel rendelkező domainek esetében a szótár nagyobb részét, míg kisebb domainek esetében kisebb részét teszteli.

Az optimális viselkedés pszeudokódját a következőképpen adom meg:

```

for all  $d \leq N_D$  do
     $i[d] \leftarrow 1$ 
end for
 $trial \leftarrow 1$ 
while  $trial \leq t$  do
     $d \leftarrow 1, p_{max} \leftarrow 0, target \leftarrow 0$ 
    while  $d \leq N_D$  do
        if  $p(W_{i[d]} \in U^d) > p_{max}$  then
             $p_{max} \leftarrow p(W_{i[d]}), target \leftarrow d$ 
        end if
    end while
    Try(word  $W_{i[target]}$ , on domain  $target$ )
     $i[target] \leftarrow i[target] + 1$ 
     $trial \leftarrow trial + 1$ 
end while

```

$N_D$ : a célpont domainek száma;  $i[d]$ : eddig kipróbált szavak száma a  $d$  indexű domainben;  $t$ : támadó erőforrása kísérletek számában kifejezve;  $U^d$ : a  $d$ . domain felhasználóinak halmaza;  $W$ : e-mail cím local\_partok valószínűség szerint csökkenő sorrendbe rendezett listája;  $p$  a valószínűséget jelöli;  $p_{max}, target, d$ : segédváltozók

A támadót algoritmusá tehát az, hogy minden kísérlet előtt kiválasztja a következő célpontot és a szótár következő tesztelendő elemét, így azt mondhatjuk, hogy a támadó azt a célpontot és azt a szót fogja tesztelni, amely esetben a következő szó tesztelése során a legnagyobb eséllyel fordul elő az, hogy segítségével újabb létező címre bukkan.

## Optimális támadás analízise

A sikeresen feltárható címek várható értékére megadott formulám a hagyományos, nem optimális algoritmus esetén a következő:

$$E(S) = \sum_{i=1}^{N_D} N_U^i \sum_{j=1}^T P(W_j)$$

ahol  $S$  A sikeresen feltárt címek száma,  $T$  az elvégzett kísérletek száma az egyes domainek tekintetében. Az optimális esetben a feltárható címek várható értékére a következő formulát adom meg:

$$E(S) = \sum_{i=1}^{N_D} N_U^i \sum_{j=1}^{t_i} P(W_j)$$

A Lagrange multiplikátor technikát használva

$$H(t) = n_1 h(t_1) + \dots + n_k h(t_k)$$

maximumát keressük, ahol  $n_i = N_U^i$ . Amennyiben  $h(x) = a/x^b$ , alakú és  $h'(x) = c/x^d$   $d = b + 1$ , úgy a következő eredményre jutok:

$$t_i = t \cdot n_i^{1/d} / \sum_{i=1}^{N_D} n_i^{1/d}, \quad i = 1, \dots, N_D$$

Az optimális támadást analizálom és bizonyítom optimális voltát. Az eredményeket szimulációval is erősítem. Valós adatok felhasználása mellett a szimulációim átlagosan azt mutatták, hogy az optimális esetén jelentősen több cím kinyerésére volt képes a támadó azonos erőforrásbefektetés mellett.

A szimuláció két scenáriót dolgoz fel, az egyik esetben 11, a másik esetben 6 megcélzott domain volt definiálva. Ezekben 1730 illetve 23000 címet tekintettünk létezőnek. A 3. tban látható, hogy 2., optimális algoritmus esetén jelentősen több cím kinyerésére volt képes a támadó azonos erőforrásbefektetés mellett.

3. táblázat. DHA szimulációs eredmények

$N_D$	Össz. felhasználó	feltárt címek, 1. alg.	feltárt címek 2. (optimális) alg.
11	1730	87	180
6	23000	5395	6754

### Központosított védelem DHA támadásra

A DHA támadások elleni védelemre központosított, hálózati alapon működő védelmi rendszert javaslok. Az irodalomban más javasolt megoldást DHA támadások ellen nem ismerek. Az általam javasolt rendszer lényege, hogy az internetes levelező szervereken folyamatosan vizsgáljuk a téves címzéssel ellátott levelek beérkezését, és mindazon IP számokat, amelyről bizonyos mennyiség feletti mértékben érkezik téves levél, egy központi rendszer felé jelentjük. A központi rendszer egy megbízható harmadik fél, aki összegzi a végpontoktól érkezett adatokat, elemzi és aggregálja azokat, majd listát állít fel azokról az IP címekről, melyek támadóknak tekinthetők. A kliensek ezután minden levél érkezése előtt képesek lekérdezni, hogy az adott szerver támadónak tekinthető-e és amennyiben támadónak tekintett a levelet küldő számítógép, úgy még a komolyabb tartalomszűrés megkezdése előtt képesek megszakítani a kapcsolatot. A tervezett mechanizmus egyfajta RBL (Real-time Black List) mechanizmus, így javasolt megoldásom a közismert RBL eljárások körében elterjedt módszerekre és szoftvermegoldásokra épül.

A javasolt megoldást prototípus elkészítésével is alátámasztottam. A prototípus tartalmaz egy logelemző és bejelentő modult, illetve egy szerver oldali modult a bejelentések kezelésére és a támadók adatbázisának karbantartására. A megoldás továbbá tartalmaz egy szűrőmodult, amely képes a szervertől lekérdezni a kapcsolódó számítógépekkel kapcsolatos esetleges támadási információkat és ez alapján szűrést végezni. A szűrést végző funkciót a DoS védelmet forgalmi analízis segítségével végző front-end modul további bővítésével hoztam létre.

A javasolt védelmi módszereket, illetve a paramétereik hatását szimulációkkal is megvizsgáltam. A szimulációkon keresztül bemutatom, hogy ha növeljük a hálózatban a detektorok számát, az javítja a

támadók felismerésének képességét, ugyanakkor a téves pozitívok száma is nő. Hasonlóan megmutatom azt is, hogy a lokális és központi detekciós rendszer toleranciájának csökkentése (kevesebb detektált esemény alapján döntünk támadónak egy forrást) szintén növeli a felismerés sebességét, de a tévesen támadónak tartott normál felhasználók számát is. Az ún. aging algoritmusok viselkedését, és azok hatását főként a fent említett téves pozitív felismerésre is szimulációval vizsgálom. Végül, egy adaptív eljárást is javasolok, amellyel hatékonyan csökkenthető a téves pozitívok száma.

## 6. Az eredmények alkalmazása

Az általam elért kutatási eredményeket munkám során mindig megpróbáltam a gyakorlati életbe is átültetni. A következő prototípusokat hoztam létre:

- Forgalmi elemzésen alapuló DoS védelmi front-end
- DHA védelmet biztosító hálózati védelmi rendszer prototípus

Az SMTP DHA és DOS elleni védelmi modulomat ipari környezetben, több valós SMTP szerveren felhasználók százainak védelme érdekében éles körülmények között is felhasználtam, teszteltem. Természetesen az eredeti megoldáson finomítani kellett a sok zavaró téves pozitív miatt, erre főleg ún. white-list megoldást alkalmazok, azaz egy listát tartok fenn azon szerverekről, ahonnan gyakorta érzékelünk tévesen támadást és ezzel az eredményeket korrigálom.

Az SMTP-vel kapcsolatos védelmi módszereimet, illetve az azokra épülő továbbfejlesztéseket és újabb védelmi megoldásokat felhasználjuk az Eu. 6. keretprogramjának (FP6) 2006. januárjában indult DESEREC (Dependable security by enhanced reconfigurability - szerződés szám 026600) IP projektje során is.

## Hivatkozások

- [1] K. Matsuura and H. Imai "Protection of authenticated key-agreement protocol against a denial-of-service attack" In Proceedings of the International Symposium on Information Theory and Its Applications (ISITA'98), pp. 466–470, October 1998.
- [2] J. Leiwo, T. Aura, and P. Nikander "Towards network denial of service resistant protocols" In Proceedins of the IFIP SEC 2000 Conference, August 2000.
- [3] C. Dwork and M. Naor "Pricing via processing or combatting junk mail" In Advances in Cryptology – Crypto '92, Springer-Verlag, LNCS volume 740, pp. 139–147, August 1992.
- [4] A. Juels and J. Brainard "Client puzzles: A cryptographic countermeasure against connection depletion attacks" In Proceedings of the IEEE Network and Distributed System Security Symposium (NDSS '99), pp. 151–165, February 1999.
- [5] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajovic. "Distributed denial of service attacks" In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 2275–2280, October 2000.
- [6] Ferguson, P. and D. Senie. "Network Ingress Filtering: Defeating Denial Of Service Attacks Which Employ IP Source Address Spoofing" RFC 2827, May 2000.
- [7] 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.
- [8] Kerio MailServer - state-of-the-art secure email server, [http://www.kerio.com/kms\\_home.html](http://www.kerio.com/kms_home.html). 2004.
- [9] Postini Enterprise Spam Filtering. "The Silent Killer: How Spammers are Stealing Your Email Directory" <http://www.postini.com/whitepapers/>, June 2004.
- [10] S.Hird "Technical Solutions for Controlling Spam" In the proceedings of AUUG2002, Melbourne, 4-6 September, 2002.
- [11] Ronald F. Guilmette "wpoison – small CGI script to combat junk email" <http://www.monkeys.com/wpoison/>.
- [12] Devin Carraway - Sugarplum automated spam-poisoner, <http://www.devin.com/sugarplum/>.
- [13] The Honeynet project "Know Your Enemy: Learning About Security Threats" Addison-Wesley. 2002.
- [14] L. Spitzner "Honeypots: Tracking Hackers" Addison-Wesley, 2002.

## A téziseket alátámasztó publikációim

### Folyóiratban megjelent cikkek

[J1] B. Bencsáth, I. Vajda, Internet Denial of Service attacks in game theoretical model (in hungarian), *Alkalmazott Matematikai Lapok* 23, 2006, pp. 335-348.

[J2] B. Bencsáth, I. Vajda, Efficient Directory Harvest Attacks and Countermeasures, *International Journal of Network Security*, vol 5. no 3. pp. 264-273. 2007.

Idézők (1):

C. Chris Erway. MicroID considered harmful (to privacy). Technical Report, Brown University, 2008.

[J3] Géza Szabó, B. Bencsáth, Protection against DHA attack with central filtering (in hungarian), *Híradástechnika*, 2006, vol. LXI, pp. pp. 2-9, 05.

[J4] I. Askoxylakis, B. Bencsáth, L. Buttyán, L. Dóra, V. Siris, D. Szili, and I. Vajda Securing Multi-operator Based QoS-aware Mesh Networks: Requirements and Design Options, *Wireless Communications and Mobile Computing (Special Issue on QoS and Security in Wireless Networks)*, accepted for publication in 2009.

### Konferencián megjelent cikkek

[C1] B. Bencsáth, M. A. Rónai Empirical Analysis of Denial of Service Attack Against SMTP Servers, *Proceedings of The 2007 International Symposium on Collaborative Technologies and Systems*, IEEE, 2007, pp. 72-79.

[C2] B. Bencsáth, L. Buttyán, I. Vajda, A game based analysis of the client puzzle approach to defend against DoS attacks, *Proceedings of IEEE SoftCOM 2003* 11., Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 2003, pp. 763-767.

Idézők (13):

E. Altman and K. Avrachenkov and G. Miller and B. Prabhu. Discrete Power Control: Cooperative and Non-Cooperative Optimization, *Proc. of IEEE Infocom 2007*, Anchorage, Alaska, USA, May 6-12, 2007.

Mehran Fallah, A Puzzle-Based Defense Strategy Against Flooding attacks Using Game Theory, *IEEE Transactions on Dependable and Secure Computing*, 12 Feb. 2008.

A. Patcha and J-M. Park. A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks, *International Journal of Network Security*, Vol. 2, No. 2, 2006, pp. 131-137.

A. Patcha. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks, *IEEE Workshop on Information Assurance and Security*, June 2004.

T.J. McNevin and J.M. Park, and R. Marchany Chained puzzles: a novel framework for IP-layer client puzzles *Wireless Networks*, 2005 International Conference on Communications and Mobile Computing, pp. 298-303.

Yi Gao, Willy Susilo, Yi Mu, and Jennifer Seberry, Efficient Trapdoor Based Client Puzzle Against DoS Attacks, *Book Chapter in Network Security*, 2006

T. J. McNevin, J.M. Park, and R. Marchany. pTCP: A Client Puzzle Protocol For Defending Against Resource Exhaustion Denial of Service Attacks, Technical Report TR-ECE-04-10, Dept. of Electrical and Computer Engineering, Virginia Tech, Oct. 2004.

V.Laurens and A. El Saddik and A. Nayak. Requirements for Client Puzzles to Defeat the Denial of Service and the Distributed Denial of Service Attacks The International Arab Journal of Information Technology. Vol. 3. No. 4., pp. 326-333, 2006.

Lin, C. and Wang, Y. and Wang, Y. and Beijing, PR. A Stochastic Game Nets Based Approach for Network Security Analysis CHINA 2008 Workshop (Concurrency methOds: Issues aNd Applications), pp. 24-35.

Yuanzhuo Wang, Chuang Lin, Yang Yang, Junjie Lv, Yang Qu, A Game-Based Intrusion Tolerant Mechanism for Grid Service,pp.380-386, Fifth International Conference on Grid and Cooperative Computing (GCC'06), 2006

Network Research Foundations and Trends EU FP6 project, Deliverable D4.1 State-of-the-art report on tools and techniques for achieving Autonomous Network Operation, 2006.

Sen, J. and Chowdhury, P.R. and Sengupta, I. A Mechanism for Detection and Prevention of Distributed Denial of Service Attacks, Lecture Notes in Computer Science, Vol. 4308, pp. 139-144, Springer, 2006.

A. E. Goodloe. A foundation for tunnel-complex protocols. PhD Thesis, University of Pennsylvania, 2008.

[C3] B. Bencsáth, I. Vajda, A game theoretical approach to optimizing of protection against DoS attacks, presented on the Second Central European Conference on Cryptography (Hajducrypt), Július, 2002

[C4] B. Bencsáth, I. Vajda, Protection Against DDoS Attacks Based On Traffic Level Measurements, 2004 International Symposium on Collaborative Technologies and Systems, The Society for Modeling and Simulation International, 2004, 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.

Idézők (6):

Tan, H.X. and Seah, WKG, Framework for statistical filtering against DDoS attacks in MANETs,2005. Second International Conference on Embedded Software and Systems, pp. 456-465, 2005.

Kumar, K. and Joshi, RC and Singh, K. An Integrated Approach for Defending Against Distributed Denial-of-Service (DDoS) Attacks, IRISS 2006, Madras, 2006.

Sardana, A. and Joshi, R. and Kim, T. Deciding Optimal Entropic Thresholds to Calibrate the Detection Mechanism for Variable Rate DDoS Attacks in ISP Domain, International Conference on Information Security and Assurance, pp. 270-275, ISA 2008, 2008.

Kumar, K. and Joshi, RC and Singh, K. An ISP level Distributed Approach to Detect DDoS Attacks, Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, pp. 235, ISBN: 978-1-4020-6265-0, Springer, 2007.

El Hassan, A.C.M. and Maalouf, S. and Zouheiry, A. A Survey of DDoS Defense Mechanisms, American University of Beirut.

Yuan-Shun Dai, Xukai Zou, Yi Pan. Trust and Security in Collaborative Computing (book), ISBN 9812703683, World Scientific, 2007.

- [C5] B. Bencsáth, The problems and connections of network virus protection and the protection against denial of service attacks, Proceedings of the Networkshop 2004 Conference, NIIF, Hungary, 2004, NIIF, Hungary.
- [C6] Géza Szabó, B. Bencsáth, Statistical analysis of the results of the DHA protection system (in hungarian), Proceedings of Networkshop 2006 conference, NIIF, 2006, NIIF.
- [C7] B. Bencsáth, I. Vajda, Efficient Directory Harvest Attacks, Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, pp. 62-68., IEEE Computer Society, July 2005.
- [C8] B. Bencsáth, Az internetes vírus- és spamvédelem rendszerszemléletben, HISEC 2004 konferencia, 2004, 10., Budapest, in Hungarian.
- [C9] B. Bencsáth, Géza Szabó, Components to improve the protection against spam and viruses, HSN LAB Workshop, 2005, Jun.

#### **Egyéb publikációk**

- [O1] B. Bencsáth, I. Zs. Berta, Empiric examination of random number generators of smart cards, HTE-BME 2002 Korszerű távközlő és informatikai rendszerek és hálózatok konferencia, BME, 2002, BME.
- [O2] B. Bencsáth, I. Vajda, Collecting randomness from the net, Proceedings of the IFIP TC6 and TC11 Joint Working Conference on Communications and Multimedia Security 2001, Kluwer, 2001, pp. 105-111, Kluwer, May.
- [O3] I. Zs. Berta, B. Bencsáth, Hiteles üzenet küldése rosszindulatú terminálról, NetWorkShop2004, NIIF, CD Proceedings, Győr, 2004.
- [O4] B. Bencsáth, S. Tihanyi, Home-made methods for enhancing network security (in Hungarian), Magyar Távközlés, 2000, vol. X, no. 4, pp. 22-27..
- [O5] B. Bencsáth, T. Tuzson, B. Tóth, T. Tiszai, G. Szappanos, E. Rigó, Sz. Pásztor, M. Pásztor, P. Papp, P. Orvos, P. Mátó, B. Martos, L. Kún, Z. Kincses, T. Horváth, M. Juhász, B. K. Erdélyi, A. Bogár, G. Vid, Az informatikai hálózati infrastruktúra biztonsági kockázatai és kontrolljai, IHM - MTA-SZTAKI, 2004.
- [O6] I. Zs. Berta, I. Vajda, L. Buttyán, B. Bencsáth, T. Veiland, E-Group Magyarország Specification of the Hungarian electronic ID card (HUNEID) Információs Társadalom Koordinációs Tárcaközi Bizottság, Intelligens Kártya Munkacsoport, <http://www.itktb.hu>, 2004
- [O7] B. Bencsáth, Simple, free encrypted tunnels using linux, Presented on Networkshop 2000, Gödöllő, Hungary, 2000
- [O8] I. Vajda, B. Bencsáth, A. Bognár, Tanulmány a napvilágra került Elender jelszavakról, 2000, Apr. (átvéve: Chip, Alaplap, On-line oldalak)

- [O9] B. Bencsáth, S. Tihanyi, Problem areas of the security aspects of network operating systems, Scientific student groups (TDK) 1999
- [O10] B. Bencsáth, Multiple Security Flaws Lead to Netenforcer Privilege Escalation (TAR Issue Details), report, BugTraq, <http://www.securiteam.com/securitynews/6V00R0K5QY.html>, CVE-2002-0399, 2002.
- [O11] B. Bencsáth, Sudo Private File Existence Information Leakage Vulnerability, report, Bugtraq <http://www.securityfocus.com/bid/321>, CAN-1999-1496, 1999.
- [O12] A. Szentgyörgyi , G. Szabó , B. Bencsáth. Bevezetés a botnetek világába (in Hungarian), Híradástechnika, 2008. vol. LXIII, pp. 10-15. Nov. 2008. (A cikk a HTE Pollák-Virág díját nyerte el)
- [O13] B. Bencsáth, I. Vajda, Trap E-mail Address for Combating E-mail Viruses, Proceedings of IEEE SoftCOM 2004 12. International conference on software, telecommunications and computer networks, University of Split, 2004, pp. 220-224, University of Split, October.