TEHNICI PENTRU SPORIREA REZISTENȚEI ÎN FAȚA ATACURILOR DOS ȘI DDOS PRIN PROTOCOALE PROOF-OF-WORK, LOCALIZARE ȘI AMPRENTARE

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI de către

Ing. Marius-Simion Cristea

Conducător ştiinţific: prof.univ.dr.ing. Nicolae Robu
Referenţi ştiinţifici: prof.univ.dr.ing. Mircea Stelian Petrescu
prof.univ.dr. Dana Petcu
prof.univ.dr.ing. Ioan Silea

Ziua susţinerii tezei: 15.02.2013.

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica” din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul Departamentului de Calculatoare și Tehnologia Informației al Universităţii „Politehnica” din Timişoara.

Aș dori să mulțumesc tuturor celor m-au ajutat și m-au inspirat pe parcursul stagiului doctoral.

Doresc să-i mulțumesc coordonatorului meu științific prof. univ. dr. ing. Nicolae Robu pentru sprijinul pe care mi l-a acordat pe întreaga perioada a stagiului.

Sunt în mod special recunoscător domnului ș.l. dr. ing. Bogdan Groza. El a fost sursă constantă de sfaturi și sprijin în toată această perioadă. Fără sprijinul său, probabil această lucrare nu ar fi existat, pot spune că majoritatea muncii și a rezultatelor mele sunt datorate dânsului. Pe perioada ultimilor ani, dl. Groza a fost părintele meu științific și îi sunt cu adevărat recunoscător pentru încrederea ce a investit-o în mine.

Nu în ultimul rând aș dori să le mulțumesc prietenilor, în special Iarinei, surorii mele Manuela și lui Marius, și colegilor mei, pentru sfaturile lor, pentru că au reușit să mă mai sustragă din când în când de la muncă, pentru că și-au alocat timp să participe la sondajele mele și pentru că m-au înțeles atunci când eu nu am putut sa-mi aloc timp ca să fiu alături de ei.

Cele mai sincere mulțumiri doresc să le aduc familiei mele pentru toată dragostea și sprijinul ce mi le-au acordat pe parcursul vieții. Părinții mei au reușit să zidească în mine încrederea necesară pentru a nu renunța niciodată să duc la capăt ceea ce mi-am propus.

Timişoara, Februarie 2013 Marius-Simion Cristea

Cristea, Marius-Simion

Tehnici pentru sporirea rezistenței în fața atacurilor DoS și DDoS prin protocoale proof-of-work, localizare și amprentare

Teze de doctorat ale UPT, Seria 14, Nr. 14, Editura Politehnica, 2013, 112 pagini, 39 figuri, 6 tabele.

ISSN: 2069-8216
ISSN-L: 2069-8216
ISBN: 978-606-554-629-5

Cuvinte cheie: DoS, spam, inginerie socială, puzzle criptografic, amprentare.

Rezumat:

În contextul răspândirii din ce în ce mai largi a atacurilor de tip DoS (Denial of Service) și a domeniilor vizate de acestea (politic, financiar, divertisment, rețele sociale), cu costuri de imense pentru cei vizați, teza de doctorat realizează o analiză a impactului pe care aceste atacuri îl pot avea asupra sistemelor de conducere automată și asupra utilizatorilor serviciilor de email sau de e-banking. Deoarece și cele mai performante sisteme au probleme în fața atacurilor DoS, lucrarea vine cu contramăsuri, ce pot fi aplicate în cadrul serviciilor web în general și serviciilor de email în particular.

Soluțiile propuse în lucrare se bazează pe puzzle-uri criptografice, ce au fost folosite pentru protecția serverelor web și a combaterii spam-ului. Deoarece puzzle-urile criptografice au o aplicabilitate limitată, pentru a le spori eficiența, în teză s-a folosit calibrarea prin localizare. Pentru identificarea adversarilor în rețea s-a folosit amprentarea fizică, aceasta mai permite și legarea puzzle-urilor de caracteristicile fizice ale echipamentului adversarului. Pentru ca soluțiile propuse în teză să aibă o aplicabilitate ridicată, s-a urmărit ca orice contramăsură propusă să implice modificări minime asupra infrastructurilor deja existente și consacrate în practică, iar impactul computațional asupra utilizatorilor legitimi să fie minim.

Cuprins

Notații, abrevieri, acronime

adv	- adversar
AP	- Access Point (punct de access)
ACK	- Acknowledgement
BSSID	- Basic Service Set Identifier
$cert_A$	- Certificatul digital al entității A
Cnp	- cod numeric personal
d(A,B)	- distanța de la A la B
DDoS	- Distribute Denial of Service
DNS	- Domain Name System
DoS	- Denial of Service
GNP	- Global Network Positioning
h(x)	- funcția hash aplicată lui *x*
HTML	- HyperText Markup Language
IDS	- Intrusion Detection System
IMEI	- International Mobile Station Equipment Identity
IMAP	- Internet Message Access Protocol
ISP	- Internet Service Provider
LBA	- Landmark based (bazat pe repere)
LM	- Landmark (reper)
MAC	- Media Access Control
MD5	- Message Digest ver. 5
MIB	- Management Information Base
Mitb	- Man-in-the-Browser
MITM	- Man-in-the-Middle
NAT	- Network Address Translator
NF-IMEI	- IMEI nefalsificabil fizic
NPS	- Network Positioning System
NTP	- Network Time Protocol
N_x	- Nodul x
OCR	- Optical Character Recognition
OPC	- OLE for Process Control
P(A)	- poziția lui A
PC	- Personal Computer
PI	- Proporțional Integrator
PIC	- Practical Internet Coordinates
PIN	- Personal Identification Number
PKI	- Public-key infrastructure
PLC	- Programmable Logic Controller
PLS	- Public Localization Service (Serviciu Public de Localizare)
PMK	- Pairwise Master Key
POP3	- Post Office Protocol, ver. 3
Ppm	- părti pe milion
Pr[x]	- probabilitatea ca evenimentul *x* să se întâmple
pwd	- parolă (password)

RFC	- Request for Comments
RNS	- Rhythmic Nonce Servers (Server de Nonce-uri Ritmice)
RTT	- Round Trip Time
SHA	- Secure Hash Algorithm
$Sign_A(m)$	- Semnătura entității *A* asupra mesajului *m*
SMTP	- Simple Mail Transfer Protocol
SO	- Sistem de Operare
SP	- Step (treaptă)
SSID	- Service Set IDentifier
SSL/TLS	- Secure Sockets Layer/Transport Layer Security
SYN	- Synchronize
TCP/IP	- Transmission Control Protocol/Internet Protocol
USB	- Universal Serial Bus
VI	- Virtual Instrument (Intrument Virtual)
WPA/WPA2	- Wi-Fi Protected Access/Wi-Fi Protected Access ver. 2
ZOH	- Zero Order Hold (Reținere de Ordinul 0)
ZTIC	- Zone Trusted Information Channel
$\varphi(n)$	- funcția Euler Phi
$\|\|$	- concatenare

Lista de tabele

Lista de figuri

Lista de secvențe de cod

Context și contribuție

În contextul răspândirii din ce în ce mai largi a atacurilor de tip DoS (Denial of Service) și a domeniilor vizate de acestea (politic, financiar, divertisment, rețele sociale), cu costuri imense pentru cei vizați, lucrarea de față dorește în primul rând să evalueze impactul pe care aceste atacuri îl pot avea asupra sistemelor de conducere automată și asupra utilizatorilor serviciilor de email sau de e-banking. Mai apoi să propună contramăsuri ce pot fi aplicate în cadrul serviciilor web, în general, și serviciilor de email în particular.

Lucrările publicate ale autorului pornesc de la puzzle-uri criptografice folosite pentru protecția serverelor web și a combaterii spam-ului. Dar, puzzle-urile criptografice au o aplicabilitate limitată datorită discrepanțelor dintre puterea de calcul a clienților legitimi și a rețelelor de tip botnet ale adversarilor. Pentru a le spori eficiența acestora, se propune calibrarea prin intermediul localizării în sistemul de coordonate Vivaldi, algoritmul de localizare fiind modificat în teză astfel încât să asigure localizare sigură. Dar și localizarea, în sisteme de coordonate virtuale, are limitele ei, de exemplu nodurile din aceeași rețea par mai apropiate decât nodurile din locații fizice apropiate, dar din rețele diferite. Astfel, în cele din urmă, identificarea intrușilor este realizată prin amprentare fizică, o metodă mult mai precisă de localizare a nodurilor în rețea, și se recomandă folosirea amprentării pentru asocierea unui puzzle cu o caracteristică fizică a echipamentului. Pentru ca soluțiile propuse să aibă o aplicabilitate ridicată s-a urmărit ca orice contramăsură propusă să implice modificări minime asupra infrastructurilor deja existente și consacrate în practică, iar impactul computațional asupra utilizatorilor legitimi să fie minim.

1. Introducere

1.1. Tema în contextul atacurilor DoS

Atacurile de tip DoS (*Denial of Service*) sau DDoS (*Distribute Denial of Service*) reprezintă o metodă prin care resursele computaţionale sunt epuizate de adversari în defavoarea utilizatorilor legitimi ale acestor resurse. Deşi metodele prin care aceste atacuri se realizează sunt foarte variate, aceste forme de atac sunt orientate în general spre a face o pagina web sau un serviciu web să funcţioneze greoi sau să nu funcţioneze deloc. Principalele ţinte ale atacurilor DoS sunt serverele web pentru servicii bancare, sau servicii de rezolvare a numelor în Internet (Domain Name System - DNS). Aceste atacuri sunt realizare de obicei prin cereri simple, repetate către staţiile ţintă, până când acestea vor începe să răspundă foarte greu la utilizatorii legitimi sau nu vor mai putea răspunde deloc.

Atacurile de tip DoS reprezintă o violare a politicilor de utilizare acceptabilă a resurselor tuturor furnizorilor de servicii Internet, în măsura în care sunt realizate de adversari, care nu sunt utilizatori legitimi sau cu bune intenții. De asemenea, aceste atacuri reprezintă şi o violare a legilor majorităţii statelor. De exemplu, în Marea Britanie persoanele implicate într-un atac DoS pot fi pedepsite cu până la 10 ani de închisoare [34]; la fel în SUA atacurile DoS reprezintă o infracțiune federală pedepsită cu închisoare.

Aceste atacuri vizează o varietate de ţinte din diferite domenii, așa cum arată exemplele:

- Politic (atacul din 2008 asupra site-ului guvernului Georgian, atacul din 2009 asupra site-ului guvernului Iranian);
- Financiar (atacul din februarie 2001 asupra serverului departamentului de finanţe al guvernului Irlandez);
- Divertisment (atacurile din februarie 2007 asupra serverelor pentru jocurile *Return to Castle Wolfenstein, Halo, Counter-Strike*);
- Reţele sociale (Atacurile din 6 august 2009 asupra reţelelor de socializare *Facebook, Twitter, Livejournal şi Google*).

Datorită accesibilităţi acestor tipuri de atac, a domeniilor foarte diferite în care pot apărea și a costurilor asociate, măsuri de prevenire eficientă a atacurilor DoS sunt necesare.

O atenție importantă trebuie acordată disponibilității sistemelor de comandă și control wireless. Aceste echipamente se considerau sigure în fața atacurilor cibernetice, deoarece rulau într-un mediu izolat, dar odată cu descoperirea viermelui W32.Stuxnet [35], în centralele atomice din Iran, lucrurile s-au schimbat radical în acest domeniu. Stuxnet infectează sistemele Windows și poate reprograma controlerele industriale. Complexitatea acestui vierme demonstrează faptul că au fost investite resurse imense în acest tip de atac. Atacurile cu un astfel de vierme

pot conduce la pagube de dimensiunea unui atac armat, cu costuri mult mai reduse de partea atacatorilor.

Spam-ul reprezintă o metodă electronică de a trimite mesaje nesolicitate și este un atac cunoscut de mai bine de două decenii. Mesajele de tip spam fiind trimise prin intermediul reţelelor *zombie*, reţele formate din computere personale infectate cu viruşi sau viermi. Spam-ul rămâne o metodă profitabilă din punct de vedere economic deoarece firmele care fac spam nu au alte costuri cu publicitatea decât cele legate de menţinerea listelor de email-uri. Deoarece metoda de transmitere a mesajelor de tip spam este una relativ simplă, numărul de spameri este foarte numeros, iar volumul de mesaje nesolicitate a crescut enorm. Costul traficului cu mesaje spam crescând continuu, deoarece ISP-urile trebuie să mărească capacitatea liniilor sale, pentru a face faţă traficului tot mai mare de date. Mesajele de tip spam au condus la dezvoltarea unei adevărate industrii de colectare a adreselor de email, pentru crearea de liste cu adrese email folosite pentru distribuirea de spam. Colectarea adreselor de email este realizată, de obicei, prin intermediul site-urilor de tip reţea socială, site-uri ce conțin, pe lângă adrese de email, detalii cu privire la preferințele utilizatorilor (informații ce pot fi folosite în scop de marketing). În general, recoltarea de adrese de email este facilitată de utilizatorii care nu citesc cu atenţie acordul care şi-l dau în momentul în care îşi creează un cont pe un anumit site.

Mesajele de tip spam (sau bulk) au apărut în mijlocul anilor ’90, volumul acestora a crescut exponenţial ajungând în prezent la 80-85% din totalul de email-uri trimise. Comisia pentru piaţa internă a Uniunii Europene a estimat în 2001 că spam-ul îi costă pe utilizatorii de internet 10 miliarde € anual. Autorităţile din California au estimat că mesajele de tip spam au cauzat pierderi de 17 miliarde $ organizaţiilor din SUA în 2007. Spam-ul include și costuri legate de solicitarea resurselor calculatoarelor, a reţelelor şi costul în timp uman legat de distragerea atenţiei de către aceste mesaje nesolicitate.

În plus spam-ul mai implică o mare varietate de costuri, variind de la costurile pentru generarea de spam, la costurile legate de combaterea acestor mesaje. Pe lângă aceste costuri, mesajele de tip spam contribuie la realizarea unor infracţiuni, cum ar fi: furt financiar (inginerie socială), furt de identitate, furt de proprietate intelectuală, răspândirea viruşilor, pornografia infantilă, înşelăciune, terorism. Datorită riscurilor de securitate şi a costurilor mari implicate de mesajele de tip spam nevoia filtrării acestor tipuri de mesaje este clară. Ingineria socială reprezintă o metodă prin care utilizatorii de servicii web (e-banking, email, rețele sociale) sunt păcăliți să divulge informații personale. În contextul dezvoltării soluțiilor de tip e-banking, creșterea accesibilității acestora trebuie să primească o atenție deosebită datorită acestor atacuri de tip inginerie socială.

Atacurile de tip inginerie socială s-au răspândit foarte mult în ultima vreme deoarece atacatorii nu au nevoie de cunoștințe foarte avansate în domeniul software-ului sau a rețelelor de calculatoare ci se bazează pe vulnerabilitățile existente în arhitectura Internetului și pe naivitatea utilizatorilor. De aceea aceste tipuri de atacuri trebuie identificate și oprite înainte ca utilizatorii să fie afectați.

Ținta acestor atacuri, de cele mai multe ori, o reprezintă frauda electronică, așa că este important ca utilizatori să cunoască ce vulnerabilități au soluțiile de tip e-banking existente pe piață pentru a nu cădea în plasa atacatorilor.

1.2. Obiective

Obiectul principal al acestei teze este de a aduce contribuții noi în domeniul securității, contribuții orientate spre protecția DoS. Contribuțiile prezentate în această lucrare sunt împărțite în două categorii, ce urmează a fi descrise pe scurt. Toate contribuțiile sunt realizate în scenarii practice, iar soluțiile propuse pot fi ușor implementate.

Prima categorie este cea a studiilor de caz. Scopul acestor studii este de a evidenția pericolul atacurilor DoS, fiind prezentate atacuri posibile în domeniul controlului automat și în domeniul e-banking-ului.

Interesul pentru securitate în domeniul controlului automat, a început să crească odată cu apariția virusului W32.Stuxnet. Până în acel moment punându-se rareori problema securității în aplicațiile de reglare automată. Lucrarea de față urmărește să prezinte vulnerabilități și atacuri posibile asupra echipamentelor wireless industriale de top, precum și implicațiile acestor atacuri în scenarii reale de conducere automată.

În domeniul aplicațiilor de e-banking interesul pentru securitate este clar mai mare, totuși, soluțiile existente nu sunt adaptate la atacurile moderne sau nu țin cont de preferințele utilizatorilor. Studiul prezentat în această lucrare urmărește să identifice vulnerabilitățile prezente în serviciile de e-banking disponibile pe piața românească. Analiza pieței românești este importantă din două motive:

- soluțiile disponibile pe piața românească de e-banking provin de la furnizori multinaționali de serviciu bancare (BRD, ING, Raiffeisen, etc.);
- România este sursa numărul unu de fraudă electronică, pe plan mondial și a doua țară emitentă de carduri de credit folosite pentru a comite fraudă financiară [4].

Studiul este completat și de o analiză a conștientizării de către utilizatorii de e-banking, a pericolului reprezentat de atacurile DoS și inginerie socială. Rezultatele studiului urmând să furnizeze informații folositoare furnizorilor de servicii de e-banking, informații ce pot contribui la îmbunătățirea securității aplicațiilor lor și la o mai bună satisfacere a nevoilor de securitate și uzabilitate a clienților.

A doua categorie este cea a contramăsurilor DoS, aceste contramăsuri sunt bazate pe metode proof-of-work (dovezi de calcul computațional) implementate sub forma puzzle-urilor criptografice. Puzzle-urile criptografice au rolul de a contrabalansa balanța computațională dintre adversari și țintele acestora. În această cercetare s-a dorit să se studieze eficiența acestor puzzle-uri în protejarea unui

server web în fața unui atac DoS, precum și impactul acestor puzzle-uri asupra unor clienți legitimi.

Eficiența puzzle-urilor criptografice va fi analizată și în contextul serviciilor de email, dar pentru aceasta se vor folosi puzzle-uri criptografice de tip time-lock. Pentru ca soluția să fie eficientă aceasta trebuie să aducă modificări minimale infrastructurii existente, să afecteze puternic profitul spamer-ilor iar impactul asupra clienților legitimi să fie insesizabil.

Deoarece eficiența puzzle-urile criptografice este redusă atunci când sunt folosite singure, am ales ca acestea să fie calibrate în funcție de locația adversarilor. Localizarea va fi realizată folosind algoritmul Vivaldi. Pentru ca localizarea să fie eficientă, în contextul identificării surselor de DoS, algoritmul Vivaldi va fi modificat astfel încât acesta să ofere o localizare sigură.

Deoarece localizarea are limitele ei se va evalua folosirea amprentării ca tehnică de identificare directă a intrușilor, în lucrare prezentându-se o metodă de amprentare bazată pe alunecarea ceasurilor de timp real, disponibile pe toate nodurile dintr-o rețea.

La finalul lucrării vor fi prezentate concluziile legate de cercetarea efectuată în domeniul DoS.

1.3. Structură

Lucrarea este structurată pe 5 capitole. Primul capitol prezintă contextul lucrării reliefând importanța măsurilor DoS. Capitolul 2 începe prin a face o trecere în revistă asupra primitivelor criptografice folosite în teză, urmând ca mai apoi să prezinte structura atacurilor DoS. Vor fi prezentate structurat și principalele metode de prevenire și combatere a acestora. Capitolul continuă cu metodele de localizare bazate pe coordonate virtuale existente. În cele din urmă, sunt descrise tehnicile de amprentare folosite în prezent. Capitolele 3,4,5 arată contribuțiile autorului.

Capitolul 3 cuprinde două studii de caz, în care se prezintă consecințe ale atacurilor DoS sau ale atacurilor DoS combinate cu alte atacuri (cum ar fi ingineria socială). Atacurile prezentate sunt în domeniul reglării automate și al e-banking-ului, domenii de actualitate în securitate.

În capitolul 4 sunt propuse contramăsuri la atacuri DoS, aceste măsuri vizează serverele web și serviciile de email. Contramăsurile sunt bazate pe tehnici proof-of-work ce pot fi calibrate prin localizare. În finalul capitolului se discută o metodă de amprentare fizică ce poate fi folosită pentru detecția intrușilor.

Capitolul 5 cuprinde concluziile acestei lucrări, subliniind contribuțiile aduse de autor în domeniul protecției DoS și sumarizează activitatea acestuia pe perioada doctoratului.

2. Atacuri DoS şi DDoS

Conform [95] un atac *Denial-of-Service* (DoS) sau un atac *Distributed Denial-of-Service* (DDoS) poate fi definit ca un atac ce vizează disponibilitatea unui calculator sau a unei reţele de calculatoare astfel încât acesta să nu mai poată furniza în mod prompt serviciul pentru care a fost creat.

Atacurile de tip DoS reprezintă o violare a politicilor de utilizare a rețelelor tuturor furnizorilor serviciilor de internet și chiar a legilor unor țări.

2.1. Atacuri DoS

Putem vorbi de un atac DoS atunci când accesul la un calculator sau la o reţea a fost blocat sau degradat în mod intenţionat de către acţiunea maliţioasă a unui utilizator.

Majoritatea atacurilor DoS vizează lăţimea de bandă a reţelei sau conectivitatea. Pentru a epuiza lăţimea de bandă, atacatorul creează un flux mare de date în reţea astfel încât un utilizator legitim nu mai poate utiliza serviciul oferit de reţea, deoarece resursele ei devin epuizate. Un atac asupra conectivităţii se realizează printr-un număr mare de cereri asupra unui server, astfel încât acesta nu va mai putea răspunde cererilor utilizatorilor legitimi, deoarece resursele sale vor fi ocupate de cererile atacatorului.

Atacurile DoS pot fi clasificate în 5 categorii [59], după cum se poate vedea în Figura 1.

Atacurile DoS asupra ***nivelului echipamentelor de reţea*** se bazează pe probleme în softul echipamentelor de reţea. Cum ar fi, de exemplu, routerele de tip Cisco 7xx [16] care pot fi atacate prin utilizarea de parole foarte lungi la conexiunea telnet. Pe lângă probleme la nivelul softului un atacator mai poate încerca şi epuizarea resurselor hardware ale echipamentelor de reţea. Un exemplu de asemenea atac utilizează pachete ce au toate opţiunile activate şi bombardează routerele cu astfel de pachete.

Atacurile la ***nivelul Sistemului de Operare (SO)*** se bazează pe exploatarea implementării protocoalelor de reţea în SO. Un astfel de atac este *Ping of Death* în cadrul căruia atacatorul trimite pachete ICMP de tip *echo request* de dimensiuni mai mari decât cele standard pentru pachete IP (2^{16}-1). Un astfel de pachet este fragmentat şi reasamblat de către destinatar. În cazul în care se transmit mai multe pachete de acest fel, bufferul (memoria tampon) a SO poate fi depăşit, ducând astfel la prăbuşirea SO.

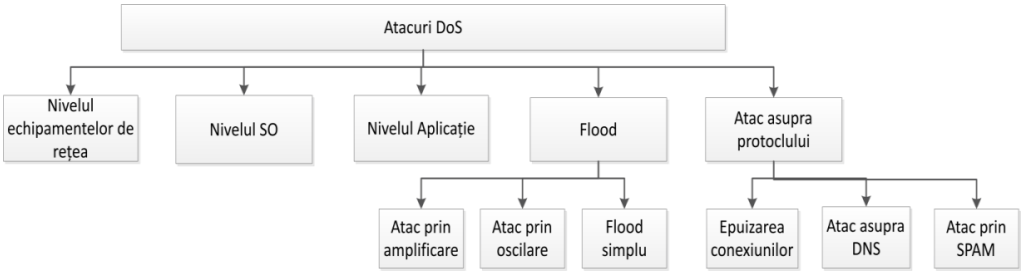


Figura 1 Clasificarea atacurilor DoS [59]

Atacurile la ***nivelul aplicaţie*** se bazează pe exploatarea unor vulnerabilităţi ale software-ului ce rulează pe maşina ţintă. Un exemplu de asemenea soft este aplicaţia *finger*. Aplicaţia finger primeşte un argument de forma *username@host1@host2@...@hostN*, cererea finger va fi transmisă la *hostN* iar de aici va fi redirecționată spre stânga până la *host1*. Un atacator poate specifica un argument de forma *username@host1@host1@...@host1* ce va conduce la executarea rutinei finger recursiv pe *host1* putând conduce la epuizarea resursei acestuia [53].

Atacurile de tip ***flood*** pot fi:

- ***Atacuri prin amplificare****,* un astfel de atac este s*murf attack*. Acest tip de atac se realizează prin falsificarea adresei sursă a unui pachet ICMP, astfel încât aceasta să conţină adresa victimei ca şi sursă şi o adresa de broadcast (x.x.x.0 sau x.x.x.255) ca şi destinaţie. Reţeaua către care acest pachet este trimis poartă numele de *amplificator*, deoarece toate staţiile din această reţea vor trimite un mesaj de tip *echo reply* către adresa victimei.
- ***Atacuri prin oscilare****,* un atac de acest fel se poate baza, de exemplu, pe serviciile *chargen* şi *echo*. Când un pachet este trimis către portul pe care serviciul chargen rulează (19), serviciul răspunde cu un şir de caractere. Serviciul echo, în schimb, răspunde cu aceleaşi date care le-am primit (pe portul 7). Un atacator poate trimite un pachet în care să specifice ca şi port sursă portul 7 iar ca şi adresă sursa adresa unei gazde pe care rulează serviciul echo. Astfel, se va produce o oscilaţie de date între maşina pe care rulează serviciul chargen şi maşina pe care rulează serviciul echo [54]
- ***Flood simplu***, acest tip de atac este cel mai răspândit atac DoS și se realizează foarte uşor prin utilizarea unor aplicaţii ce permit *flood ping*, acestea trimit pachete ICMP la rata de transfer maximă la care serverul către care sunt destinate poate răspunde. Atacatorul poate falsifica adresa sursă făcând acest tip de atac foarte greu de identificat. Acest tip de atac este întâlnit foarte des și în cadrul atacurilor de tip DDoS.

Plecând de la implementări practice în [43] se arată că majoritatea atacurilor informatice sunt posibile datorită unor probleme în proiectarea și implementarea protocoalelor, în cazul DoS ***atacurile asupra protocoalelor*** pot fi*:*

- ***Atacuri prin epuizarea conexiunilor***, acest atac se datorează faptului că multe protocoale orientate pe conexiune au nevoie de un server pentru a menţine starea imediat după ce s-a primit o cerere de conexiune, dar înainte ca această conexiune să fie stabilită. Un atac prin epuizarea conexiunilor este atacul SYN flooding, acesta se realizează prin trimiterea unui mesaj de tip SYN de către atacator la un server fără ca atacatorul să completeze cel de-al treilea pas al *handshake-ului*, aceasta conducând la ocuparea unei părți din memoria serverului până când conexiunea expiră (adesea după 75 sec). Deoarece aceste conexiuni „semi-deschise" ocupă memorie, SO limitează numărul lor, astfel atacatorul va lansa o serie de cereri de conectare către server, până când va atinge această limită, făcând ca orice altă cerere de conectare să fie respinsă [25].
- ***Atacuri asupra DNS****,* acest atac a fost posibil datorită faptului că primele versiuni de BIND nu verificau corectitudinea răspunsului unei interogări de nume. Astfel, dacă un server de nume va primi un răspuns de la un server de nume corupt, acesta va stoca răspunsul în memoria cache. Un atacator ce deţine un server de nume poate răspunde la o astfel de cerere cu adresa site-ului său, urmând ca serverul de nume vulnerabil să stocheze acest răspuns şi să furnizeze clienţilor săi adresa atacatorului. [26].
- ***Atac prin SPAM****,* acest atac poate fi văzut ca un atac de tip DoS asupra protocoalelor de email (SMTP, POP3, IMAP) deoarece traficul cauzat de mesajele de tip spam poate duce la epuizarea resurselor reţelelor de calculatoare implicate în furnizarea serviciilor de email. De asemenea mesajele de tip spam contribuie şi la suprasolicitarea serverelor de mail, acestea trebuind să proceseze un număr tot mai mare de mesaje nesolicitate. Atacul prin SPAM este văzut ca un atac DDoS datorită *zombie-lor* implicaţi în acest atac.

2.2. Atacuri DDoS

Conform [95] un atac DDoS (Distributed DoS) este un „atac ce utilizează mai multe noduri din rețea pentru a lansa un atac DoS coordonat asupra uneia sau mai multor ţinte. Folosind tehnologia client/server, atacatorul este capabil să amplifice eficacitatea atacului DoS în mod semnificativ prin valorificarea resurselor mai multor calculatoare complice fără voia lor ce sunt folosite ca şi platforme de atac." Aceste calculatoare complice poartă numele de zombii sau agenți. Deoarece atacurile DDoS sunt practic atacuri DoS distribuite, acestea pot fi clasificate asemănător atacurilor DoS.

Atacurile de tip DDoS reprezintă cea mai avansată formă a atacului DoS şi se bazează pe arhitectura distribuită a Internetului. Deoarece proiectarea Internetului a fost orientată pe funcţionalitate şi nu pe securitate, designul Internetului deschide câteva oportunităţi pentru realizarea atacurilor de tip DDoS:

- **Securitatea internetului este puternic interdependentă.** Internetul reprezintă o comunitate uriașă în care se găsesc sisteme sigure și sisteme mai puțin sigure. Chiar dacă un utilizator își poate proteja propriul sistem nu îi poate forța și pe ceilalți utilizatori din Internet să facă la fel. Astfel, un atacator poate găsi suficiente noduri nesigure în internet pentru a le compromite în vederea realizării unui atac DDoS. Realizând un atac, folosind aceste stații compromise, puterea atacatorului crește semnificativ, astfel că oricât de sigur ar fi sistemul victimei, aceasta nu poate dispune de resursele adversarului pentru a face față unui astfel de atac. Rezistenţa sa în faţa unui atac DDoS depinzând așadar de securitatea celorlalte sisteme din Internet [51].
- **Puterea celor mulţi faţă de puterea celor puţini.** Atacurile coordonate şi simultane ale unor participanţi vor fi întotdeauna în detrimentul altora dacă resursele atacatorilor sunt mai mari decât cele ale victimelor.
- **Resursele internetului sunt limitate.** Orice entitate din Internet (gazdă, reţea, serviciu) are un număr limitat de resurse ce este consumat de mai mulţi utilizatori.
- **Informaţiile şi resursele sunt dispersate.** Majoritatea informaţiei necesare pentru asigurarea furnizării unui serviciu se află în staţii terminale, limitând nivelul de procesare în reţelele intermediare. În acelaşi timp dorinţa unei rate de transfer cât mai mari a condus la proiectarea unor canale de comunicaţie, în reţelele intermediare, cu o lăţime de bandă foarte mare, pe când reţelele terminale au atâta lăţime de bandă cât au nevoie. Astfel, un atacator poate utiliza resursele de bandă ale unei reţele intermediare pentru a trimite o mulţime de mesaje victimei.
- **Răspunderea nu este forţată.** În pachetele IP adresa sursă nu este validată, aceasta putând conduce la *atacuri prin falsificarea adresei sursă* – cum ar fi atacul *smurf* [12].
- **Controlul este distribuit.** Managementul Internetului este distribuit și fiecare reţea are propria politică de securitate. E imposibil să se impună o politică globală de securitate datorită problemelor de confidenţialitate ce pot apărea. Astfel, e adesea imposibil de investigat comportamentul traficului dintre reţele.

Arhitectura unui astfel de atac atac DDoS este compusă din următoarele 4 elemente, după cum se poate vedea în Figura 2:

- Atacatorul propriu-zis;
- Stăpânii – sisteme compromise, capabile să controleze mai mulţi agenţi;
- Agenţii sau zombii – sunt sisteme compromise pe care rulează un program capabil de a trimite un flux de pachete spre victima vizată. Aceste sisteme sunt exterioare reţelei victimei pentru a evita un răspuns prompt din partea acesteia şi sunt exterioare atacatorului pentru a împiedica identificarea acestuia;
- Victima.

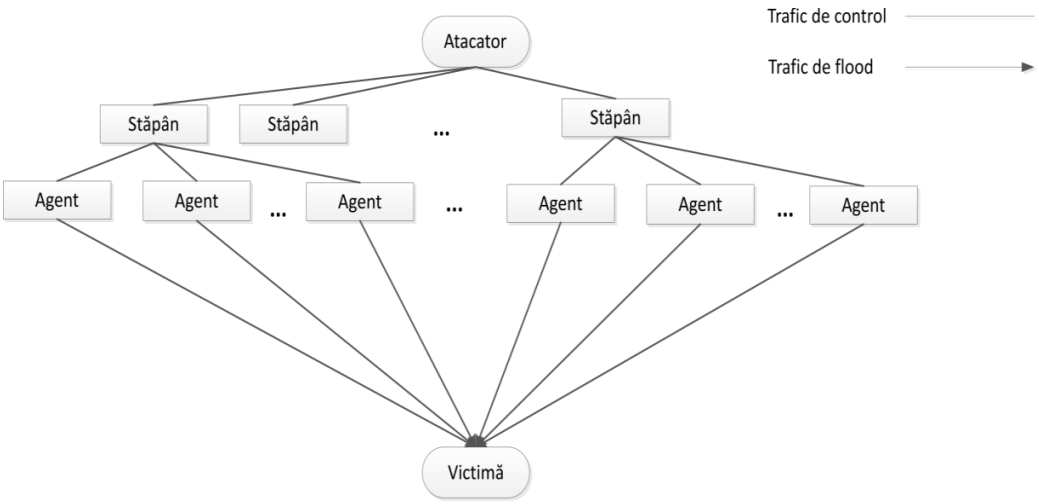


Figura 2 Arhitectura unui atac DDoS

2.3. Măsuri de prevenire şi combatere

Măsurile de prevenire şi combatere vor fi prezentate pe baza clasificării atacurilor DoS [59] (Figura 3):

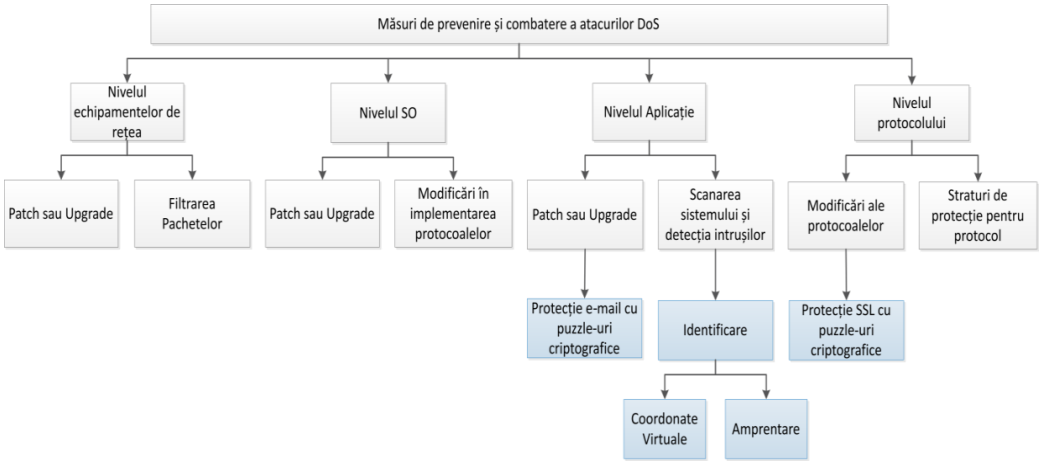


Figura 3 Clasificarea măsurilor de prevenire și combatere, clasificarea o extinde pe cea prezentată în [59]

2.3.1. Nivelul echipamentelor de reţea

Pe lângă patch-uri şi update-uri pentru rezolvarea problemelor software, filtrarea pachetelor reprezintă o metodă de combatere a atacurilor DoS.

Ca şi soluţii de filtrare avem:

- *Filtrarea la intrare,* aceasta presupune utilizarea unui router astfel încât acesta să blocheze accesul pachetelor cu o adresă nelegitimă în reţea. Mecanismul de filtrare la intrare propus în [36] presupune eliminarea pachetelor IP ce nu au un prefix specific domeniului routerului de intrare. Acest mecanism aduce rezultate foarte bune pentru combaterea atacurilor DoS prin falsificarea adreselor IP, dar câteodată [81] traficul ce provine de la agenţi mobili poate fi blocat de filtrarea la intrare.
- *Filtrarea la ieşire* [9]*,* acesta presupune faptul că doar anumite adrese de IP pot părăsi reţeaua. Filtrarea la ieşire nu previne atacurile DoS în interiorul reţelei dar contribuie la prevenirea epuizării resurselor în afara reţelei de origine a traficului.
- *Filtrarea distribuită bazată pe rute,* acest tip de filtrare propus în [77] foloseşte informaţia din tabela de rutare pentru a determina pachetele IP falsificate. Acest mecanism poate filtra o mare parte dintre pachetele IP falsificate, dar este dezavantajat de faptul că are nevoie de topologia globală a reţelei, acest lucru, în cazul Internetului, cauzând probleme de scalabilitate [78].
- *Filtrarea bazată pe istoricul adreselor IP (HIP),* reprezintă un mecanism introdus în [80] pentru prevenirea atacurilor DoS ce presupune ca routerul de la graniţa reţelei să admită doar pachete cu adrese IP dintr-o bază de date, baza de date fiind construită pe baza istoricului conexiunilor realizate prin router. Această metodă de filtrare are avantajul că nu necesită cunoaşterea globală a topologiei, dar, pe de altă parte, dacă atacatorul cunoaşte faptul că filtrul se bazează pe conexiuni realizate anterior, poate realiza astfel de conexiuni.

2.3.2. Nivelul SO

Pentru combaterea atacurilor de tip SYN (Synchronize) flooding în [3] este propusă o soluţie bazată pe SYN cookies, aceasta presupune evitarea alocării resurselor pentru o conexiune, până când această conexiune este completă. Utilizarea SYN cookies se datorează faptului că aproape toată informaţia de stare poate fi dedusă pe baza ultimului ACK (Acknowledgement), excepţie făcând opţiunea MSS (Maximum Segment Size) ce trebuie stocată în cookie.

O altă metodă de combatere a atacurilor DoS la nivelul SO este cea a receptorului leneş introdusă în [29]. Aceasta presupune ca pachetele să ajungă iniţial la cozile de aşteptare ale soclurilor (sockets) şi nu la coada IP partajată. Pachetele vor fi procesate cu prioritatea aplicaţiilor care le cer, iar procesarea la nivel de protocol nu are loc decât atunci când aplicaţia cere un anumit pachet. Aceasta conducând la reducerea procesării bazate pe întreruperi.

2.3.3. Nivelul aplicaţie

Există tot mai multe soluţii de tip IDS (Intrusion Detection Systems - sisteme de detecţie a intruşilor) folosite pentru a determina activităţile ilicite din reţea. Acestea pot utiliza detecţia anomaliilor sau detecţia abuzurilor.

Detecţia anomaliilor încearcă să identifice comportamente anormale în reţea faţă de standardele normale. Un sistem ce utilizează detecţia anomaliilor este NOMAD [96], acesta poate detecta anomalii făcând o analiză statistică a informaţiei din pachetele IP. Poate fi folosit pentru a analiza traficul dintr-o reţea locală, dar nu şi pentru analiza reţelelor distribuite.

O altă metodă de detecţie a atacurilor DoS utilizează MIB (Management Information Base - Baza de Informaţii de Management) a routerelor. În [10] se propune o metoda de detecţie a atacurilor DDoS bazată pe diferite tipare statistice a parametrilor regăsiţi în MIB. Această cercetare pare promiţătoare pentru detecţia anomaliilor în pachetele ICMP, UDP sau TCP, dar trebuie evaluată în reţele reale.

Un alt mecanism numit filtrare declanşată de congestie este propusă în [52]. Acest mecanism propune ca anumite pachete ce au cauzat o congestie să fie supuse unei analize statistice. Dacă se descoperă o anomalie routerul va filtra aceste pachete.

În [63] se propune o metodă de detecţie bazată pe data-mining pentru descoperirea tiparelor de atac. O îmbunătăţire a acestei metode aste adusă în [64].

În [69] este propus un sistem numit D-WARD ce urmează să identifice un atac DoS la sursă. Astfel, D-WARD este instalat pe routerele de graniţă a reţelelor şi monitorizează traficul ce trece prin ele, iar dacă se identifică o asimetrie în rata de pachete generată de un sistem intern reţelei, D-WARN va reduce rata acestuia de transfer. Sistemul are dezavantajul că de multe ori poate identifica un sistem legitim ca şi atacator.

În [41] se propune o structură de date euristică (MULTIPOS) ce are scopul de identifica adresele ce sunt implicate într-un atac DoS. Acest sistem are însă dezavantajul că necesită reconfigurarea roterului şi scheme de management ale memoriei noi şi nu poate detecta atacuri DoS în care sunt falsificate mai multe adrese IP de o singură sursă sau un atac DDoS în care sunt implicaţi mai mulţi zombii.

Detecţia abuzului - se bazează pe tipare bine cunoscute de abuzuri şi pe identificarea acestora. Tiparele de atac pot fi informaţii din pachetele de date, condiţii sau relaţii între evenimentele ce pot duce la un atac. Aceste tipare definesc semnătura atacului. Există mai multe instrumente ce pot realiza o detecţie bazată pe semnături: CISCO NetRanger (www.cisco.com), SecureNet PRO (http://www.intrusion.com/), IBM RealSecure (http://www.intrusion.com/), NFR-NID (http://www.checkpoint.com), Snort (http://www.snort.org/).

2.3.4. Load balancing şi replicare

Load balancing-ul reprezintă o metodă simplă ce le permite furnizorilor de servicii de reţea o modalitate de a creşte lăţimea de bandă existentă şi care previne căderea totală a serviciilor, în cazul unui atac, prin distribuirea sarcinilor spre nodurile neafectate. În plus, se poate realiza şi o replicare a serverelor în cazul în care unul din ele cade în urma unui atac DoS.

2.3.5. Nivelul standard al protocolului

Şi protocoalele existente pot fi îmbunătăţite prin extensii de securitate ce pot oferi protecţie DoS, dar, din păcate schimbările la nivelul protocoalelor pot fi foarte greu de pus în practică dacă implică schimbări şi la nivelurile superioare (modificări în SO, ale topologiei, echipamentelor, etc.). Un exemplu de astfel de extensie îl reprezintă cel adus de *puzzle-urile criptografice.*

Puzzle-urile criptografice reprezintă dovezi de calcul computațional (proof-of-work) nereutilizabile ce sunt propuse frecvent ca metodă de protecţie împotriva atacurilor DoS [7]. Ideea care stă la baza puzzle-urilor criptografice este următoarea: pentru ca un client să se conecteze la un server aflat sub atac, trebuie să rezolve un puzzle matematic. Dificultatea puzzle-ului va fi direct proporţională cu numărul de conexiuni iniţiate de client. Astfel, un client legitim nu va sesiza vreo diferenţa prea mare în timpul de conectare la un server, deoarece nu solicită multe conexiuni, pe când un atacator care face multe cereri pentru a bloca rețeaua, se va conecta din ce în ce mai lent.

Ideea puzzle-lor criptografice aparţine lui Merkle [68], dar Merkle a folosit puzzle-uri criptografice pentru stabilirea cheilor și nu pentru DoS. Puzzle-urile criptografice au fost folosite pentru a contracara atacurile asupra TCP/IP de Juels şi Brainard [57], care au menţionat că şi SSL poate fi protejat într-o manieră similară. Aura, Nikander şi Leiwo [1] au aplicat puzzle-urile criptografice la protocoalele de autentificare în general. Ei prezintă modul în care puzzle-urile criptografice propuse de Jules şi Brainard pot fi folosite pentru a preveni atacurile DoS. În lucrarea lor enunţă următorul principiu de design: *clientul trebuie să-şi utilizeze mai întâi resursele într-un proces de autentificare şi serverul trebuie să poată verifica clientul înainte de a-şi aloca propriile resurse.* Regula principală fiind ca în fiecare punct dinaintea autentificării, costul rulării protocolului pe client să fie mai mare decât costul rulării protocolului pe server. Costul pe client poate fi crescut în mod artificial prin faptul că i se cere să rezolve puzzle-uri criptografice ce sunt uşor de generat dar a căror dificultate de rezolvare poate fi ajustată la orice nivel. Serverul trebuie să refuze efectuarea oricăror operaţii criptografice complexe până când va verifica validitatea soluţiei oferite de client la puzzle-ul criptografic.

Dwork şi Naor au prezentat puzzle-urile criptografice ca şi soluţie generală pentru controlul folosirii resurselor, în special pentru a controla spam-ul [31]. Schema lor s-a dezvoltat într-o altă direcţie, motivată în primul rând de dorinţa lor de a se folosi o scurtătură în rezolvarea puzzle-urilor în cazul în care se utilizează o

parolă secretă. Frankyn şi Malkhi [38] au folosit puzzle-uri criptografice pe baza inversării funcţiilor *hash* puternice pentru a fi folosite în aplicaţii de monitorizare a traficului web. Rivest, Shamir şi Wagner [87] au propus şi ei o variantă de puzzle-uri criptografice în lucrarea lor din 1996.

O primă lucrare studiată legată de studiul puzzle-urilor criptografice pentru prevenirea atacurilor de tip DoS este cea a lui B. Waters, şi alţii [100]. În lucrarea lor se prezintă o metodă de distribuire a puzzle-urilor criptografice prin intermediul unui serviciu extern robust pe care ei îl numesc *bastion*. Multe servere se pot baza pe puzzle-urile distribuite de un singur bastion. Ei arată cum bastionul nu are nevoie să ştie ce servere se bazează pe serviciul oferit de el. În lucrarea propusă de ei bastionul se bazează pe o sursă de date aleatoare publică, şi nu pe un server dedicat. Soluţia lor contribuie în a face distribuirea puzzle-urilor să nu mai fie un punct de compromitere. Designul lor prezintă trei avantaje: în primul rând este mai rezistent la atacuri de tip DoS ţintite asupra mecanismului de generare a puzzle-lor în sine. În al doilea rând schema poate fi aplicată foarte uşor la nivelul IP, deşi funcţionează şi la nivele mai superioare ale stivei de protocoale. În a treilea rând puzzle-urile pot fi rezolvate off-line. În lucrare, autorii prezintă un prototip de implementare al abordării lor şi câteva rezultate experimentale.

Deşi ideea este relativ veche, implementările sunt puţine. O implementare a acestei idei a fost realizată de Dean şi Stubblefield [27]. Autorii au creat o implementare compatibilă TLS prin introducerea unei stări noi în handshake-ul de SSL ce este responsabilă de tratarea puzzle-lor criptografice. Pentru a analiza comparativ rezultatele în [27] s-au efectuat o serie de măsurători cu privire la încărcarea procesorului pe server în timpul unui atac DoS, atât atunci când sunt utilizate puzzle-uri criptografice dar şi când acestea nu sunt utilizate. În implementarea lor puzzle-urile criptografice sunt trimise către clienţi doar în momentul în care serverul se află sub atac, moment determinat pe baza încărcării serverului. Dificultatea puzzle-urilor fiind în funcţie de încărcarea serverului. În momentul în care serverul este sub atac acesta va trimite puzzle-uri criptografice la toţi clienţi, atât la cei legitimi cât şi la cei nelegitimi. Se observă în lucrarea lor că în acest mod serverul va reuşi să iasă de sub un atac de tip DoS efectuând mai puţine calcule criptografice complexe, reuşind în cele din urmă să servească clienţii legitimi, aceştia pierzând totuşi un anumit timp în rezolvarea puzzle-urilor criptografice.

2.4. Identificarea adversarilor

Este cunoscut faptul că un atacator poate modifica cu ușurință adresa unui agent compromis sau poate modifica mesajele trimise de acesta astfel încât să poată păcălii IDS-urile. Pe de altă parte găsirea unor noi locații de atac în Internet ar trebui să fie mai grea. De obicei atacatorii folosesc botnets (rețele de boți) pentru a trimite spam sau pentru a realiza atacuri DoS, aceste rețele fiind refolosite de-a lungul timpului pentru a comite mai multe atacuri [28]. De aceea localizarea boților poate reduce eficacitatea botnet-ului prin blocarea locațiilor malițioase.

2.4.1. Localizare folosind sisteme de coordonate virtuale

Una dintre cele mai simple metode de localizare într-o rețea este bazată pe măsurarea timpului de răspuns (*RTT*- round trip time) între nodurile rețelei prin transmiterea de pachete *ICMP* (Internet Control Message Protocol). Dar această metodă presupune foarte mult trafic în rețea, iar dacă rețeaua este foarte mare un nod poate percepe localizarea ca pe un atac de tip DDoS prin flood. Pentru a evita această situație au apărut diferite metode care încearcă să estimeze RTT-ul dintre două noduri.

Primele metode de estimare a RTT-ului de bazează pe *timpul de latență*, astfel pentru ca un nod să se localizeze în rețea trebuie să măsoare RTT-ul doar către anumite noduri din rețea (noduri de referință). Nodurile de referință vor măsura, la rândul lor, RTT-ul între ele.

Primul sistem ce a folosit estimarea RTT-ului a fost *IDMaps* [37]. Acest sistem folosea o topologie virtuală a rețelei stocata pe un server dedicat numit *HOPS*. Harta conținea părți din rețea (delimitate pe baza prefixului IP) și indicatori (*TS*). Fiecare indicator trebuia să măsoare RTT-ul către cel mai apropiat prefix din rețea și către celelalte TS-uri din rețea. IDMaps presupune că distanța dintre două noduri cu același prefix este neglijabilă. Pentru ca un nod N_a, aparținând prefixului de rețea *NA,* să estimeze RTT-ul spre nodul N_b, aparținând lui *NB*, trebuie să acceseze serverul HOPS și să calculeze:

$$RTT(N_a, N_b) \approx RTT(NA, TS_a) + RTT(TS_a, TS_b) + RTT(TS_b, NB).$$

În relația de mai sus TS_x reprezintă cel mai apropiat TS de *NX.*

Sistemul *King* [46] folosește de infrastructura DNS pentru a estima RTT-ul. King se bazează pe următoarele prezumții:

- Serverele DNS se află la o distanță neglijabilă de noduri;
- Interogări DNS recursive sunt permise în rețea.

Pentru a estima RTT-ul de la nodul *N*a*,* aparținând domeniului DN_a, spre nodul N_b, aparținând domeniului DN_b, nodul N_a trebuie să găsească adresa serverului DNS_b ce este autoritar peste N_b și să interogheze apoi DNS_b asupra unui nume arbitrar din *NB.* Numele arbitrar este necesar pentru a evita situația în care N_a cere o intrare cacheuită de DNS_a. Deoarece se presupune că serverele DNS sunt în apropierea nodurilor asupra cărora sunt autoritare, atunci durata acestui proces poate estima RTT-ul dintre N_a și N_b.

Algoritmul debarasării (binning) [86] folosește același principiu ca și IDMaps, dar în loc să grupeze nodurile în funcție de IP, le grupează în *coșuri* în funcție de RTT-ul lor față de nodurile de referință numite *repere (landmarks - LM)*. Nodurile din rețea trebuie să măsoare RTT-ul către toate LM-urile din rețea și să se poziționeze în coșul ce corespunde celui mai apropiat LM.

Deși aceste metode de estimare a RTT-ului folosind timpul de latență sunt mai eficiente decât măsurarea directă a RTT-ului, totuși prezintă trei dezavantaje:

- Încă sunt necesare numeroase cereri ICMP;
- Sunt soluții centralizate cu un singur punct avarie;
- Localizarea nu este foarte precisă.

Datorită acestor dezavantaje au început să apară sisteme ce folosesc *coordonate virtuale (sau sintetice)*. Aceste sisteme încearcă să rezolve problemele enumerate mai sus și pot fi de două tipuri: *centralizate* și *descentralizate.*

2.4.1.1. Sisteme de coordonate virtuale centralizate

Aceste sisteme folosesc un set de noduri specializate numite repere (de aceea aceste sisteme mai sunt cunoscute și ca sisteme de coordonate virtuale bazate pe repere – *LBA* – Landmark-Based Algorithm). Pentru ca nodurile să se poată localizeze într-un LBA, mai întâi reperele trebuie să-și determine coordonatele, iar apoi nodurile din rețea vor folosi coordonatele reperelor pentru a-și determina poziția.

În cazul *GNP* [74] reperele își măsoară RTT-ul dintre ele după care se face broadcast la matricea acestor RTT-uri. Reperele sunt poziționate în sistem, astfel încât eroarea dintre RTT-ul măsurat și distanța în sistemul de coordonate (eroarea de estimare) să fie minimă, această poziționare este realizată de un nod central din rețea. Când un nod nou intră în rețea, va măsura RTT-ul către toate reperele din rețea, va primi *matricea RTT-urilor reperelor* de la nodul central și își va determina poziția în sistemul de coordonate virtuale astfel încât eroarea de estimare să fie minimă. Pentru a funcționa corect GNP are nevoie de un număr relativ mic de repere (5-10), în schimb alegerea reperelor influențează în mod semnificativ precizia localizării.

Lighthouse (far) [83] reprezintă o extensie mai scalabilă a GNP, în care nodurile nu au nevoie de coordonatele reperelor, ci pot folosi coordonatele oricărui nod din rețea. Astfel, când un nod intră în rețea va primi o listă cu noduri alese în mod arbitrar (pe care nodul cel nou le va considera repere). Noul nod își va determina coordonatele locale în sistemul de coordonate determinat de reperele sale, iar apoi va primi o *matrice de transformare* de la un nod central, pentru a-și calcula coordonatele globale pe baza coordonatelor locale.

GNP a fost optimizat prin introducerea de *repere virtuale* [97]. Astfel, noua soluție optimiza modul în care se estimează distanța, iar reperele fixe nu mai erau necesare. Un nod N_a va avea nevoie de un set de repere ales aleatoriu, pentru a-și determina poziția. Pentru a calcula distanța spre un alt nod N_b, ce folosește un alt set de repere, N_a va trebui să translateze coordonatele lui N_b în propriul sistem de coordonate (bazat pe reperele lui N_a). Această translatare se realizează folosind o matrice de transformare obținută de la niște noduri specializate din rețea, numite *"chei" ("spanners").*

Un alt sistem de coordonate virtuale centralizat este *IDES* [66]. Ca și GNP, IDES folosește un set de repere fixe. Nodurile din rețea, pentru a-și determina coordonatele, trebuie să măsoare RTT-ul spre aceste repere, iar poziția lor va fi

determinată în funcție de valoarea RTT-ului și de coordonatele reperelor. Spre deosebire de GNP, IDES nu folosește distanța euclidiană pentru estimarea RTT-ului, ci folosește o *matrice de factorizare* (astfel de sisteme mai sunt cunoscute ca și *coordonate virtuale bazate pe produs scalar*). Nemaifiind utilizată distanța euclidiană, nodurile nu mai trebuie să respecte inegalitatea triunghiului (o condiție necesară pentru funcționarea corectă a celorlalte sisteme de coordonate virtuale). Dar, câteodată, algoritmul folosit de IDES poate returna o valoare negativă a distanței (RTT-ului) între două noduri, ceea ce nu este acceptabil.

2.4.1.2. Sisteme de coordonate virtuale descentralizate

Spre deosebire de sistemele de coordonate centralizate, cele descentralizate nu au nevoie de noduri referință. Într-un astfel de sistem de coordonate orice nod își poate determina poziția pe baza poziției oricărui alt nod din rețea și a RTT-ului către acel nod. Deoarece nu sunt necesare noduri specializate, într-un sistem de coordonate descentralizat, calculul computațional al poziției este distribuit în mod egal între nodurile din rețea.

NPS [73] este unul dintre primele sisteme de coordonate descentralizat. NPS a evoluat din GNP [74], dar acesta, spre deosebire de GNP, nu necesită un nod central pentru a distribui informația necesară calculării poziției, ci orice nod poate fi folosit pe post de reper de către un nod nou care intră în rețea și dorește să-și determine poziția. NPS se bazează pe o arhitectură ierarhică, nodurile reper sunt considerate noduri de nivel *0*, un nod este de nivel *L+1* dacă poziția sa a fost determinată pe baza unui nod de nivel *L.* Structura ierarhica este folosită pentru a obține un sistem consistent. Fără această structură ierarhică, alegând nodurile referință în mod aleatoriu, se poate ajunge la dependență circulară în rețea. NPS propune și un mecanism de identificare a nodurilor referință malițioase. Un nod referință este considerat malițios dacă poziția lui se schimbă frecvent sau dacă eroarea sa de estimare este prea mare. Totuși un nod ce nu este nod referință poate minți cu privință la poziția sa.

Sistemul *PIC* [17] propune o soluție descentralizată în care, de asemenea, fiecare nod existent în rețea poate fi folosit ca și reper de către un nod nou ce intră în rețea. PIC folosește algoritmul *Simplex* pentru calcularea coordonatelor unui nod și pentru reducerea erorii de estimare. Pentru a obține valori mici ale erorii de estimare atât pentru distanțe mici, cât și pentru distanțe mari, PIC folosește o strategie hibridă pentru alegerea nodurilor reper (strategia are nevoie de câteva noduri reper în apropierea noului nod ce intră în rețea). Pentru această strategie hibridă este prezentat un algoritm care determină nodurile din apropierea noului nod din rețea. PIC propune și o metodă pentru identificarea nodurilor reper malițioase din rețea, această strategie fiind bazată pe inegalitatea triunghiului. Totuși PIC nu este recomandat sistemelor foarte dinamice, deoarece reacționează foarte rapid la noi valori ale măsurătorilor, creând oscilații în sistemul de coordonate.

Big-Bang [91] și *Vivaldi* [24] sunt două sisteme de coordonate virtuale descentralizate dezvoltate pe baza analogii cu sistemele fizice. Primul este bazat pe

simularea efectului forță-câmp, iar al doilea pe baza aducerii resortului în starea de repaos. Starea în care energia sistemului este minimă corespunde coordonatelor cu eroare de estimare minimă. Deși algoritmul Vivaldi este destul de răspândit (mai ales în rețelele de tip bit torrent) are totuși o problemă: nodurile mai apropiate au o eroare de estimare mai mare decât nodurile depărtate. Sistemul *Pharos* [15] încearcă să rezolve problema algoritmului Vivaldi prin utilizarea a două tipuri de coordonate pentru un nod (N_a): o coordonată globală (P_a^{global}) și o coordonată locală (P_a^{local}), nodurile foarte apropiate unele de altele formând un grup. Gruparea nodurilor se realizează cu ajutorul unor noduri specializate numite *ancore*. Vivaldi va rula odată pentru nodurile din același grup, iar apoi va rula global. Astfel, distanța ($d(N_a,N_b)$) între două noduri (N_a și N_b) poate fi definită după cum urmează:

$$d(N_a,N_b) = \begin{cases} \mid P_a^{local} - P_b^{local} \mid, grup_a = grup_b \\ \mid P_a^{local} - P_b^{local} \mid, grup_a \neq grup_b \end{cases}.$$

Pentru a depăși problemele legate de inegalitatea triunghiului, s-a dezvoltat sistemul de coordonate virtuale descentralizat *Phoenix* [14]. Acesta, ca și IDES [66], este un sistem bazat pe produs scalar, dar spre deosebire de IDES, în cazul lui Phoenix nu poate ajunge la situația în care distanța dintre două noduri să fie negativă. Phoenix, însă, nu poate fi reprezentat într-un sistem Euclidian de coordonate.

2.4.2. Verificarea locației

Ideea verificării locației nu este nouă, dar a fost tratată mai ales în rețelele wireless, deoarece comunicația wireless între două noduri din rețea, se realizează în linie dreaptă, iar viteza comunicării wireless este limitată superior de viteza luminii.

Un nod dintr-o rețea wireless poate pretinde că este mai departe de un nod ce îl verifică (prin întârzierea răspunsului), dar nu poate pretinde că este mai aproape. Pentru a stabili, totuși, o limită superioară a distanței la care un nod se poate afla, s-au introdus protocoalele de *încadrare a distanței (distance bounding)* [8]. Acestea au fost implementate cu succes în rețelele wireless [11].

Tot în rețelele wireless a apărut protocolul de localizare sigură *echo* [89], acest protocol se folosește de *ultra-sunete* și de *durata-zborului* pentru verificarea localizării pe regiuni.

În Ethernet, folosind coordonate virtuale, se pot identifica nodurile de referință malițioase (după cum s-a prezentat mai sus) în cazul sistemelor NPS [73] sau PIC [17].

2.4.3. Amprentare

Prin amprentare se înțelege procesul prin care se poate identifica un echipament sau o clasă de echipamente fără cooperarea echipamentului care se

dorește a fi identificat. Amprentarea a evoluat de la identificarea unor clase de echipamente la identificarea fizică a unui echipament. Această evoluție s-a datorat, la început, nevoii de a evalua automat vulnerabilitățile software-ului ce rulează pe nodurile din rețea. O componentă esențială a acestei evaluări o reprezintă *amprentarea sistemului de operare*, deoarece pornind de la sistemul de operare, pe baza vulnerabilităților bine cunoscute ale fiecărui sistem, se poate verifica rezistența nodului la diverse atacuri specifice sistemului. Amprentarea sistemului de operare se poate realiza cu instrumente bine cunoscute care analizează: segmentele TCP (*nmap* –http://nmap.org/), răspunsurile ICMP (*X-Probe* - http://sourceforge.net/projects/xprobe/) sau cu ajutorul unui instrument mai avansat, cum ar fi RING [99], ce realizează o analiză temporală a conexiunilor TCP/IP.

Această amprentare poate fi și mai rafinată, existând tehnici de *amprentare fizică* a unui echipament din rețea. Amprentarea fizică se realizează pe bază unor proprietăți fizice ce sunt unice echipamentului sau a componentelor din structura sa. O astfel de proprietate o reprezintă alunecarea ceasurilor de timp real (*clock skew*). Această alunecare este constantă și unică fiecărui ceas ([70], [79]).

Deviația ceasurilor de timp real a fost folosită în [60] pentru identificarea nodurilor din rețea fără cooperarea acestora. În [60] este propusă o metodă pasivă cât și un semi-pasivă, aceste metode pot funcționa și atunci când nodul ce se dorește a fi amprentat se află în spatele unui NAT (Network Address Translator) sau firewall, metodele funcționând fără probleme chiar și atunci când nodul folosește NTP (Network Time Protocol) pentru sincronizarea ceasului. Pentru a implementa amprentarea fizică în [60] se exploatează câmpul *timestamp* al pachetelor TCP (vezi RFC 1323 [55]). Astfel, tot ce trebuie să facă identificatorul (cel care realizează amprentarea) este să captureze pachete TCP, aceasta se poate realiza observând pasiv pachetele care circulă prin rețea, metoda funcționând pentru majoritatea sistemelor de operare.

Nodurile ce folosesc un sistem de operare Windows nu completează câmpul timestamp din cadrul pachetelor TCP. Pentru a amprenta un astfel de nod, identificatorul trebuie să adopte o strategie semi-pasivă de amprentare, adică va trebui ca el să includă timestamp-ul în răspunsurile TCP către nodurile cu sistem de operare Windows (ceea ce intră în contradicție cu RFC 1323 [55], care precizează că dacă un nod primește un pachet TCP fără câmpul timestamp completat, acesta va trebui să răspundă, la rândul lui, cu un pachet TCP ce are acest câmp necompletat). Primind un răspuns ce include câmpul timestamp în pachetul TCP, nodurile bazate pe Windows vor include și ele, în continuare, acest câmp în comunicarea TCP cu identificatorul, putând fi astfel amprentate.

T. Kohno, A. Broido, and K. C. Claffy [60] propun și exploatarea câmpului timestamp inclus în pachetele ICMP (RFC 792 [84]), dar această opțiune nu este analizată de autori deoarece metoda propusă în lucrarea lor este destinată rețelelor WAN, iar pachetele ICMP pot fi (și sunt de cele mai multe ori) blocate de către firewall-urile din WAN.

Această amprentare are nevoie de foarte multe pachete pentru a funcționa (aproximativ 10000) iar o metodă pasivă ca cea prezentată în [60] poate dura și câteva ore. Pentru a reduce timpul în care se realizează identificarea, în [56] este propusă o metodă de amprentare fizică bazată tot pe alunecare ceasurilor de timp real. Această alunecare este determinată din timestamp-urile funcției de sincronizare temporală, prezente în pachetele de management a rețelelor wireless. În rețelele wireless aceste timestamp-uri sunt setate de către punctele de acces (AP) în *pachetele de semnalizare (beacons)*. Aceste pachete de semnalizare sunt distribuite periodic (10-100 cadre pe secundă) în rețea de către AP cu scopul de a face cunoscut SSID-ul (Service Set IDentifier) rețelei clienților wireless. Astfel, se pot captura, în mod pasiv, foarte multe timestamp-uri într-un timp considerabil mai mic (16.6 min – 1.66 min pentru 10000 cadre). Deoarece distribuirea pachetelor de semnalizare poate fi blocată de către administratorul rețelei, în [56] se propune și o metodă de captură activă. Astfel, un client wireless poate sonda AP-ul, iar acesta dacă este activ va răspunde cu un cadru ce conține și timestamp-ul AP-ului. Deoarece timestamp-urile AP-ului au o rezoluție de 1ms și prin această metodă activă se pot captura foarte multe timestamp-uri într-un interval de timp relativ scurt (durata capturii depinde de viteza cu care clientul wireless lansează cereri către AP).

Odată capturate pachetele, la fel ca și în [60], se folosește programarea liniară (metoda propusă în [70]) pentru determinarea alunecării ceasului de pe AP. Pe lângă metoda programării lineare, în [56] se mai folosește și metoda celor mai mici pătrate pentru calculul alunecării, metodă ce nu este tolerantă la zgomot.

La fel ca și în cazul exploatării timestamp-urilor din pachetele TCP [60] și în [56] metoda funcționează chiar dacă AP-ul folosește NTP pentru sincronizarea ceasului de timp real. Amprentarea este utilizată în [56] cu scopul de a identifica AP-uri false folosite ca și honeypot-uri pentru capturarea de date senzitive de la clienții wireless ce se conectează la acestea. Această metodă, prin natura ei, este limitată doar la amprentarea echipamentelor wireless.

2.5. Primitive criptografice

Deoarece contribuțiile tezei se bazează pe criptografie în continuare voi face o scurtă trecere în revistă a principalelor primitive criptografice folosite în lucrare.

Primitivele criptografice reprezintă baza sistemelor criptografice. Există o mulţime de astfel de primitive criptografice, acestea pot fi clasificate după cum urmează:

- *Primitive simetrice:*
  - Criptări simetrice;
  - Funcţii hash;
  - Coduri de autentificare a mesajelor;

- *Primitive asimetrice:*
  - Criptări asimetrice;
  - Semnături digitale.

Astfel, primitivele simetrice utilizează aceeaşi cheie atât pentru criptare cât şi pentru decriptare. Acest tip de primitive prezintă avantajul că necesită o putere redusă de calcul, dar prezintă dezavantajul partajării cheii între cei doi participanţi la comunicare. Primitivele asimetrice, spre deosebire de cele simetrice, utilizează chei diferite pentru criptare şi decriptare, această constituie un avantaj, deoarece nu mai trebuie partajată cheie secretă între participanţii la comunicare. Dezavantajul acestor primitive îl reprezintă nevoia de putere de calcul mare. În general securitatea se asigură prin utilizarea combinată a primitivelor simetrice cu cele asimetrice. Pe lângă primitivele simetrice şi asimetrice mai există şi *primitive fără cheie*.

2.5.1. Funcții hash

O **funcţie hash** este o funcţie deterministă ce primeşte la intrare un bloc de date de dimensiune variabilă şi returnează un mesaj de lungime fixă (valoare hash), pe baza căruia nu se poate reconstitui mesajul iniţial.

Proprietăţile ideale ale funcţiilor hash:

- Valoarea hash este uşor de calculat pentru orice mesaj;
- Este nefezabilă găsirea unui mesaj pentru o valoare hash dată;
- Este nefezabilă modificarea mesajului fără modificarea valorii hash;
- Este nefezabila găsirea a două mesaje diferite cu aceeaşi valoare hash.

Notând cu *x* mesajul iniţial şi cu *H(x)* valoarea hash a mesajului, pentru o funcţie hash se impun următoarele obiective de securitate:

1. **Rezistenţa primară a imaginii:** având *H(x)* nu putem găsi *x.*
2. **Rezistenţa secundară a imaginii:** având *x1, H(x1)* nu putem găsi *x2≠x1* astfel încât *H(x1)=H(x2).*
3. **Rezistenţa la coliziune:** nu se poate găsi o pereche *x1,x2 (x2 ≠ x1)* astfel încât *H(x1)=H(x2).*

Funcţiile criptografice au numeroase aplicaţii în cadrul securităţii informaţiei, cum ar fi *semnăturile digitale*, *coduri de autentificare a mesajelor (MAC)* şi alte forme de *autentificare*. Funcţiile hash mai sunt folosite şi la indexarea datelor utilizând *tabele de dispersie*, ca şi *amprente* pentru a identifica în mod unic fişierele, sau ca şi *sumă de corecţie* pentru a detecta coruperea accidentală a fişierelor. Cele mai utilizate funcţii hash sunt în prezent MD5 şi SHA-1, din păcate ambele nu prezintă rezistenţă secundară a imaginii. Pentru o mai bună securitate se recomandă utilizarea SHA-256 sau mai puternic.

2.5.2. Coduri de autentificare a mesajelor

Un algoritm MAC, numit şi *funcţie hash cu cheie*, primeşte ca şi intrare o cheie şi un mesaj de lungime variabilă ce trebuie autentificat, si are ca şi ieşire codul MAC. Codul MAC asigură atât integritate cat şi autenticitatea datelor (mai fiind cunoscute şi sub denumirea de cod de integritate a mesajului – MIC). Deşi codurile MAC sunt similare cu funcţiile hash, totuşi acestea au cerinţe de securitate diferite, cum ar fi rezistenţa la atacuri pe baza textului sursă. Adică dacă atacatorul dispune de o maşină ce deţine cheia secretă şi generează coduri MAC pentru mesajele alese de atacator, el nu poate ghici niciodată codul MAC pentru alte mesaje decât pentru cele care a folosit deja maşina. Funcţiile hash pot fi utilizate la generarea codurilor MAC, având de a face astfel cu coduri HMAC sau utilizând cifru bloc (OMAC, CBC-MAC, PMAC). Astfel, emiţătorul rulează un algoritm MAC pentru a obţine codul MAC. Mesajul va fi trimis împreuna cu codul MAC. La primirea mesajului receptorul rulează acelaşi algoritm pentru a calcula propriul cod MAC pe baza cheii care este partajată între cei doi participanţi la comunicare. În cazul în care codul MAC calculat este identic cu cel primit atunci mesajul poate fi acceptat ca fiind autentic, altfel mesajul nu este acceptat deoarece a fost modificat sau alterat pe parcursul transmisiei.

2.5.3. Semnături digitale

Semnătura digitală este un tip de primitivă asimetrică. Pentru un mesaj ce a fost trimis pe un canal nesecurizat, o semnătură digitală îi poate da receptorului certitudinea că emiţătorul este ceea ce pretinde că este. Semnăturile digitale pot fi considerate echivalentul semnăturilor tradiţionale, o semnătură digitală construită corect este mult mai dificilă de falsificat decât una de mână. Semnăturile digitale asigură şi non-repudierea, adică odată ce un mesaj a fost semna de către autor, acesta nu mai poate pretinde ulterior că nu a semnat mesajul.

Un mecanism de semnare digitală este alcătuit din următorii algoritmi:

- Un algoritm de generare a cheilor, ce alege în mod aleatoriu o cheie privată dintr-o mulţime de chei private posibile. Algoritmul va avea ca şi ieşiri cheia privată şi cheia publică asociată;
- Un algoritm de semnare, care pe baza mesajului şi a cheii private va genera semnătura;
- Un algoritm de verificare, care pe baza mesajului, a cheii publice şi a semnăturii va accepta mesajul ca fiind autentic sau îl va respinge.

În [42] găsim următoarele proprietăţi ale semnăturilor digitale:

i. Trebuie să fie **uşor de calculat** de cel ce semnează mesajul;
ii. Trebuie să fie **uşor de verificat** de oricine;
iii. Trebuie să deţină o **durată de viaţă corespunzătoare.**

Conform [67] mecanismele de semnare digitală se pot împărţi în două categorii:

- **Mecanisme cu anexă,** acestea au nevoie ca mesajul original să fie trimis odată cu semnătura, pentru a fi utilizat le verificare.
- **Mecanisme cu recuperarea mesajului**, nu au nevoie ca mesajul original să fie transmis odată cu semnătura, acesta putând fi refăcut din semnătură.

3. Studii de caz

În cele ce urmează sunt prezentate două studii de caz ce evidențiază, pe de o parte, pericolul reprezentat de atacurile DoS asupra echipamentelor wireless de control industrial de tip SCALANCE, prezentându-se atacurile posibile precum și consecințele acestora. Pe de altă parte, este realizat un studiu al securității principalelor soluții de e-banking disponibile pe piața românească și o analiză a conștientizării utilizatorilor asupra pericolului reprezentat de atacurile DoS combinate cu mesajele de tip spam și cu atacuri de tip inginerie socială asupra serviciilor de e-banking.

3.1. Analiza rezistenței echipamentelor wireless industriale SCALANCE

Modulele SCALANCE sunt folosite pentru a oferi suport wireless în aplicaţiile industriale. Punctele de acces (AP) de tip W788-2RR şi clienţii de tip W747-1RR fiind proiectaţi şi documentaţi foarte bine. În manualul care vine cu aceste echipamente [93] apar şi atacurile ce pot afecta funcţionarea acestor echipamente. Pe lângă aceste atacuri, care sunt generice, în [22] s-au prezentat alte atacuri ce pot afecta funcţionarea modulelor SCALANCE.

3.1.1. Atacuri asupra interfeței de configurare

Pentru configurarea setărilor necesare conectării wireless (SSID, autentificare, etc.) modulele SCALANCE dispun de o interfață web, aceasta fiind singura modalitate de configurare disponibilă. Deși foarte bine gândită, totuși interfața are câteva vulnerabilități (completarea automată a formularelor este activă, valoarea aleatoare pe baza căreia este generat id-ul de sesiune nu este verificată) datorate proiectării sau implementării. Aceste vulnerabilități în combinație cu altele, ce se datorează protocolului SSL/TLS (posibilitatea realizării unui atac de tip MitM, rezistență redusă la epuizarea resurselor, tratarea incorectă a padding-ului folosit în timpul handshake-ului de SSL), fac și mai nesigur procesul de autentificare.

3.1.1.1. Completarea automată a formularelor este activă

Deşi nu reprezintă un atac în sine, completarea automată a formularelor trebuie dezactivată în punctele în care se autorizează accesul la aplicaţii ce procesează informaţii sensibile. Deoarece, de exemplu, dacă PC-ul de unde a fost accesată pagina de configurare a fost compromis, atacatorul poate obţine acces la interfaţa de configurare datorită funcţiei de completare automată a câmpurilor de logare.

3.1.1.2. Atac asupra interfeţei de administrare

Analizând codul JavaScript, cod ce nu este obfuscat, a interfeţei de administrare se poate observa modul în care se realizează autentificarea utilizatorilor:

1.) $C \rightarrow AP : request,$

2.) $AP \rightarrow C : nonce_{AP},$

3.) $C \rightarrow AP : C, MD_5(C, pwd_C, nonce_{AP}), nonce_{AP}.$

Dacă protocolul prezentat mai sus se desfășoară pe un canal nesigur, acesta e vulnerabil la un atac de tip dicţionar, deoarece nu apare nici un *nonce* (valoare aleatoare variabilă în timp) din partea clientului. O altă problemă şi mai serioasă este faptul că serverul web ce rulează pe modul nu verifică dacă hash-ul calculat în pasul 3 a fost sau nu calculat folosind ultimul nonce trimis, ci acceptă toate hash-urile pentru sesiunile neînchise. Astfel, un hash capturat într-o sesiune HTTP poate fi folosit ulterior pe HTTP sau HTTPS.

3.1.1.3. Atacuri asupra SSL/TLS

Deşi protocolul TLS/SSL este sigur, totuşi anumite atacuri se pot realiza datorită hardware-ului sau implementării acestuia.

Man-in-the-middle (MITM). Deoarece certificatele digitale disponibile pe echipamentele SCALANCE sunt semnate de SIEMENS AG, autoritate nerecunoscută de browser, acestea vor apărea ca fiind de neîncredere, permiţând astfel realizarea unui atac de tip MITM. Astfel, adversarul poate crea un certificat fals ce va fii identificat ca fiind de neîncredere de către browser, dar deoarece la fel se întâmplă şi cu certificatul original, e posibil ca utilizatorul să ignore acest avertisment. Atacatorul ar putea astfel să impersoneze modulul SCALANCE şi să obţină parola de administrare a acestuia.

Epuizarea resurselor. Modulele SCALANCE nu oferă suport pentru rezistenţă la atacuri prin epuizarea resurselor. Astfel, supus unui atac de DoS serverul HTTPS de pe modul nu va mai răspunde cererilor iniţializate de utilizator, dar va putea răspunde pe HTTP, expunând astfel valoarea hash-ului utilizat la autentificare, în cazul conectării voluntare a unui utilizator.

MAC incorect. În cazul în care padding-ul de criptare, trimis în timpul handshake-ului SSL/TLS, nu este corect, protocolul va răspunde cu "Bad record MAC". Partea interesantă este că modulul SCALANCE va bloca adresa de IP de pe care a venit mesajul cu MAC eronat, dar va permite cereri HTTP de pe aceeași adresă, la fel ca și în cazul mai sus.

Atacurile prezentate mai sus pot fi combinate într-un singur atac în 3 pași. În primul pas adversarul va crea un pachet de Handshake SSL/TLS ce va conduce la MAC incorect și are ca adresă sursă adresa administratorului. Aceasta va duce la blocarea adresei de IP a administratorului pentru accesul HTTPS. Administratorul va putea să se conecteze utilizând HTTP, dacă această metodă de conectare la interfața de administrare este utilizată, atacatorul va putea captura un hash MD5 valid și îl va putea utiliza ca să se conecteze prin HTTP sau HTTPS ulterior, atâta timp cât sesiunea administratorului este deschisă. Atacul este formalizat mai jos:

a) Într-o primă fază atacatorul compromite conexiunea HTTPS prin inserarea unui padding invalid ($\overline{ClientKeyExchange}$):

1.) $C \overset{SSL/TLS}{\rightarrow} adv(AP) : ClientHello$,

2.) $adv(C) \overset{SSL/TLS}{\rightarrow} AP : ClientHello$,

3.) $AP \overset{SSL/TLS}{\rightarrow} C : ServerHello$,

4.) $adv(C) \overset{SSL/TLS}{\rightarrow} AP : \overline{ClientKeyExchange}$.

b) În a doua fază administratorul vede că nu poate inițializa o conexiune HTTPS și încearcă o conexiune HTTP, furnizând atacatorului id-ul său de sesiune (valoarea funcției hash MD5):

1.) $C \overset{HTTP}{\rightarrow} AP : request$,

2.) $AP \overset{HTTP}{\rightarrow} C : nonce_{AP}$,

3.) $C \overset{HTTP}{\rightarrow} adv(AP) : C, \underline{MD5(C, pwd_c, nonce_{AP})}, nonce_{AP}$.

c) În a treia fază atacatorul va întrerupe conexiunea cu AP astfel încât sesiunea administratorului va rămâne deschisă. Acum atacatorul poate folosi id-ul de sesiune capturat, deoarece AP-ul nu verifică nonce-ul pe baza căruia a fost calculat, pentru a se autentifica la interfața de administrare:

1.) $adv(C) \overset{HTTP}{\rightarrow} AP : request$,

2.) $AP \overset{HTTP}{\rightarrow} adv(C) : nonce'_{AP}$,

3.) $adv(C) \overset{HTTP}{\rightarrow} AP : C, \underline{MD5(C, pwd_C, nonce_{AP})}, nonce'_{AP}$.

În urma acestui atac în trei pași adversarul are acces la interfața de administrare a AP-ului putând modifica parametrii rețelei.

3.1.2. Atacuri asupra comunicației wireless

Atacurile din această secțiune au fost realizate cu ajutorul suitei de audit wireless Aircrack-ng (www.aircrack-ng.org) și sunt atacuri bine cunoscute de spargere a parolelor utilizate la criptarea traficului wireless.

3.1.2.1. Spargerea parolelor WEP

WEP poate fi spart destul de ușor datorită faptului că e bazat pe RC4. Este cunoscută vulnerabilitatea RC4-ului în cazul utilizării de parole identice, de aceea se utilizează un vector de inițializare IV pe 24 biți. Cei 24 de biți fac ca parola efectivă folosită la criptare să se repete după aproximativ 5000 pachete, ceea ce e puțin într-o rețea cu trafic normal. Astfel, pentru a sparge parola WEP folosim aircrack mai întâi pentru a captura pachete:

```
sudo airodump-ng --channel 11 -write cap mon0
```

Dacă sunt capturate suficiente pachete atunci pentru a afla parola se va executa comanda:

```
sudo aircrack-ng cap-01.cap
```

3.1.2.2. Spargerea parolelor WPA/WPA2

WPA/WPA2 folosesc PMK (Pairwise Master Key) pentru a autentifica clienții printr-un handshake în patru pași. Această cheie este obținută prin executarea a 4096 de operații SHA-1 în modul următor: $PMK=SHA1^{4096}(pwd,SSID,SSID_{Len})$. Unde: *pwd* – parola, $SSID_{Len}$ – lungimea SSID-ului. Deci puterea cheii stă, de fapt, în puterea parolei alese de utilizator.

Și în acest caz se poate folosi aircrack pentru spargerea parolei:

```
sudo cowpatty -r cap-01.cap -f dict -s SCALANCE.
```

În comanda de mai sus *-r* reprezintă o secvență handshake capturată printr-un atac de de-autentificare, *-f* reprezintă tabela rainbow (o tabelă de valori hash pre-calculate), iar *–s* SSID-ul rețelei.

3.1.2.3. Atac prin de-autentificare

Pentru a realiza acest tip de atac e nevoie de o placă de rețea wireless ce suportă injectarea de pachete de management. Pentru aceasta s-a folosit o placă de rețea Atheros AR5BMB5 la care s-a modificat driver-ul original pentru a permite injecția de pachete de management (Secvența 1).

```
wget http://wireless.kernel.org/download/compat-wireless-
2.6/compat-wireless-2010-10-16.tar.bz2
tar -jxf compat-wireless-2010-10-16.tar.bz2
cd compat-wireless-2010-10-16
wget http://patches.aircrack-
ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch
patch -p1 < mac80211.compat08082009.wl_frag+ack_v1.patch
wget http://patches.aircrack-ng.org/channel-negative-one-
maxim.patch
patch ./net/wireless/chan.c channel-negative-one-
maxim.patch
gedit scripts/update-initramfs
#*** GĂSEȘTE LINIA 13: KLIB=/lib/modules/2.6.31-wl/build
#*** ÎNLOCUIEȘTE CU: KLIB=/lib/modules/$(uname -r)/build
make
sudo make install
sudo make unload
sudo reboot
```

Secvența 1 Modificarea driver-ului plăcii de rețea Atheros pentru a permite injecția de pachete de management

După ce s-a modificat driver-ul trebuie activat modul de monitorizare:

```
sudo airmon-ng start wlan0
```

Această comandă va crea o interfața virtuală, numită *mon0*. Utilizând această interfață se va monitoriza mediul wireless între AP și victimă pe canalul de comunicare folosit de aceștia (11 în cazul nostru):

```
sudo airodump-ng --channel 11 mon0
```

Monitorizând mediul se va obține BSSID-ul AP-ului, cu această informație disponibilă se va porni atacul:

```
sudo aireplay-ng -0 0 -a 00:0E:8C:BF:25:78 -c 00:0E:8C:BC:2D:60 mon0.
```

În comanda prezentată mai sus *-0* reprezintă tipul atacului (de de-autentificare în cazul nostru), *0* numărul de repetări (0 = infinit), *-a* reprezintă BSSID-ul AP-ului, *-c* este MAC-ul victimei iar *mon0* este interfața virtuală folosită la monitorizare.

3.1.3. Utilizarea atacului prin de-autentificare asupra unei aplicații de control la distanță

În continuare se vor prezenta implicațiile atacului prin de-autentificare asupra unui proces simulat utilizând LabView.

3.1.3.1. Descrierea aplicației de control

Controler-ul țintă este Simatic S7-315F PLC [92] pe care rulează algoritmul de reglare. Pentru programarea controler-ului s-a utilizat Step7. Comunicarea este realizată folosind *Ole for Process Controll (OPC Server)* și are ca suport infrastructura wireless oferită de Siemens SCALANCE. Structura sistemului de reglare este prezentată în Figura 4:

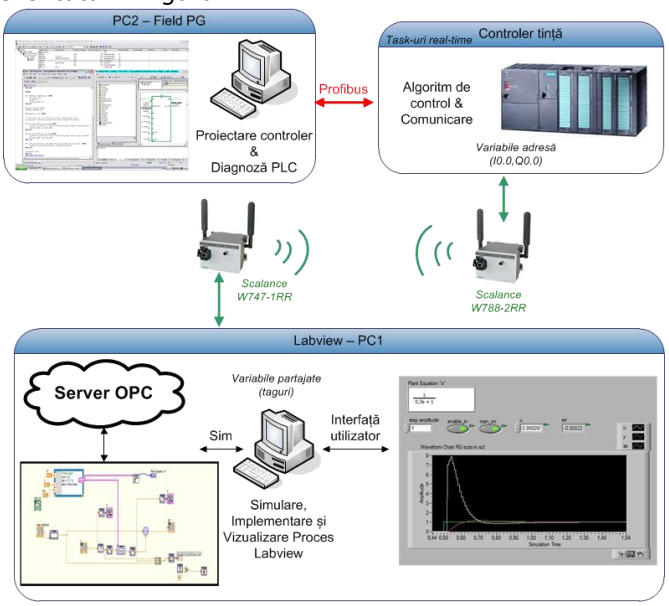


Figura 4 Structura generală a sistemului de reglare [22]

Procesul folosit, pentru analiza efectelor atacurilor, este unul de ordinul 1 cu o constantă de timp 0.5s. Acest proces este folosit frecvent pentru modelarea pompelor electrice. Structura de reglare este prezentată în Figura 5, funcția de transfer a procesului condus fiind dată de relația:

$$H(s) = \frac{1}{0.5s + 1}$$

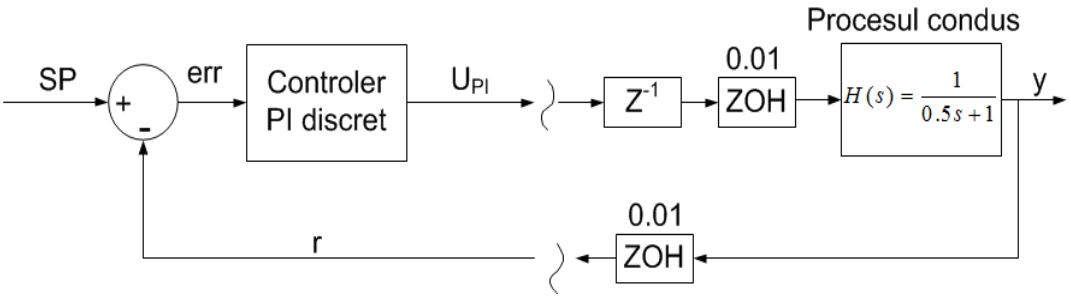


Figura 5 Controlerul și procesul condus [22]

Structura de reglare este bazată pe un controler PI discret, ce folosește o regulă de integrare trapezoidală, dată de relația:

$$U_{PI}(k) = K_{p^e}(k) + \left\{ u_i(k-1) + \frac{K_p}{T_i}\left[\frac{e(k) + e(k+1)}{2}\right]h \right\}.$$

Nivelul de sincronizare atins de comandă (Figura 6) și de răspuns este acceptabil dacă diferențele, dintre metoda de control implementată în PLC și cea simulată în Labview, sunt minore.

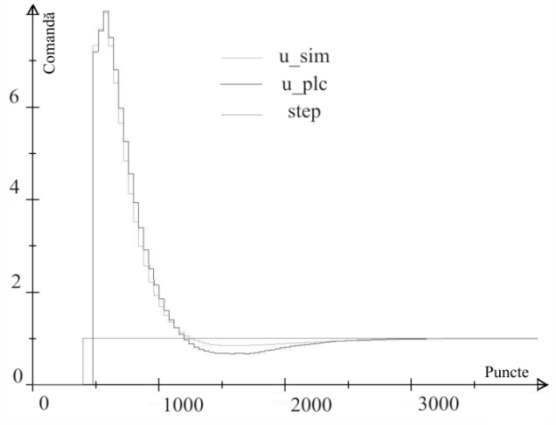


Figura 6 Nivelul de sincronizare pentru comandă [22]

3.1.3.2. Scenariul de atac

Scenariul de atac este realizat asupra sistemului prezentat în Figura 4 prin intermediul pachetelor de de-autentificare trimise clientului wireless. Parametrii de acordare ai controlerului au fost fixați la: K_p=7 și T_i=2s, comanda fiind limitată în intervalul [-50,+50]. Lungimea semnalului treaptă (SP) de discretizare al controlerului este h=10ms. La momentul t=0,5s un semnal treaptă de amplitudine A=4 (valoarea de referință) este aplicat procesului, după care eroarea este calculată si furnizată controlerului din PLC. Comanda furnizată de PLC inițializează și stabilizează sistemul în jurul valorii de referință. Astfel, în cazul *scenariului simulat,* un atac prin de-autentificare conduce la întreruperea completă a comunicării dintre controler și proces. Serverul OPC și simulatorul schimbă date prin intermediul unui mecanism bazat pe variabile partajate. Dacă se întrerupe comunicația cu PLC-ul, serverul OPC consideră aceste variabile ca fiind de *proastă calitate*. Ca și consecință simulatorul va furniza procesului ultima comandă validă primită de la controler.

În Figura 7 și Figura 8 sunt prezentate comparativ comanda dată de controler și răspunsul procesului, într-o situație normală de control automat, respectiv în scenariul de atac. După cum se observă, în scenariul de atac, comanda este menținută la ultima valoare primită iar răspunsul procesului continuă să crească punând în primejdie aplicația.

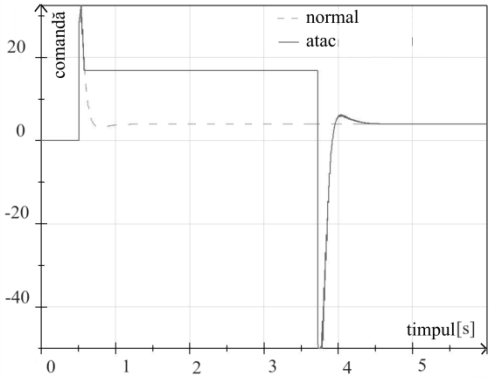


Figura 7 Comanda oferită de controler [22]

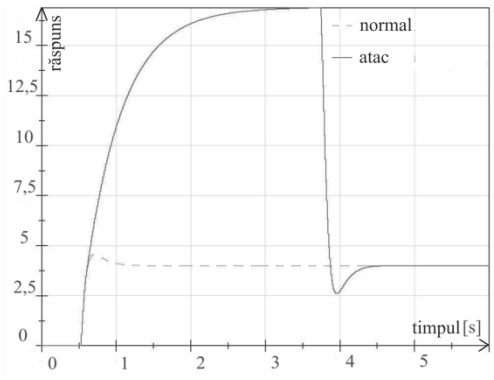


Figura 8 Răspunsul procesului [22]

Într-un *scenariu real*, comanda trimisă procesului condus este nulă, întreg sistemul comportându-se impredictibil. În plus, dacă atacatorul cunoaște modul în care procesul răspunde, fapt destul de frecvent în realitate, acesta poate conduce procesul într-o anumită stare, prin activarea/dezactivarea atacului. Acest comportament poate fi comparat cu acțiunea unui controler proporțional sau bipozițional. Principala problemă legată de întreruperea comunicației cu un controler (de obicei PI), într-un scenariu real de control automat, este că acesta va detecta o eroare constantă și indiferent de dimensiunea comenzii, această eroare nu poate fi compensată. Astfel, chiar dacă acest controler va funcționa perfect, în lipsa unui

mecanism anti wind-up, semnalul de comandă va continua să crească (în valoare absolută), punând în pericol sistemul.

3.2. O analiză a eficienței soluțiilor de tip e-banking, disponibile pe piața din România, în fața atacurilor mixte DoS și inginerie socială

Soluțiile de e-banking au cunoscut o cunoscut evoluție în ultimii ani deoarece clienții economisesc bani și timp prin faptul că nu mai trebuie să se deplaseze la sediile băncilor pentru a realiza tranzacții. Un studiu efectuat de American Bankers Association - ABA (www.aba.com) în septembrie 2010 arată că 36% dintre consumatori preferă serviciile bancare on-line în detrimentul celor tradiționale. Odată cu dezvoltarea serviciilor bancare prin Internet a început să crească și numărul fraudelor în acest sector, punând în pericol finanțele consumatorilor acestor servicii. Pentru a evita pierderea clienților, furnizorii de soluții de e-banking au început să introducă diverse metode de autentificare a utilizatorilor, făcând din autentificare principalul obiectiv de securitate al aplicațiilor de e-banking.

3.2.1. Autentificarea în e-banking

Pentru a asigura autentificarea, soluțiile sunt divizate în două categorii [49]: soluții care folosesc parole one-time (parole de unică folosință și cu timp de viață redus) și soluții bazate pe certificate digitale. Parolele one-time sunt obținute fie folosind liste de parole sau token-uri. Soluțiile bazate pe liste de parole trebuie evitate deoarece utilizatorii obișnuiesc să salveze aceste liste pe PC, iar odată compromis PC-ul sunt compromise și listele. Pe de altă parte token-urile generează parole în funcție de timp sau folosind un mecanism challenge-response, parole ce nu mai trebuiesc notate de utilizator. Chiar dacă o parolă este compromisă, ea are un timp de viață limitat.

Totuși token-urile au o problemă, se bazează pe securitate prin obscuritate. Deși această metodă este mult mai sigură decât precedenta, totuși nu poate face față unui atac Man-In-The-Middle (MITM), când un atacator se plasează între utilizator și bancă. Acest atac poate fi prevenit, însă, dacă se utilizează SSL/TLS, dar adversarul poate apela în acest caz la atacuri de tip inginerie socială (phishing, pharming). Soluțiile bazate pe certificate digitale adaugă un al doilea factor de autentificare (pe lângă parole, PIN-uri, întrebări de securitate, etc.). Dar trebuie evitată situația în care soluția devine prea greu de utilizat, deoarece această soluție nu va reuși să-și atingă scopul. De aceea realizarea unui compromis între securitate și uzabilitate reprezintă o sarcină foarte dificilă ([47], [72], [102]). Pe de altă parte, și soluții bine cunoscute (cum ar fi SSL/TLS) nu sunt folosite corect de clienți [48], iar unii utilizatori, în special cei novici, percep greșit securitatea [39]. De aceea, de cele mai multe ori securitatea nu ar trebui să se bazeze pe opțiunile utilizatorului.

O ierarhie excelentă a tehnicilor de autentificare este prezentată în [49], soluțiile fiind divizate în următoarele categorii:

- Soluții ce nu sunt rezistente la atacuri prin utilizarea de software malițios de către atacator: 0 – nici un fel de securitate; 1 – parole statice; 2 – certificate digitale.
- Soluții ce nu sunt rezistente la atacuri de tip inginerie socială: 3 – parole de tip one-time.
- Soluții ce sunt rezistente la atacuri de tip inginerie socială: 4 – parole bazate pe timp; 5 – parole bazate pe un challenge din partea serverului; 6 – autentificare SSL/TLS (PKI hardware); 7 – semnarea tranzacțiilor pe platforme de încredere.

Tot conform [49] soluțiile ce au nivel de securitate mai mic decât 5 sunt considerate ca fiind nesigure în fața unor atacuri off-line orientate spre furtul informațiilor de logare, doar o singură bancă din cele analizate are un proces de autentificare cu un nivel de securitate egal cu 5 iar alte trei bănci au un nivel de securitate egal cu 5 pentru autorizarea tranzacțiilor. De asemenea, aplicațiile ce au un nivel de securitate mai mic de 6 sunt considerate nesigure în fața unor atacuri on-line elaborate. După analiza aplicațiilor de e-banking disponibile pe piața românească s-a constatat că nicio aplicație nu are un nivel de securitate mai mare decât 5.

3.2.2. Atacuri de tip inginerie socială

În această categorie întâlnim atacurile de tip *phishing* si *pharming*, acest tip de atacuri sunt orientate spre obţinerea de informaţii personale (cum ar fi nume de utilizator, parole, detalii despre cărţile de credit, etc.) prin impersonarea unei entităţi în care utilizatorul are încredere. Aceste atacuri se realizează, de obicei, prin intermediul serviciilor de email, mesagerie electronică sau prin reţelele de socializare. Astfel, în cazul atacurilor de tip phishing, atacatorul trimite un mesaj care conţine un link către o pagină web ce imită pagina originală în care utilizatorul are încredere.

Deoarece un utilizator „educat” poate identifica acest tip de atac prin simpla analiză a URL-ului conţinut în link, o altă formă de atac a apărut: pharming-ul. În acest caz atacatorul redirecţionează traficul destinat paginii originale către o pagină falsă prin compromiterea sistemului utilizatorului sau a serviciului de DNS utilizat de victimă, aceasta conectându-se la aplicația de e-banking folosind URL-ul original.

Pentru a merge și mai departe, atacurile cele mai recente vizează browser-ul utilizatorului. Aceste atacuri sunt cunoscute ca și atacuri de tip Man-in-the-Browser (MitB) [32], iar în fața lor nici măcar mecanismele cu cheie publică (cum ar fi SSL/TLS) nu sunt eficiente. Atacurile MitB sunt realizate prin intermediul virușilor de tip *cal troian* prezenți în extensiile browser-ele web. O analiză a troienilor disponibili pentru Internet Explorer și Firefox este prezentată în [98].

Aceste atacuri sunt direcţionate mai ales spre săvârşirea fraudei electronice. Pentru a contracara acest tip de atacuri, societăţile care oferă soluţii de tip e-banking au început să ofere token-uri pentru autentificarea clienţilor. Dar aceste token-uri nu sunt suficiente în faţa unor atacuri mai complexe ce îmbină ingineria socială cu atacurile de tip DoS sau DDoS. Atacurile DoS vizând de multe ori serverele băncilor, un raport [2] recent prezintă situația unei bănci din Olanda care a căzut victima unui atac DDoS. Din această cauză serverele băncilor trebuie să fie rezistente la astfel de atacuri.

3.2.3. Prezentare generală a soluțiilor de securitate a aplicațiilor de e-banking din România

Conform BitDefender [4], în România 27% dintre utilizatorii de internet nu au auzit de phishing. Probabil, din această cauză România a fost a doua țară emitentă de carduri de credit ce au fost folosite pentru a comite fraudă electronică [75] (17% din totalul cardurilor de credit, folosite pentru a comite fraudă, din întreaga lume). De asemenea, România este țara sursă numărul 1 în ceea ce privește frauda electronică, 32% din frauda electronică își are originea în România [4]. Pentru analiza realizată în [19] s-a făcut următoarea presupunere: atacatorul se poate amplasa între client și serverul băncii și are posibilitatea de a modifica orice parte a paginii web pe care o dorește, lăsând restul intact (Figura 9).

Figura 9 Atacatorul intervenind asupra canalului de comunicare [19]

Procesul de autentificare împreună cu autorizarea tranzacțiilor este prezentat în Tabelul 1. Ca și notații generale vom folosi: *U* pentru utilizator, *B* pentru bancă, *adv* pentru atacator și $sign_{token}$ pentru semnătura realizată cu token-ul, chiar dacă nu este o semnătură digitală în sine, o vom numi astfel, deoarece așa apare în manualele aplicațiilor de tip e–banking.

Celelalte notații ce apar în Tabelul 1 pot fi clasificate ca:

- Parole și nume de utilizator: $user_B/user_U$ și $password_B/password_U$ care pot fie să fie definite de utilizator fie sunt impuse de bancă, sau $password_{token}$ care reprezintă un cod din 6 cifre generat de digipass-ul Vasco (http://www.vasco.com) ce este protejat de un cod PIN;
- Informații personale: $iban^4{}_R$ ce reprezintă ultimele 4 cifre din IBAN-ul destinatarului sumei de bani, $amnt_R$ este suma de bani ce urmează a fi

transferată, cnp^2_U sunt două cifre din CNP-ul utilizatorului, $psw1^3_U/psw2^3_U$ sunt 3 caractere aleatoare din prima/a doua parolă definită de utilizator;

- Date de proces: $authorize_U$ reprezintă o cerere de autorizare inițializată de utilizator, $nonce_U$ este un nonce creat de bancă pentru cererea de autorizare făcută de utilizator, $sign_{token}(x)$ este semnătura realizată pe intrarea *x* folosind token-ul, *cnt* reprezintă un contor folosit de digipass pentru generarea parolelor unice, $authcode_B^{SMS}$ reprezintă codul de autorizare a unei tranzacții trimis de bancă prin SMS către utilizator, $authcode_B^{PC}$ reprezintă codul de autorizare trimis prin PC, *cert* este certificatul utilizatorului semnat de bancă.

3.2.3.1. Mecanisme și vulnerabilități

Analiza securității soluțiilor de e-banking a fost făcută în raport cu următoarele atacuri: furtul datelor de autentificare, spargerea canalului de comunicare și manipularea conținutului. Aceste atacuri pot fi realizate, după cum urmează: primul, prin intermediul unui malware instalat pe sistemul utilizatorului; al doilea, prin intermediul unui site de phishing, în care atacatorul are acces la canalul de comunicare; iar al treilea, prin intermediul unui atac de tip MitB. În continuare se vor prezenta rezultatele analizei securității aplicațiilor de e-banking a principalelor bănci de pe piața românească.

1. *BCR Erste Bank:* chiar dacă dispun de un mecanism de autentificare destul de sigur, un atacator poate monitoriza conexiunea dintre utilizator și bancă în timpul unei cereri de autentificare inițializată de utilizator și prin intermediul unei pagini de phishing, care imită pagina de logare a BCR, atacatorul poate captura parola generată de token pentru a se autentifica la site, dar această parolă poate fi folosită doar într-o fereastră îngustă de timp (~30 sec). Pentru a autoriza un transfer de bani în interesul său, atacatorul trebuie să creeze un cont care va avea ultimele 4 cifre ale IBAN-ului identice cu cele ale destinatarului unei tranzacții inițializate de utilizator sau poate folosi un atac de tip MitB pentru a manipula IBAN-ul în interesul atacatorului. Totuși atacatorul nu va putea transfera decât suma pe care utilizatorul dorește să o transfere, deoarece aceasta e semnată de utilizator.
2. *BRD Bank – Groupe Societe Generale:* sistemul de banking online al celor de la BRD este cel mai simplu și cel mai nesigur deoarece se bazează doar pe un nume de utilizator ales de acesta și o parolă impusă de bancă ce va fi introdusă de pe o tastatură virtuală. Atât parola cât și numele de utilizator pot fi ușor capturate de o pagină de phishing.
3. *Transilvania Bank:* dispune de trei metode de autentificare. În cazul primei metode de autentificare utilizatorul are nevoie de un certificat digital semnat de bancă împreună cu un nume de utilizator definit de bancă și o parolă aleasă de el. Utilizatorul își va introduce datele de autentificare prin intermediul unei tastaturi virtuale, tranzacțiile putând fii autorizate imediat după ce se face autentificarea. Siguranța acestei aplicații de tip e-banking se bazează pe siguranța păstrării certificatului digital. Deoarece cererea pentru certificatul digital se realizează on-line,

un atacator poate crea un site de phishing pentru a captura această cerere, de-asemenea atacatorul trebuie să obțină numele de utilizator și parola printr-un procedeu asemănător.

O altă problemă cu această metodă de autentificare este faptul că certificatul digital trebuie instalat în fiecare browser din care aplicația este accesată de către client. Iar dacă utilizatorul accesează site-ul de pe un calculator public și acesta uită să-l dezinstaleze, un atacator poate face rost de acest certificat de pe un astfel de calculator public. Aplicația de tip e-banking folosește un alt port decât 80 pentru conectare, lucru deranjant pentru utilizatori care se conectează la internet printr-un firewall.

A doua metodă de autentificare se realizează prin intermediul unui cod trimis de bancă utilizatorului prin SMS, împreună cu un nume de utilizator definit de bancă și o parolă definită de utilizator. Un atacator poate captura aceste informații de autentificare folosind phishing, dar codul trimis prin SMS are o durată de viață limitată (~30 sec). Pentru autorizare utilizatorul, odată autentificat, trebuie doar să confirme transferul folosind parola capturată de el.

A treia metodă de autentificare este asemănătoare cu cea folosită de BCR. Dar, pentru autorizare este nevoie doar de un cod generat de token, cod cu o durată de viață de aproximativ 30 sec. Astfel, capturarea datelor de autentificare de către atacator nu reprezintă un atac foarte eficient. Dar, aplicația nu este rezistentă în fața unui atac de tip MitB, ce poate manipula conținutul conversației dintre utilizator și bancă.

4. CEC Bank: este o bancă cu tradiție în România și are o evoluție foarte bună pe piața bancară în ultimul timp. CEC Bank folosește un mecanism de autentificare/autorizare ce este asemănător cu a treia metodă folosită de Transilvania Bank, având, astfel, aceleași vulnerabilități.

5. Raiffeisen Bank: folosește un mecanism de autentificare asemănător cu cel al lui CEC Bank, dar este folosit un contor în loc de timestamp la generarea codului de autentificare. Astfel, acest cod poate fi folosit de către un atacator în orice moment de timp. Pentru autorizarea tranzacțiilor se folosește același mecanism ca și la BCR, prezentând aceleași vulnerabilități. S-a mai observat că în codul JavaScript al paginii web apare foarte mult cod sursă comentat, cod sursă ce poate fi exploatat de către atacator.

6. *UniCredit Țiriac Bank:* avem de a face aici cu același mecanism de autentificare utilizat de BCR. Cu privire la autorizarea tranzacțiilor se folosește aceeași strategie ca la CEC, deci vulnerabilitățile pot fi deduse de la cele două bănci.

7. *Alpha Bank:* utilizează un mecanism de autentificare similar cu prima metodă de autentificare a Transilvania Bank, având aceleași vulnerabilități. Legat de tranzacții, Alpha Bank nu folosește nici un mecanism pentru autorizarea lor.

Bank	Authentication	Transaction authorization
BCR Erste Bank	1. $U \rightarrow B: user_B, password_{token}(t)$	1. $U \rightarrow B: authorize_U$ 2. $B \rightarrow U: iban_R^4, amnt_R$ 3. $U \rightarrow B: Sign_{token}(iban_R^4, amnt_R, t)$
BRD Bank - GSG	1. $U \rightarrow B: user_B, password_U$	not required
Transilvania Bank	1a. $U \rightarrow B: user_B, password_U, cert_U$ or	1a. $U \rightarrow B: authorize_U, password_U$
	1b. $U \rightarrow B: user_B, password_U$ 2b. $B \rightarrow U: authcode_B^{SMS}$ 3b. $U \rightarrow B: authcode_B^{PC}$ or	1b. $U \rightarrow B: authorize_U, password_U$
	1c. $U \rightarrow B: user_B, password_{token}(t)$	1c. $U \rightarrow B: authorize_U, password_{token}(t)$
CEC Bank	1. $U \rightarrow B: user_B, password_{token}(t)$	1. $U \rightarrow B: authorize_U, password_{token}(t)$
Raffeisen Bank	1. $U \rightarrow B: user_U, password_{token}(cnt)$	1. $U \rightarrow B: authorize_U$ 2. $B \rightarrow U: iban_R^4, amnt_R$ 3. $U \rightarrow B: Sign_{token}(iban_R^4, amnt_R, t)$
UniCredit Tiriac Bank	1. $U \rightarrow B: user_B, password_{token}(t)$	1. $U \rightarrow B: authorize_U, password_{token}(t)$
Alpha Bank	1. $U \rightarrow B: user_B, password_U, cert_U$	not required
ING Bank	1. $U \rightarrow B: user_B, password_{token}(t)$	1. $U \rightarrow B: authorize_U$ 2. $B \rightarrow U: nonce_U$ 3. $U \rightarrow B: Sign_{token}(nonce_U)$
Bancpost Bank	1. $U \rightarrow B: user_B, password_U$	not required
	2a. $B \rightarrow U: nonce_U$ 3a. $U \rightarrow B: Sign_{token}(nonce_U)$ or	
	2b. $U \rightarrow B: password_{token}(t)$ or	
	2c. $B \rightarrow U: authcode_B^{SMS}$ 3c. $U \rightarrow B: authcode_B^{PC}$	
CitiBank	1. $U \rightarrow B: user_U, password_U, question_U$	1'. $U \rightarrow B: add_R$

Bank	Authentication	Transaction authorization
		2'. $B \rightarrow U: authcode_B^{SMS}$ 3'. $U \rightarrow B: authcode_B^{PC}$ 1. $U \rightarrow B: authorize_U, password_U$
Millennium Bank	1. $U \rightarrow B: user_B, password_U, cnp_U^2$	1a. $U \rightarrow B: authorize_U, psw1_U^3 \mid psw2_U^3$ or
		1b. $U \rightarrow B: authorize_U$ 2b. $B \rightarrow U: authcode_B^{SMS}$ 3b. $U \rightarrow B: authcode_B^{PC}$
ProCredit Bank	1. $U \rightarrow B: user_U, password_U, password_{token}(t)$	1. $U \rightarrow B: authorize_U, password_U, password_{token}(t)$

Tabelul 1 Autentificare și autorizare în cazul băncilor românești (băncile sunt sortate după cota de piață) [19]

8. *ING Bank:* folosește același mecanism de autentificare ca și BCR. Legat de autorizarea tranzacțiilor, atacatorul trebuie să aștepte momentul în care utilizatorul dorește să facă un transfer iar în acest moment să modifice valoarea nonce-ului cu cea de care are el nevoie, folosind MitB.

9. *Bancpost Bank:* cei de la Bancpost utilizează o metodă de autentificare în doi pași. În primul pas utilizatorul introduce un nume de utilizator oferit de bancă și o parolă definită de el. În al doilea pas, utilizatorul trebuie să aleagă tipul de autentificare dorit. Utilizatorul poate opta pentru autentificare challenge-response folosind token-ul; autentificare bazată pe un cod generat de token, sau cod de autentificare trimis de bancă prin SMS. În toate cele trei situații, dacă acest cod e capturat de atacator, acesta îl poate folosi pentru a se autentifica doar într-o perioadă redusă de timp. Dar acest cod este folosit o singură dată, astfel dacă un utilizator reușește să se autentifice, poate autoriza tranzacții fără un alt efort.

10. *CitiBank:* are un mecanism de autentificare simplu bazat pe un nume de utilizator și o parolă alese de client și pe răspunsul la o parolă aleasă ($question_U$) aleator dintr-un set standard de 8 întrebări. Parola definită de utilizator este folosită și pentru autorizarea tranzacțiilor, dar tranzacțiile pot fi efectuare doar în conturi predefinite. Pentru a adăuga un cont nou (add_R) utilizatorul trebuie să valideze acțiunea folosind un cod trimis de bancă prin SMS. Folosind MitB adversarul poate schimba contul, dar acest lucru poate fi observat de către un utilizator vigilent.

11. Millennium Bank: folosește pentru autentificarea la aplicația de e-banking o informație personală specifică fiecărui client (cnp-ul), dar această informație poate fi foarte ușor capturată cu un site de phishing. Pentru tranzacții se

oferă două opțiuni una bazată pe două parole (ușor de capturat folosind un site de phishing) și una mai avansată folosind telefonul mobil. Și cea de a doua metodă poate fi atacată folosind inginerie socială în modul următor: atacatorul va modifica informația tranzacției în interes propriu, astfel încât banca va trimite prin telefon codul pentru autorizarea tranzacției atacatorului. Acesta va trebui să captureze acest cod atunci când clientul îl va folosi ca să autorizeze propria tranzacție.

12. ProCredit Bank: folosește pe lângă un nume de utilizator și o parolă aleasă de client și un cod generat de token. Chiar dacă aceste informații sunt capturate de atacator, codul generat de token poate fi folosit doar într-o perioadă scurtă de timp, după care expiră. Pentru a autoriza o tranzacție este nevoie de parola definită de utilizator și de un cod generat de token. Acest mecanism de autentificare/autorizare nu este rezistent în fața unui atac de tip MitB.

3.2.4. O ierarhie a nivelurilor de securitate

În Figura 10 este prezentată o ierarhie a securității (**S**) autentificării și autorizării tranzacțiilor pentru aplicațiile de e-banking analizate, ierarhia este bazată pe nivelele de securitate definite în [49].

După cum se poate observa, nivelurile de securitate corespunzătoare autentificării nu sunt tot timpul identice cu cele corespunzătoare autorizării tranzacțiilor. De exemplu, ING folosește un mecanism de tip challenge-response pentru autorizarea tranzacțiilor. Deși acest mecanism este considerat mai sigur decât un mecanism bazat pe timestamp-uri, ca cel folosit de BCR sau Raiffeisen, nivelul de securitate pentru autorizarea tranzacțiilor al BCR si Raiffeisen a fost ridicat la 5, deoarece în acest caz se semnează suma de bani și IBAN-ul, pe când nivelul de securitate al autentificării pentru BCR și Raiffeisen a rămas la 4.

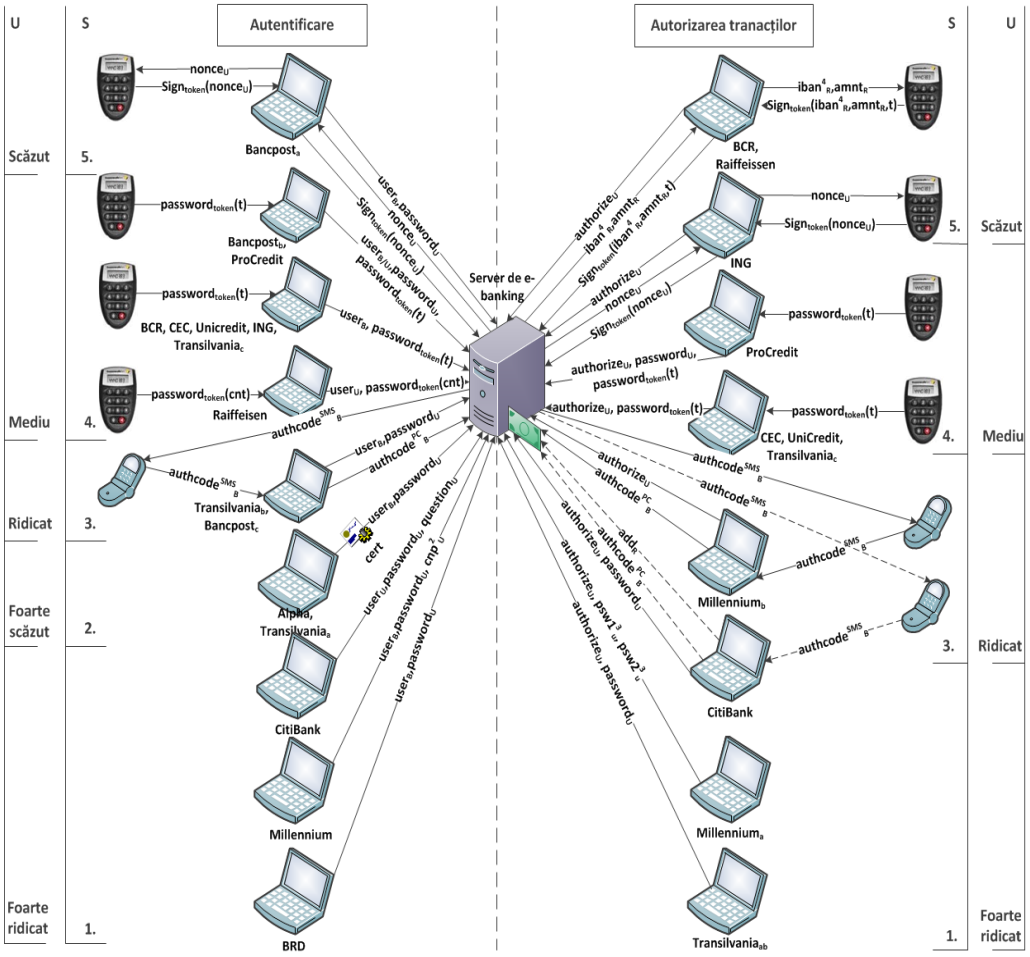


Figura 10 Posibilă ierarhizare a mecanismelor de autentificare/autorizare în aplicații e-banking [19]

Ca și concluzie parțială, în Figura 11 este prezentată o comparație a cotei de piață, conform datelor din 2011 [50], cu securitatea aplicațiilor de e-banking în fața transferului fraudulos de bani. Trebuie menționat că securitatea aplicațiilor de e-banking a evoluat până în 2013 (la fel și atacurile) iar băncile nu au făcut publice valorile pierderilor datorate fraudei electronice, întărind ideea că se bazează pe securitatea prin obscuritate.

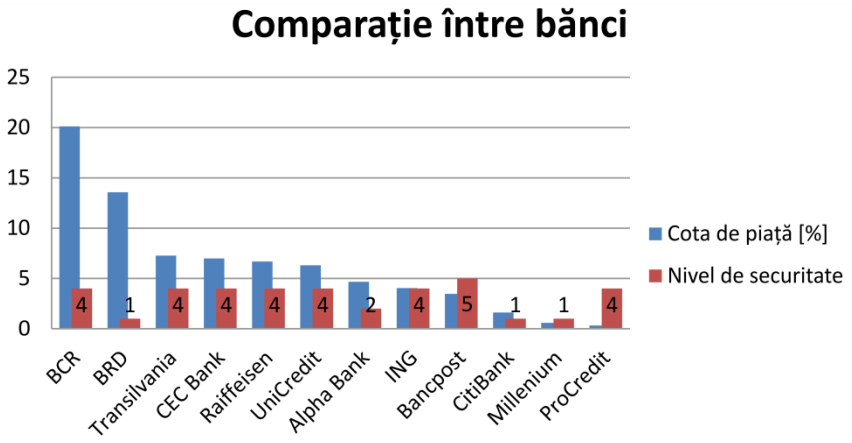


Figura 11 Comparație între cota de piață și nivelul de securitate a băncilor analizate [19]

3.2.5. O ierarhie a uzabilității

Pentru a realiza o ierarhizare a uzabilității soluțiilor de e-banking din România mai întâi, în Tabelul 2 sunt prezentate atributele echipamentelor folosite pentru autentificare/autorizare.

	Raiffeisen Bank	ING	Alpha Bank	Millenium Bank	BRD	BCR	UniCredit Tiriac Bank	CEC Bank	Transilvania Bank	Bancpost Romania	ProCredit Romania	CitiBank Romania
Intrare	Y	Y	*	N	*	Y	Y	Y	Y	Y	Y	*
Ieșire	Y	Y	*	Y	*	Y	Y	Y	Y	Y	Y	*
cost	M	M	S	R	S	M	M	M	S	M	M	S
Confort	S	S	S	M	R	M	M	M	S	M	S	R

Tabelul 2 Atribute ale echipamentelor de autentificare: intrare, ieșire, cost, confort

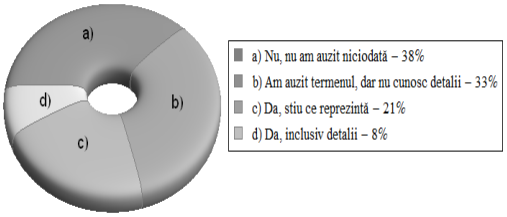
Pentru fiecare bancă se evidențiază dacă dispozitivul de autentificare oferă (Y) sau nu (N) un mecanism de intrare (tastatură) sau de ieșire (display) iar apoi se face o ierarhie (pe trei nivele: scăzut-S, mediu-M, ridicat-R) a costului și comodității utilizării echipamentului de autentificare. Telefoanele mobile sunt mai scumpe decât token-urile, iar parolele sunt cele mai ieftine. De asemenea, parolele sunt cele mai comode, nefiind necesar ca utilizatorul să dețină un alt echipament. Parolele sunt urmate de telefoanele mobile în ierarhia confortului, deoarece utilizatorii au de obicei un telefon la ei. Token-urile sunt și mai incomode deoarece utilizatori au

nevoie de un echipament în plus pentru a se autentifica/autoriza. Certificatele digitale sunt cele mai incomode, deoarece o soluție bazată pe acestea nu este portabilă și presupune un anumit nivel de cunoștințe tehnice pentru a le utiliza.

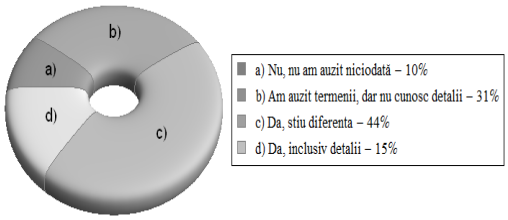
Ierarhia uzabilității este de obicei inversa ierarhii securității, după cum se vede în Figura 10 (**U**).

Ierarhia uzabilității s-a realizat pe baza unui studiu ce a implicat mai mult de 100 persoane, grup format din bărbați 55% și femei 45% cu vârste între 18 și 40 ani. Rezultatele studiului sunt prezentate în Figura 12, Figura 13, Figura 14 și Figura 15.

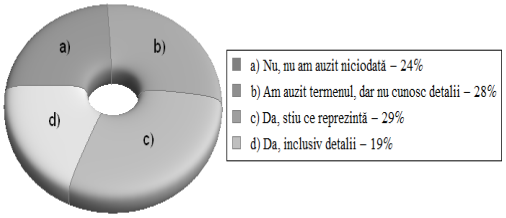
Întrebările au adresat conștiințele de securitate ale utilizatorilor și preferințele lor în materie de uzabilitate. Rezultatele arată că majoritatea utilizatorilor nu deschid mesajele marcate ca spam și jumătate din participanți știu ce reprezintă indicatorii de securitate din browser și nu acceptă certificate ce nu sunt semnate de autorități de încredere. Aceste rezultate demonstrează o anumită conștientizare a importanței securității, dar nu sunt cele mai bune. Totuși, există o mulțime de persoane care nu știu ce este acela un phishing sau un certificat digital. Pe de altă parte, marea majoritate a utilizatorilor sunt preocupați mai mult de securitate decât de uzabilitate, preferând token-urile sau mesajele primite pe telefonul mobil pentru autentificare. Acest rezultat confirmă faptul că ignorarea securității în soluțiile de e-banking este o greșeală critică [33], deoarece utilizatorii sunt foarte preocupați de aceasta. Se poate presupune că parolele pot fi sigure pentru utilizatorii care pot identifica un site autentic și folosesc doar calculatoare ”curate” pentru realizarea conexiunii cu aplicația de e-banking. Majoritatea participanților la studiu folosesc Windows și își actualizează antivirusul.



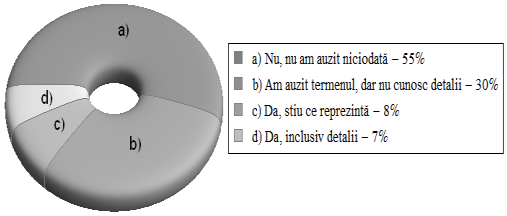
Știți ce reprezintă SSL/TLS?



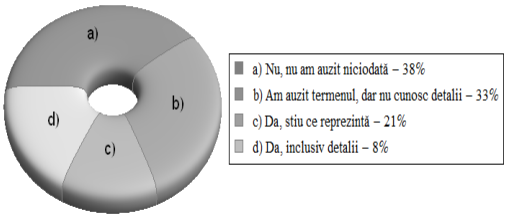
Știți care este diferența dintre HTTP și HTTPS?



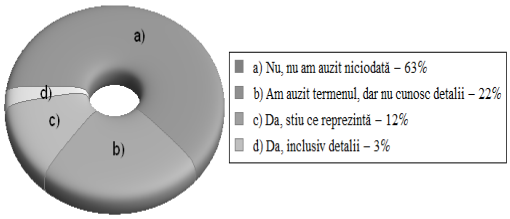
Știți ce este un atac de tip phishing?



Știți ce este un atac de tip pharming?

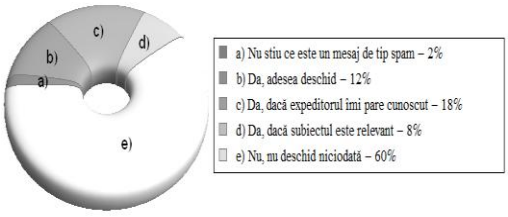


Știți ce este un atac de tip DoS?

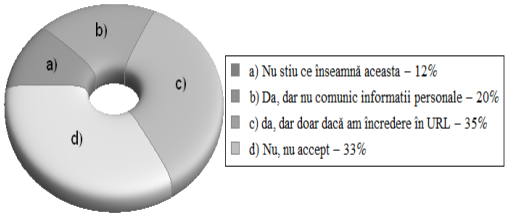


Știți ce este un atac de tip Man-in-the-Browser?

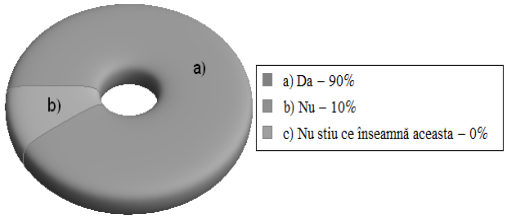
Figura 12 Chestionar securitate/uzabilitate 1



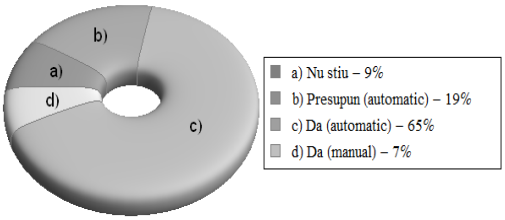
Obișnuiți să deschideți mesaje de tip spam?



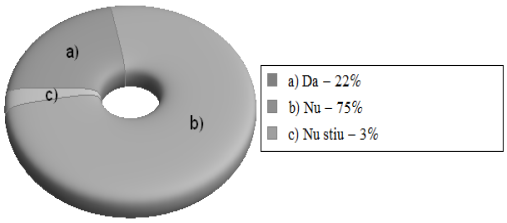
Acceptați certificate ce nu sunt semnate de o autoritate de încredere?



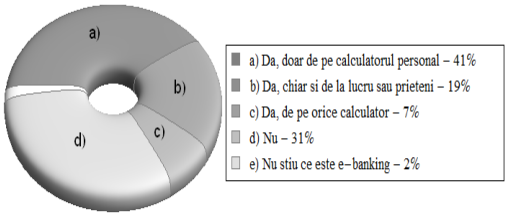
Folosiți antivirus?



Actualizați antivirusul folosit?

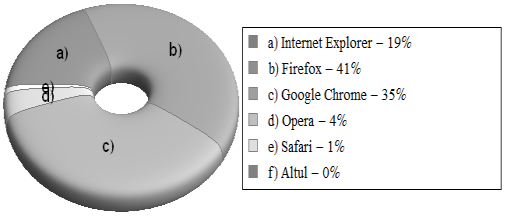


Folosiți mai multe soluții antivirus?

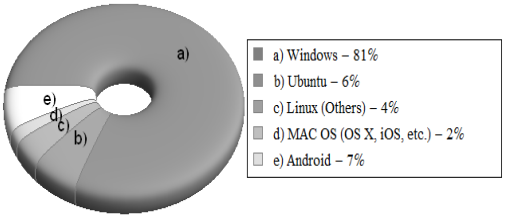


Folosiți servicii de e-banking?

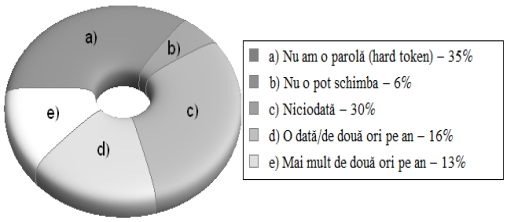
Figura 13 Chestionar securitate/uzabilitate 2



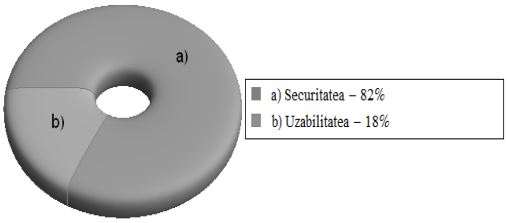
Ce browser preferați?



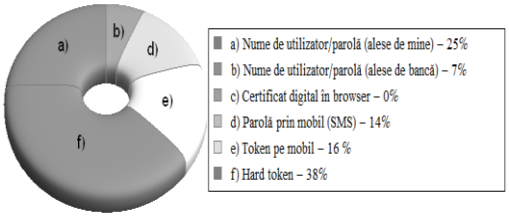
Ce sistem de operare preferați?



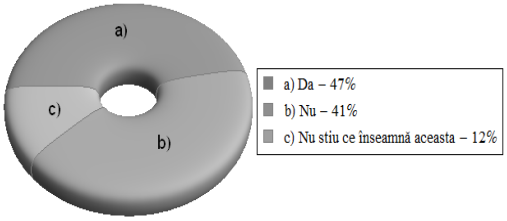
Cât de des vă schimbați parola de la aplicația de e-banking?



Ce considerați mai important securitatea sau uzabilitatea?

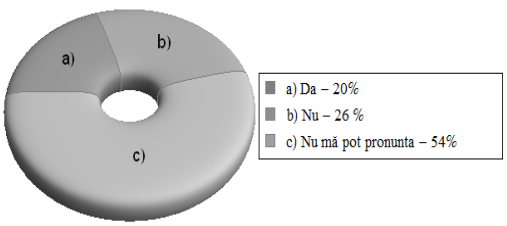


Ce tip de autentificare preferați?

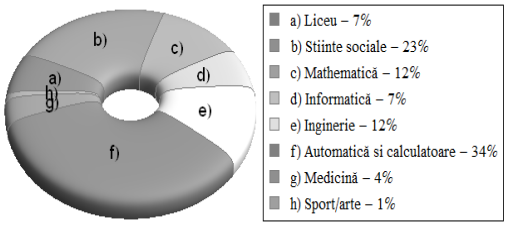


Verificați indicatorii de securitate din browser?

Figura 14 Chestionar securitate/uzabilitate 3



Consider că politica mea de securitate este corectă?



Ce nivel de educație aveți?

Figura 15 Chestionar securitate/uzabilitate 4

Deoarece nu există autorizare pentru a realiza efectiv atacuri DoS asupra aplicațiilor de e-banking, dar fiindcă mulți utilizatori au conturi la mai mulți furnizori de servicii de e-banking și aceștia pot fi tentați să renunțe la unii furnizori dacă aplicația lor nu răspunde corespunzător, s-a analizat și timpul de răspuns al aplicațiilor de e-banking. După cum se poate vedea în Figura 16, timpii de răspuns sunt asemănători pentru majoritatea băncilor, prezentând vârfuri în timpul pauzei de masă și după orele de muncă. Doar CitiBank are un timp de răspuns ce este vizibil mai lent, dar care nu afectează experiența utilizatorului, iar aplicația CEC Bank a fost indisponibilă pe perioada nopții (timp de răspuns 0). Măsurătorile au fost făcute de un script ce a rulat timp de 24 ore, în mai multe zile, rezultatele fiind aceleași.

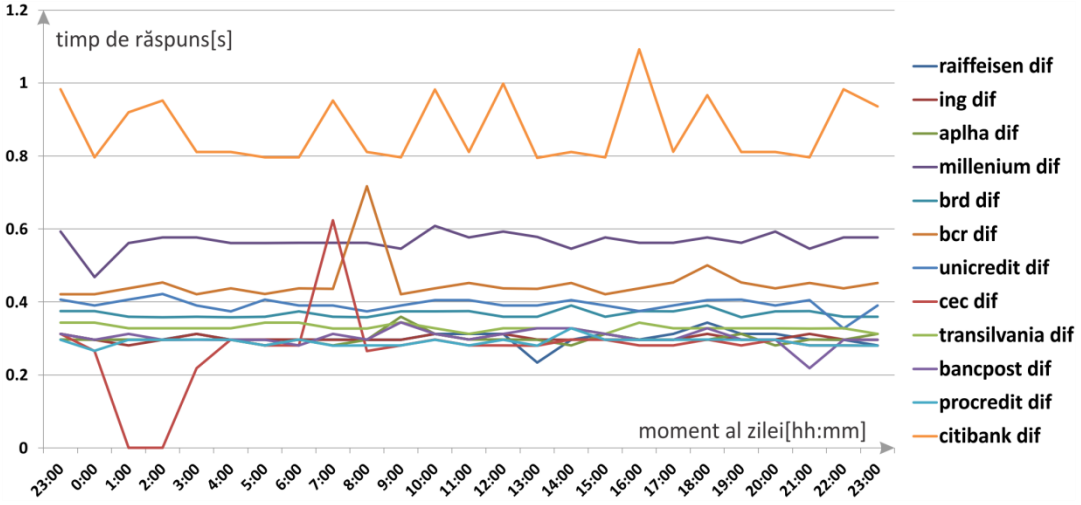


Figura 16 Timpul de răspuns a aplicațlor de e-banking pentru conexiuni HTTPS

3.2.6. Măsuri de protecţie în faţa atacurilor de tip inginerie socială

Datorită importanței acordate de utilizatori securității aplicațiilor de e-banking și a expansiunii acestui tip de serviciu financiar, în continuare vor fi prezentate principalele metode de protecție existente în fața atacurilor de tip inginerie socială.

3.2.6.1. Măsuri anti-phishing

În continuare vor fi prezentate metodele ce pot fi folosite de utilizatori pentru protecţia în faţa phishing-ului. Aceste metode sunt de tip software şi presupun instalarea de partea clientului a unei aplicaţii sau a unui plug-in ce are rolul de a avertiza utilizatorul în cazul în care acesta este pe cale de a accesa un site de phishing. O taxonomie a acestor metode de protecţie, prezentată în [30], împarte aceste soluţii în soluţii bazate pe *liste* sau soluţii *euristice.*

Soluţiile bazate pe liste utilizează fie *black-lists* (doar site-urile din listă sunt blocate) sau *white-lists* (doar site-urile din listă sunt permise) pentru a bloca încercarea utilizatorului de a accesa un site de phishing. Acest tip de soluţie este foarte popular în cadrul browser-elor, cum ar fi Microsoft SmartScreen Filter (http://www.microsoft.com) pentru Internet Explorer sau Google Safe Browsing API (https://developers.google.com/safe-browsing/) pentru Firefox şi Chrome.

Această abordare se bazează pe faptul că aceste liste sunt complete şi actuale, PhishThank (http://www.phishtank.com/) reprezintă o sursă pentru astfel de liste, dar acest lucru nu poate fi întotdeauna valabil deoarece durează un anumit interval de timp până când un site de phishing este adăugat în aceste liste, iar majoritatea site-urilor de phishing au o durată de viaţă foarte redusă [71]. De asemenea acest tip de protecţie nu poate detecta atacurile orientate spre un anumit utilizator sau grup de utilizatori.

Soluţiile bazate pe euristici încearcă să identifice diferite tipare în cadrul site-urilor de phishing. Aceste tipare pot fi identificate în URL-ul paginilor sau în cadrul codului HTML sau JavaScript al paginii.

În [76] se propune o soluţie bazată pe identificarea anomaliilor în cadrul obiectelor din DOM-ul paginii. Această soluţie are o rată de fals-negative de 12% şi o rată de fals-pozitive de 29%. Deoarece această metodă se bazează pe un identificator şi un clasificator terţ este necesară extinderea infrastructurii existente.

În funcţie de informaţia din codul HTML al paginii şi din URL, în [65] este propusă o soluţie anti-phishing ce are o rată de fals-pozitive de 0.43% şi o rată de fals-negative de 16.91%, dar această soluţie nu este disponibilă pentru un anumit tip de browser ci reprezintă doar o analiză off-line a datelor de pe diferite site-uri.

O metodă euristică bazată doar pe informaţii extrase din URL-ul paginii este prezentată în [40]. Deoarece este o soluţie bazată doar pe informaţii din URL aceasta nu va face faţă unui atac de tip pharming, în schimb are o rată de fals-pozitive de 0,7% şi o rată de fals-negative de 12%.

În [103] este propusă o soluţie anti-phishing ce utilizează algoritmul de clasificare TF-IDF împreună cu alte 7 euristici. Această soluţie are o rată de fals-pozitive de 1% şi o rată de fals-negative de 11%. Toate datele utilizate de euristici sunt extrase din text-ul paginilor web, dar cum majoritatea site-urilor web utilizează imagini, această metodă a fost îmbunătăţită în [30] prin utilizarea de OCR (Optical Character Recognition) pentru extragerea textului din imagini, obţinând astfel o rată de fals-negative de 2% şi o rată de fals-pozitive de 0%. Dar deoarece aceste euristici sunt bazate pe Google PageRank, un atac de tip pharming ce va încerca să impersoneze Google PageRank va trece de acest filtru.

3.2.6.2. Măsuri anti-pharming

Cu privire la pharming este propusă o soluţie în [85], aceasta se bazează pe trimiterea unei cereri DNS la un server DNS terţ. Dacă răspunsul este diferit de cel oferit de DNS-ul utilizatorului atunci este realizată o analiză bazată pe conţinutul HTML al paginii. Această soluţie nu poate fi eficientă dacă atacatorul poate determina victima să se conecteze printr-un proxy pe care acesta îl poate controla. De asemenea, dacă pagina de logare a site-ului este identică cu cea originală (ca în cazul unui atac de tip man-in-the-middle) atunci aceasta nu va fi identificată de către soluţia propusă.

3.2.6.3. Măsuri anti-MitB

Pentru a preveni astfel de atacuri, este important ca utilizatorii să conștientizeze riscul la care se expun în cazul în care instalează extensii pentru browser-ele lor din surse neverificate. Un model pentru asigurarea conștientizării utilizatorilor este prezentat în [61]. Pe lângă aceasta, au apărut și instrumente specializate pentru atenuarea efectelor unui atac MitB [45].

Ca urmare a pericolului reprezentat de atacurile de tip MitB, cercetarea s-a orientat și spre prevenirea lor, apărând soluții foarte interesante. Stahlberg oferă instrucțiuni băncilor pentru identificarea atacurilor de tip MitB prin analiza comportamentului utilizatorilor în relația lor cu aplicațiile de e-banking [94]. O altă soluție eficientă în fața atacurilor de tip MitB este reprezentată de ZTIC (Zone Trusted Information Channel) [101]. Soluția este bazată pe un dispozitiv USB cu display, dispozitivul fiind responsabil de autentificare, autorizare și criptare. Dacă se folosește ZTIC, PC-ul se ocupă doar de traficul pachetelor IP utilizate de aplicația de e-banking.

4. Contramăsuri DoS

Ca și contramăsuri DoS vor fi folosite tehnici proof-of-work implementate prin puzzle-uri criptografice, la început vor fi utilizate pentru protecția serverelor web. Apoi puzzle-urile criptografice de tip time-lock sunt aplicate pentru combaterea fenomenului spam. În final localizarea este întrebuințată pentru adaptarea dificultății puzzle-urilor criptografice aplicate pentru protecția DoS și filtrarea spam-ului. Amprentarea este prezentată ca și metodă mai precisă de identificare a adversarilor.

4.1. Protecţia în faţa atacurilor DoS asupra OpenSSL folosind puzzle-uri criptografice

OpenSSL (http://www.openssl.org/) reprezintă o implementare open source a protocoalelor SSL (Secure Sockets Layer) şi TLS (Transport Layer Security). Librăria (scrisă în limbajul C) implementează principalele funcţii criptografice şi asigură o serie de funcţii utilitare. Există deja o mulţime de implementări open source ale SSL şi pentru alte limbaje de programare (cum ar fi Java sau C#).

OpenSSL poate fi utilizat atât în sisteme de operare bazate pe Unix cât şi pe sisteme Windows.

OpenSSL oferă suport pentru următorii algoritmi criptografici:

- Coduri bloc: Blowfish, Camellia, DES, RC2, RC4, RC5, IDEA, AES;
- Funcţii hash: MD5, MD2, SHA, MDC-2;
- Criptografie cu cheie publică: RSA, DSA, schimb de chei Diffie-Hellman, curbe eliptice.

Aplicaţia prezentată în [23] reprezintă o extensie a OpenSSL, ce suportă standardele RFC 4366 [5] și RFC 4680 [88], și oferă suport pentru *puzzle-uri criptografice (Client Puzzles – noţiune întâlnită în* [27]*)* pentru a proteja serverele *HTTPS* de atacuri de tip DoS. Modelul atacului este prezentat în Figura 17.

După cum se vede în Figura 17 avem de-a face cu un *atacator* ce încearcă să convingă *clientul legitim,* prin intermediul poştei electronice sau al altui sistem de mesagerie electronică (pasul 3), de faptul că *serverul HTTPS legitim* este momentan indisponibil şi să se conecteze la un alt server temporar – *serverul HTTPS de phishing.* Pentru aceasta *atacatorul* va trimite o mulţime de cereri de verificare de certificat digital către *serverul legitim* (pasul 1) până când acesta nu va mai reuşi să răspundă la cererea *clientului* (pasul 2). Clientul crezând că *serverul legitim* este momentan indisponibil va încerca să se conecteze la *serverul de phishing* (pasul 4). Atacatorul reuşeşte astfel să obţină de la client informaţie senzitivă, cum ar fi CNP-ul sau numărul cărţii de credit.

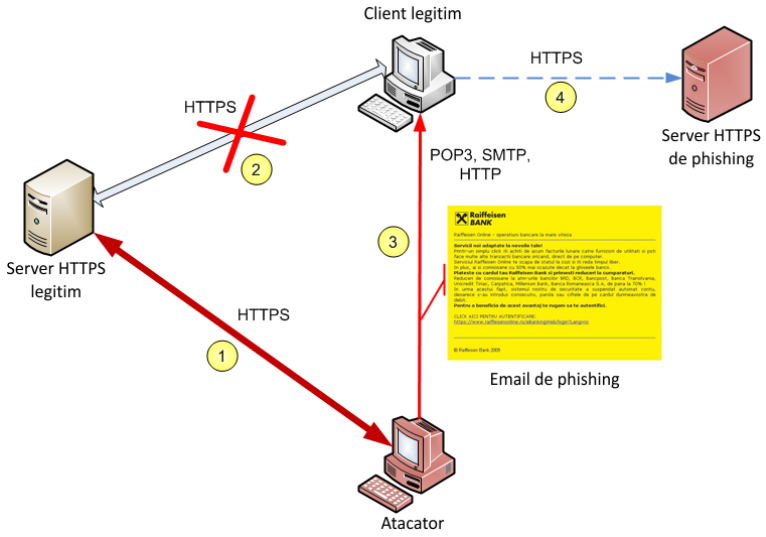


Figura 17 Modelul atacului [23]

Pentru a combate acest atac aplicaţia dezvoltată utilizează *puzzle-uri criptografice* pentru a preveni situaţia în care *atacatorul* face o multitudine de cereri de verificare, acesta trebuind să depună şi el, la rândul lui, efort computaţional pentru rezolvarea puzzle-ului primit de la *serverul legitim*.

Sistemul consistă dintr-un simulator de server HTTPS ce poate primi o cerere HTTPS şi returnează o pagina HTML simplă, serverul HTTPS suportă extensia pentru *puzzle-uri criptografice. Clientul legitim* este orice browser web ce suportă HTTPS. *Atacatorul* va fi un client ce poate rezolva *puzzle-uri criptografice*.

Pentru a înţelege funcţionarea sistemului trebuie prezentată diagrama de secvenţă (Figura 18) a sistemului:

În care:

1. *ClientHello* şi *ServerHello* cu *extensie TLS* nu sunt modificate, ele putând fi utilizate de client pentru a confirma faptul că suportă *puzzle-uri criptografice;*
2. După ce serverul a trimis ServerHello va fi apelată funcţia *server_subb_data_cb()* ce va genera *puzzle-ul* şi îl va trimite clientului ca şi *dată suplimentară pentru handshake;*
3. În momentul în care clientul primeşte *puzzle-ul* se intră în funcţia *client_supp_data_cb()*, aceasta având rolul de a rezolva *puzzle-ul* şi de a trimite soluţia către server;
4. *Handshake-ul* continuă până ce server-ul ajunge în starea *SERVER_DONE;*

5. Pe server se va intra în funcţia *server_finish_cb()* unde se va verifica corectitudinea răspunsului. Dacă răspunsul este corect se va deschide conexiunea cu clientul altfel această conexiune nu va fi deschisă;
6. Dacă s-a deschis conexiunea clientul poate schimba date cu serverul la nivel de aplicaţie, adică va putea cere pagina web iar serverul îi va putea servi această pagină.

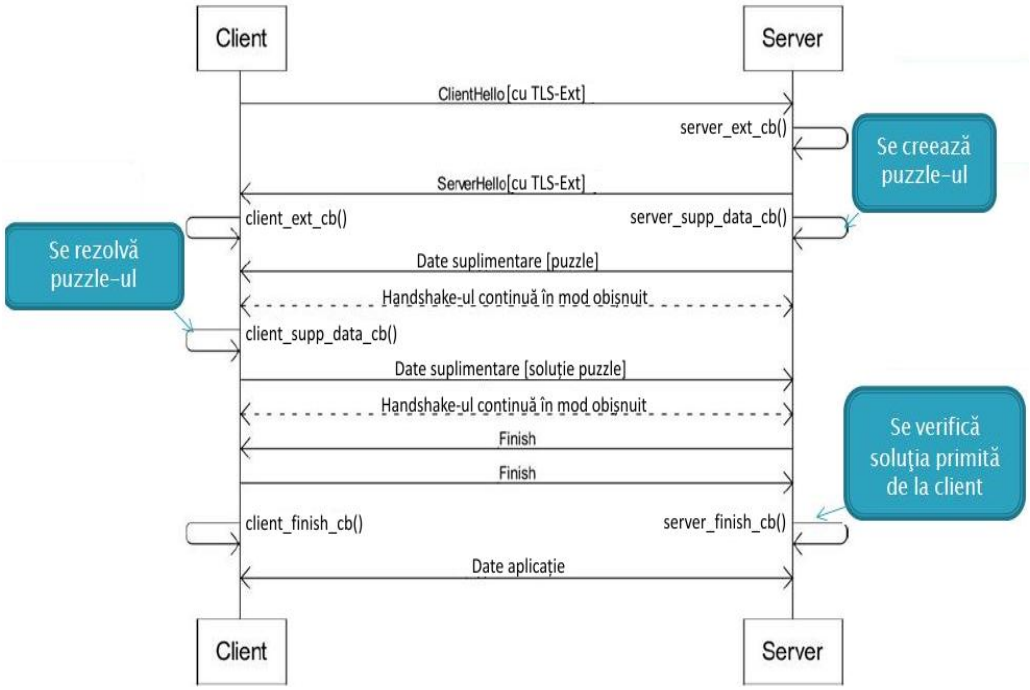


Figura 18 Diagrama de secvenţă a sistemului

Pentru a sesiza rolul puzzle-lor criptografice în protecţia serverelor HTTPS împotriva atacurilor DoS s-au realizat următoarele grafice:

a) evoluţia timpului de rezolvare a puzzle-lor în cazul unui atac (Figura 19):

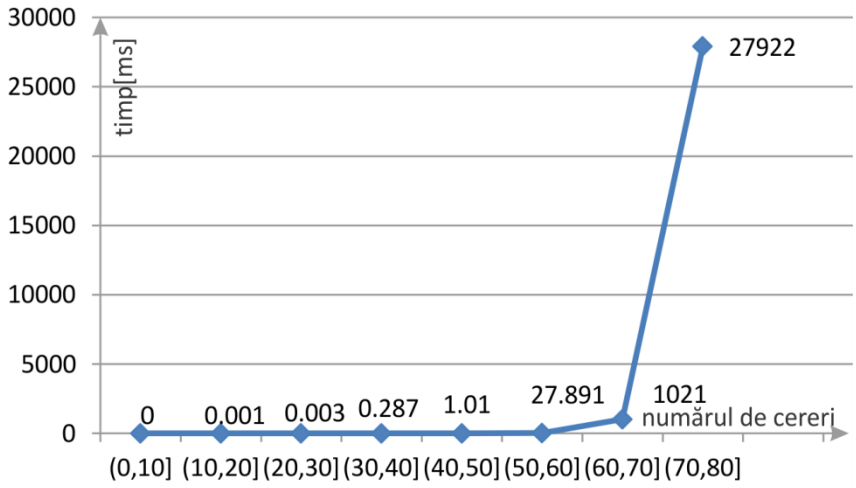


Figura 19 Timpul de rezolvare a puzzle-ului

Pentru a genera acest grafic s-a utilizat adaptarea lungimii puzzle-ului în funcţie de numărul de cereri făcute de client la server.

b) evoluţia latenţei pentru un client legitim, în timpul unui atac DoS, în cazul în care nu s-au utilizat puzzle-uri criptografice (Figura 20):

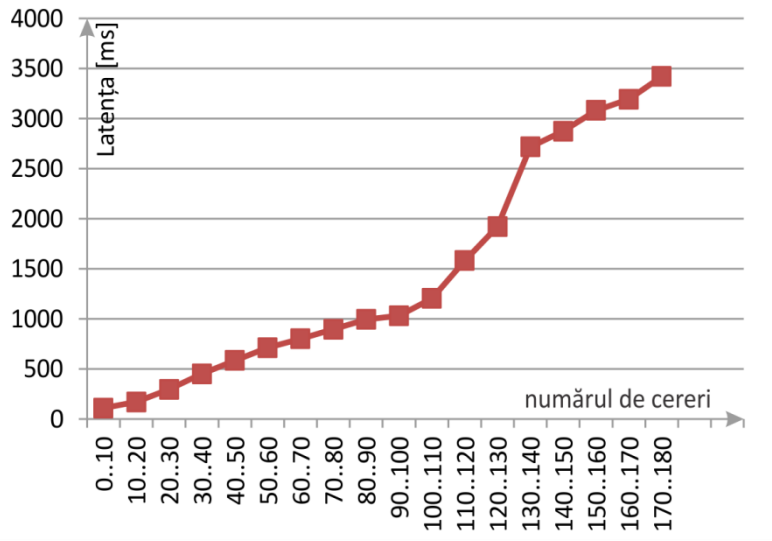


Figura 20 Latenţa unui client legitim în modul fără puzzle

c) evoluţia latenţei pentru un client legitim, în timpul unui atac DoS, în cazul în care s-au utilizat puzzle-uri criptografice (Figura 21):

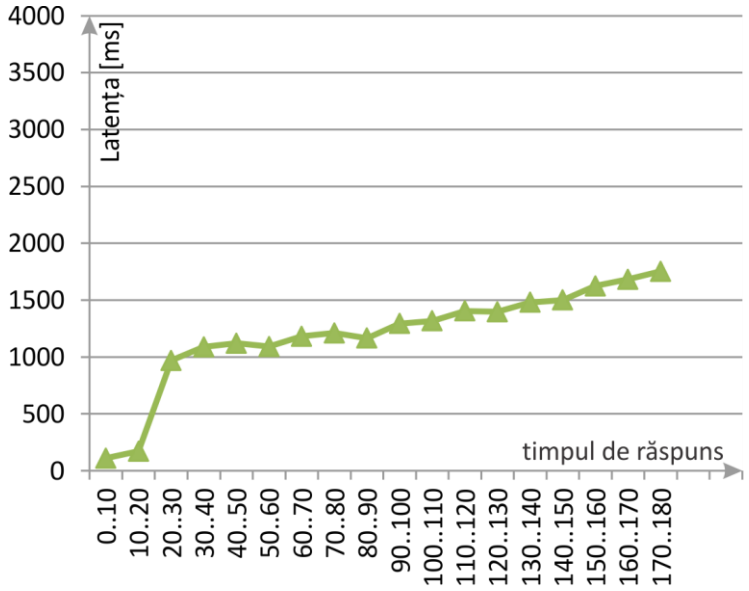


Figura 21 Latenţa unui client legitim în modul cu puzzle

Pentru a genera acest grafic s-a utilizat un puzzle criptografic de dimensiune constantă iar puzzle-urile criptografice vor fi trimise la client din momentul în care acesta a depăşit 20 de cereri.

Ca şi comparaţie se observă că dacă serverul aflat sub atac nu folosește puzzle-uri criptografice atunci acesta va avea o latență tot mai mare, aceasta continuând să crească până în momentul în care serverul nu va mai reuși să răspundă cererilor provenite de la clienții legitimi. În cazul în care serverul utilizează puzzle-uri criptografice, latența acestuia va crește odată brusc (după aproximativ 20 de cereri) datorită generării puzzle-urilor, după care va rămâne relativ constantă, acesta reușind să servească pagina web cu succes clienților săi legitimi.

4.2. Sporirea rezistenței serviciilor de webmail împotriva epuizării resurselor prin mesaje de tip SPAM utilizând puzzle-uri criptografice

Pentru a îmbunătăţii rezistenţa în faţa atacurilor de tip SPAM în [20] s-a implementat un protocol bazat pe puzzle-uri criptografice pentru un client webmail. Soluţia propusă este compatibilă cu infrastructura existentă şi nu necesită modificări pe parte de server, ci doar prezenţa unui server de ticketing. Puzzle-urile folosite sunt de tip time-lock deoarece acestea nu pot fi victima unui atac paralelizabil.

4.2.1. Descrierea protocolului

Participanţii la comunicare sunt descrişi în Figura 22:

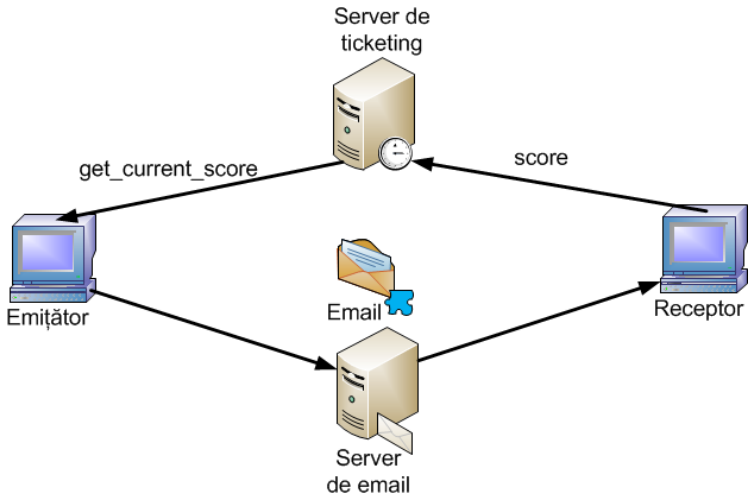


Figura 22 Participanţii la comunicare [20]

Emiţătorul va trebui să rezolve un puzzle pentru fiecare email pe care îl trimite, dificultatea acestui puzzle este în funcţie de un scor obţinut pentru email-urile anterioare. Acest scor este actualizat în mod continuu de către receptori, prin notificarea serverului de ticketing (TS). Inițial dificultatea puzzle-urilor va fi mică, dar aceasta va crește odată cu numărul de email-uri trimise de emițător (în implementarea realizată se dublează dificultatea) ce sunt identificate ca fiind spam de către un filtru de spam apelat de serviciul de webmail. Emițătorul poate refuza să rezolve puzzle-ul, dar aplicația de webmail va marca automat email-urile sale ca fiind de tip spam.

Serverul de ticketing este responsabil de diferite sarcini (distribuirea detaliilor legate de puzzle-uri, scor și orice altă informație necesară rezolvării puzzle-urilor specifice fiecărui email) și asigură sincronizare temporală între participanții la comunicare. Serverul de email nu are nevoie de nici o modificare.

Soluția puzzle-ului ce va fi atașată fiecărui email, trebuie să depindă de o valoare proaspătă și impredictibilă pentru evitarea un atac prin reluare (replay) precum și rezolvarea puzzle-urilor offline. Această valoare este dată de un timestamp furnizat de serverul de ticketing, semnată. Astfel, un adversar nu poate prezice această semnătură, deci nu poate rezolva un puzzle înainte ca această valoare să fie generată de serverul de ticketing. Pentru a evita situația în care emițătorul trebuie să refolosească soluția unui puzzle în același interval dintre două timestamp-uri (presupunem că server-ul de ticketing semnează timestamp-uri cu o periodicitate regulată), emițătorul va trebui să adauge un *nonce* puzzle-ului. Iar receptorul va trebui să stocheze toate soluțiile primite de la emițător pentru a verifica unicitatea lor.

Astfel, când A dorește să-i trimită un email la B va genera un puzzle în felul următor: $h(sign_{TS}(t_{TS}), nonce_i, email_i, B, A)^{\varepsilon} \bmod n_B$, unde n_B reprezintă modulul receptorului *B*, $nonce_i$ este nonce-ul generat pentru email-ul *i*, $sign_{TS}(t_{TS})$ reprezintă timestamp-ul semnat de către serverul de ticketing, iar ε este un exponent de forma 2^k unde *k* este dificultatea puzzle-ului, acesta depinzând de scorul emițătorului: $\varepsilon = f(score_{AB})$.

Soluția acestui puzzle nu ocupă mult spațiu, deoarece puzzle-ul este de mică dimensiune, iar pentru a găsi dacă puzzle-ul a mai fost sau nu rezolvat anterior se poate utiliza o căutare binară.

Astfel, un email poate fi marcat ca fiind spam dacă una din următoarele condiții este îndeplinită:

i. Soluția puzzle-ului nu este corectă;
ii. Soluția puzzle-ului nu este asociată cu timestamp-ul corespunzător (datorită întârzierilor din rețea se poate utiliza o marjă de siguranță de câteva minute, ore, etc.);
iii. Aceeași soluție este refolosită pentru mai multe email-uri.

Pe scurt, protocolul funcționează în modul următor: un client care dorește să trimită un email trebuie să rezolve un puzzle conform specificațiilor din certificatul receptorului (modul) și a scorului său, toate aceste informații fiind obținute de la TS. Soluția puzzle-ului este adăugată la email iar receptorul o va verifica în momentul în care recepționează email-ul.

Pașii protocolului sunt prezentați mai jos. Săgeata dublă reprezintă un canal de comunicare sigur, acesta poate fi implementat folosind SSL/TLS. Canalul de comunicare cu TS trebuie să fie sigur pentru a asigura autenticitatea mesajelor schimbate cu acesta. Singurul canal de comunicare ce nu are nici o restricție este cel cu serverul de email. De asemenea în protocol se poate folosi un alias ($alias_X$) pentru ca TS să nu poată monitoriza participanții la comunicare. Astfel, participanții se pot înregistra pe TS folosind un alias aleatoriu.

Comunicare cu TS în momentul transmiterii unui email:

$1. A \twoheadrightarrow TS : alias_B;$

$2. TS \twoheadrightarrow A : cert_B, \varepsilon, sign_{TS}(t_{TS}).$

Expedierea email-ului i în intervalul δ:

$1. A \rightarrow MS : email_i, sign_{TS}(t_{TS}), nonce_i,$

$puzzle = h(sign_{TS}(t_{TS}), nonce_i, email_i, B, A)^{\varepsilon} \bmod n_B.$

Comunicare cu TS după primirea email-ului:

$1. B \twoheadrightarrow TS : alias_A, score_{AB}.$

Certificatul $cert_B$ conține modulul n_B al receptorului.

4.2.2. Evaluarea costului

Pentru a evalua performanța protocolului propus, e important să se analizeze profitul spamer-ului, acesta depinde de următorii parametrii:

- n – numărul de clienți țintă ai spamer-ului;
- c_{send} – costul trimiteri unui email;
- c_{puz} – costul rezolvării unui puzzle;
- a – factorul de creștere al dificultății puzzle-ului;
- r – câștigul spamer-ului per email;
- p – probabilitatea ca un spam să fie identificat corect.

Profitul spamer-ului trebuie analizat pentru k sesiuni. În prima sesiune profitul spamer-ului va fi: $n(r-(c_{send}+c_{puz}))$. În cea de-a doua sesiune spamer-ul va trebui să plătească un cost dublu pentru email-ul trimis clienților ce au identificat email-ul din prima sesiune ca fiind spam și nici un alt cost pentru ceilalți. Probabilitatea ca un email de tip spam să fie identificat de un filtru de spam este, în general, binomială, astfel pentru k sesiuni avem:

$$\text{Profit} = \sum_{i=0}^{k} n \cdot (r - (c_{send} + 2^i c_{puz})) \cdot p^i \cdot (1-p)^{k-i} \cdot \frac{k!}{i!(k-i)!}$$

Formula de mai sus este valabilă în cazul în care dificultatea puzzle-ului se dublează ($a=2$), după fiecare sesiune.

Pentru a face o estimare concretă a profitului, valorile parametrilor trebuie alese în funcție de context, lucru deloc ușor. Pentru analiză s-au folosit valorile propuse de Laurie și Clayton în lucrarea [62]. Astfel, pentru expedierea unui număr de 15000 de email-uri într-o zi, costului expedierii unui email este de 0,005 cenți, având în vedere costul pentru electricitate (25 cenți/zi), și al unui calculator (50 cenți/zi). În total expedierea celor 15000 de email-uri într-o zi costă 75 cenți. Pentru a calcula costul rezolvări unui puzzle se poate pleca de la aceleași valori, acesta depinzând de timpul rezolvării unui puzzle. Plecând de la prețul unei zi de calcul (75 cenți) se poate deduce că 1ms de calcul costă aproximativ 0.003 cenți. Pentru calcularea valorii lui p se pot folosi valorile din [90]. Conform lucrării, filtrele de spam au o rată de detecție de 60%, cele mai bune ajungând la 90%. Astfel, putem considera $p=0,6$. De asemenea, trebuie luate în calcul și pierderile emițătorului datorate unei clasificări greșite. Pentru aceasta se consideră p' rata fals-pozitivelor și avem:

$$\text{Pierdere} = \sum_{i=0}^{k} n \cdot 2^i \cdot c_{puz} \cdot p'^i \cdot (1-p')^{k-i} \cdot \frac{k!}{i!(k-i)!}$$

Folosind aceste relații, în Figura 23 se reprezintă profitul spamer-ului și pierderea unui emițător pentru cazul în care $n=100000$ (un spamer vizează de la

câteva mii de ținte până la câteva milioane, am ales 100000 de ținte pentru exemplificare) . După cum se poate vedea, după expedierea a aproximativ 10 email-uri profitul spamer-ului scade sub 0. Pentru o rată de creștere polinomială ($i^{4/3}$) a dificultăți puzzle-lor și pentru o dificultate inițială mai mare (x10), profitul spamer-ului este și mai mic, dar pierderile în cazul fals-pozitivelor sunt mai mari.

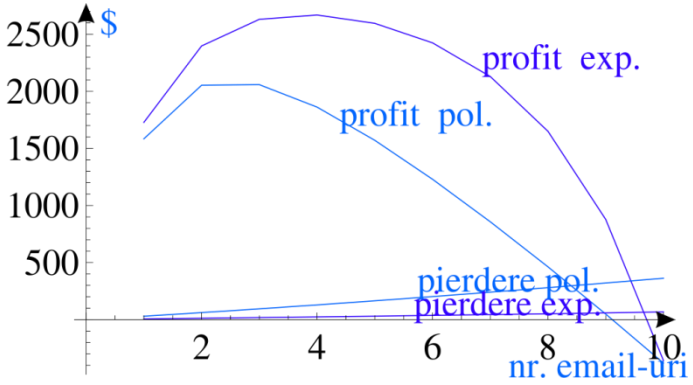


Figura 23 Profitul spamer-ului și pierderea computațională pentru un factor de creștere exponențial, respectiv polinomial, a dificultății puzzle-lor

O potențială limitare a acestui protocol stă în faptul că spamer-ul trebuie să plătească doar în sesiunea următoare. Pentru a evita această situație, email-ul poate fi trimis *Serverului de Ticketing* (TS – Ticketing Server) TS-ului pentru a fi evaluat, pe TS poate rula un filtru de spam, iar dacă email-ul este identificat ca fiind spam, costul asociat email-ului este ajustat în mod corespunzător. Cu toate acestea, ne interesează spameri care acționează pe termen lung (cum ar fi companiile ce se ocupă cu spam) și care vor reveni pentru a trimite spam.

4.2.3. Implementare și rezultate experimentale

Utilizarea serviciului de email este bazată pe două protocoale pentru recepţia email-urilor (POP3 şi IMAP) şi un protocol pentru expedierea acestora (SMTP). În lucrare protocolul IMAP va fi folosit deoarece email-urile sunt stocate pe server. Pentru a realiza filtrarea spam-ului fiecare emiţător va fi forţat să genereze un puzzle în funcţie de cerinţele receptorului. Pentru aceasta se vor adăuga două antete mesajului email: *puzzle* acesta reprezintă soluţia la puzzle-ul generat de emiţător şi $nonce_l$ care este o valoare aleatoare folosită la generarea puzzle-ului. Procesul expedierii şi recepţiei unui email este prezentat în Figura 24:

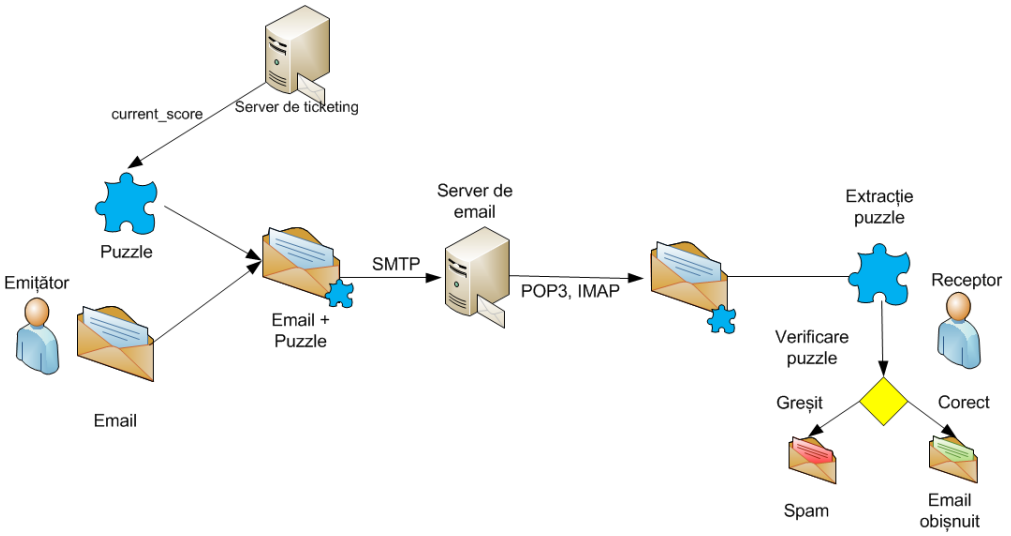


Figura 24 Procesul de emise/recepţie a email-ului [20]

Introducerea puzzle-urilor de tip time-lock este transparentă pentru serverul de email, dar pentru a implementa soluţia, avem nevoie de următoarele:

- un client de email capabil să calculeze şi să verifice puzzle-uri de tip time-lock. Aceasta s-a realizat prin extinderea *Yawebmail* (http://yawebmail.sourceforge.net/);
- un server de ticketing prin care clienţii îşi pot face publice informaţiile necesare comunicării.

Yawebmail este o aplicație open source scrisă în Java. Pentru implementarea mecanismului de filtrare a spam-ului prin puzzle-uri de tip time-lock Yawebmail a fost extinsă cu două pachete *security* (Figura 25) şi *data*.

Pachetul *security* este responsabil de generarea, rezolvarea şi verificarea puzzle-lor time-lock. Acestea sunt realizate în clasa *PuzzleUtils*. Generarea şi rezolvarea puzzle-lor este realizată de către metoda *PuzzleUtils.generatePuzzle()* după cum urmează:

- se generează o valoare aleatoare $nonce_i$;
- se calculează $puzzle = h(t_\delta + nonce_i + email_i + adr_B + adr_A)^{2^k} \bmod n_B$, folosind k iteraţii, unde n_B reprezintă modulul receptorului, *email* conţinutul email-ului, t_δ reprezintă amprenta de timp semnată primită de la serverul de ticketing, adr_A/adr_B adresa de email a emiţătorului/receptorului.

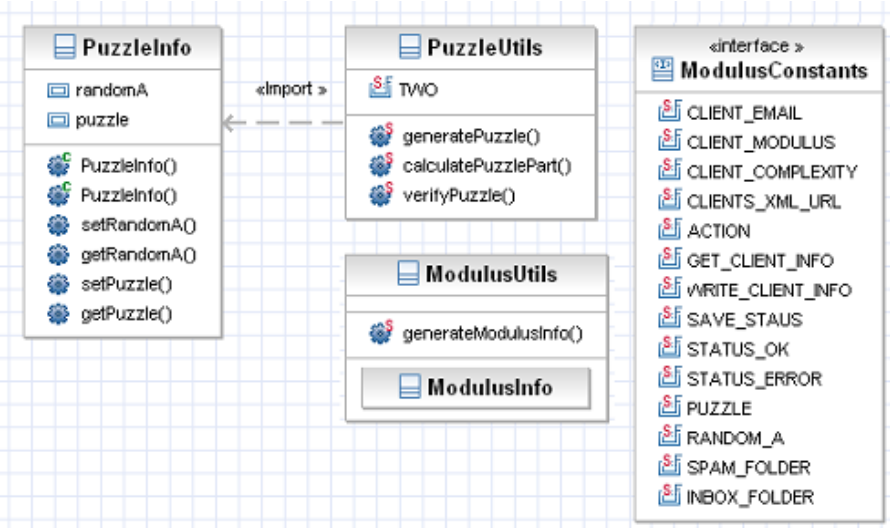


Figura 25 Pachetul *security* [20]

După ce soluţia puzzle-ului a fost obținută, valorile $nonce_i$ şi *puzzle* sunt trimise ca şi antete suplimentare ale mesajului email.

Verificarea puzzle-ului se realizează de către metoda *PuzzleUtils.verifyPuzzle()*:

- se calculează $r=2^k mod \varphi(n)$, iar apoi $b=h(t_\delta+nonce_i+email_i+adr_B+adr_A)^r$ unde $\varphi(n)=(p-1)(q-1)$, iar p şi q sunt două numere prime mari;
- dacă *puzzle*=*b* atunci puzzle-ul a fost rezolvat corect şi email-ul nu va fi marcat ca şi spam.

De asemenea pe receptor va rula un filtru de spam (jASEN) (http://www.jasen.org/) pentru a obţine scorul pentru email-ul primit. În cazul în care acest email este identificat ca şi spam costul pentru emiţător va fi crescut cu un factor de valoare 2.

Pachetul *data* este utilizat pentru a ţine informaţiile necesare pentru puzzle-uri.

S-a analizat efortul computaţional pentru puzzle-urile de tip time-lock pe 4 maşini de test diferite:

1. Laptop cu procesor Intel Centrino 1.73 GHz, 1 GB RAM, cu Windows XP SP3;
2. PC cu procesor Intel Core 2 Quad 2.4 GHz, 4 GB RAM, cu Windows Vista Business SP1 32 bit;
3. PC cu procesor Intel Core 2 Duo 2.66 GHz, 2 GB RAM, cu Windows XP SP2;
4. Telefon Nokia N95 8GB cu procesor Dual ARM 11 333 MHz, 128 MB RAM, cu Symbain OS 9.1, S60 rel 3.1.

Timpul necesar unei multiplicări este prezentat în Tabelul 3 de mai jos – pentru versiunea mobilă a aplicaţiei s-a folosit implementarea de *BigInteger* din *BouncyCastle.*

Modul / **Maşina**	**512 BITS**	**1024 BITS**	**2048 BITS**	**4096 BITS**
(1)	$141x10^{-6}s$	$422x10^{-6}s$	$1573x10^{-6}s$	$6000x10^{-6}s$
(2)	$76x10^{-6}s$	$225x10^{-6}s$	$763x10^{-6}s$	$2950x10^{-6}s$
(3)	$78x10^{-6}s$	$208x10^{-6}s$	$703x10^{-6}s$	$2672x10^{-6}s$
(4)	$27x10^{-3}s$	$69x10^{-3}s$	$168x10^{-3}s$	$641x10^{-3}s$

Tabelul 3 Comparaţie între timpii de rezolvare a puzzle-ului [20]

După cum se poate vedea în tabelul de mai sus un emiţător onest nu va fi afectat de trimiterea puzzle-ului, în schimb un spamer va fi afectat deoarece fiecare mail care este filtrat ca spam va duce la creşterea dificultăţii puzzle-ului cu factorul 2, deci profitul spamer-ului va scădea.

4.3. Sporirea eficienței puzzle-urilor criptografice împotriva DoS și SPAM prin utilizarea coordonatelor sintetice verificabile

Puzzle-uri criptografice, deși eficiente în anumite situații, au totuși anumite probleme, după cum se arată în [44] și [62], fiind recomandată utilizarea lor în combinație cu alte tehnici. De aceea, în continuare, locația adversarului va fi folosită pentru ajustarea dificultății puzzle-urilor.

Dintre algoritmii de localizare prezentați în capitolul 2, în [21] Vivaldi a fost ales pentru localizare sigură deoarece este foarte răspândit (de exemplu în aplicații cum ar fi BitTorrent), se scalează la nivelul Internetului, este ușor, are complexitate redusă și nu face mult trafic în rețea, având nevoie de mai puține măsurători (în comparație cu soluțiile alternative) pentru a converge spre stare în care eroarea de estimare este minimă (mai mică decât un anumit prag de eroare).

Dar, bazându-se doar pe RTT pentru localizarea în sistemul de coordonate virtuale, Vivaldi nu poate detecta dacă un nod minte în legătură cu poziția lui sau a erori de localizare [58]. Pentru a depăși această problemă s-a propus un mecanism alternativ la măsurarea RTT-ului. Astfel, în loc să măsoare RTT-ul direct, nodurile din sistemul de coordonate virtuale vor folosi o funcție de distanță bazată pe nonce-uri distribuite ritmic.

4.3.1. Localizarea în sistemul de coordonate Vivaldi

Vivaldi este un sistem de coordonate virtuale cu două dimensiuni standard, folosite pentru măsurarea întârzierilor în rețeaua centrală, și o dimensiune suplimentară, numită înălțime, folosită pentru măsurarea întârzierilor dintre rețeaua de acces și rețeaua centrală.

Deși coordonatele sferice pot părea mai naturale într-un astfel de caz de localizare, datele experimentale [24], [74] nu susțin această idee deoarece canalele de comunicare, din infrastructura existentă, nu înfășoară pământul. Coordonatele Euclidiene fiind cele mai eficiente. Astfel, considerând două puncte $P_1(X_1,Y_1,P_1^h)$ și $P_2(X_2,Y_2,P_2^h)$, folosind coordonate Euclidiene, următoarele ecuații sunt valide:

$$|P_1| = \sqrt{X_1^2 + Y_1^2} + P_1^h$$
$$\alpha \times P_1 = (\alpha X_1, \alpha X_2, \alpha P_1^h)$$
$$P_1 - P_2 = (X_1 - X_2, Y_1 - Y_2, P_1^h + P_2^h).$$

Aceste coordonate pot fi folosite atâta timp cât inegalitatea triunghiului este validă. Dar sunt cunoscute situații, în Internet, când această inegalitate este violată. Astfel, după cum este sugerat și în [24] inegalitatea triunghiului trebuie relaxată cu o anumită constantă ε. De exemplu pentru ε = 5ms doar 4.5% din setul de date mai violează inegalitatea triunghiului [24].

Vivaldi se bazează pe principiul că fiecare nod trebuie să-și determine coordonatele, astfel încât eroarea de estimare a RTT-ului dintre el și celelalte noduri din rețea să fie minimă. Având un resort fizic (vezi Figura 26) RTT-ul măsurat poate fi văzut ca fiind lungimea resortului relaxat RTT_{ij}, iar RTT-ul estimat ca fiind lungimea resortului extins $|P_i-P_j|$. Eroarea de estimare va fi $d=|P_i-P_j|-RTT_{ij}$, conform legii lui Hooke dacă $d \to 0 \Rightarrow E_{ij} = -kd \to 0$, unde E_{ij} reprezintă valoarea forței ce încearcă să aducă resortul în starea de repaos. Urmând această analogie, nodurile din rețea pot fi văzute ca și când ar fi conectate între ele prin resorturi. Iar pentru a le localiza, funcția: $E = \sum_i \sum_j E_{ij} = \sum_i \sum_j (RTT_{ij} - |P_i - P_j|)$ trebuie minimizată. Acest lucru este echivalent cu a aduce întreg sistemul nod-resort în starea de repaos.

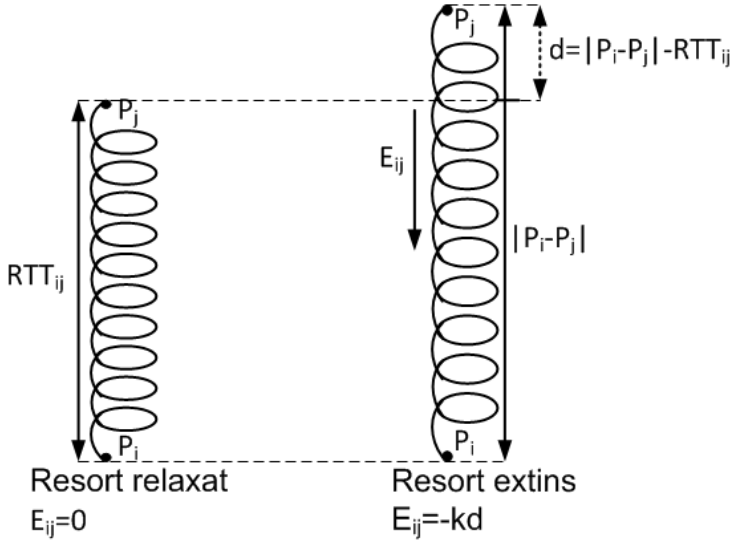


Figura 26 Relația dintre legea lui Hooke și eroarea de estimare folosită de algoritmul Vivaldi

Algoritmul Vivaldi, bazat pe algoritmul prezentat în [24] este descris în Secvența 2.

```
      /*Calculează poziția lui P_1 în funcție de distanța d_12
dintre P_1 și  P_2, poziția lui P_2 și eroarea locală e_2 a
altui nod din rețea*/
      function vivaldi(d_12, P_2, e_2){
         /*w - ponderea*/
         w = e_1/(e_1 + e_2);

         /*e_s – eroarea relativă*/
         e_s = abs(dist(P_1, P_2) - d_12)/d_12;

         /*e_1 – eroarea locală
         c_e, c_c – constante de ajustare*/
         e_1 = e_s*c_e*w+e_1*(1 - c_e*w);
         step = c_c*w;

         /*direcția noi poziții*/
```

```
        dir = (P_1 - P_2)/dist(P_1, P_2);
        P_1 = P_1+step*(d_12-dist(P_1,P_2))*dir;

        return P_1;
    }
```

Secvența 2 Determinarea poziției unui nod din rețea folosind algoritmul Vivaldi

4.3.2. Nonce-uri ritmice

Noțiunea de nonce ritmic a fost introdusă în [13] ca și sursă pentru puzzle-uri criptografice folosite în combaterea DoS. Un nonce ritmic este definit ca un număr din secvența $(u_i) \in A^{\mathbb{N}}$ cu proprietatea că nu se poate determina u_{i+1} cunoscând $u_i,u_{i-1},\ldots,u_0$ și $\forall k > 0$ există o funcție, ușor de calculat $d_k(u_i,u_j)=min(|i-j|,k)$, numită funcție distanță.

Se va folosi o noțiune similară dar formalismul va fi cel prezentat mai jos, deoarece în [13] noțiunile de securitate sunt informale (de exemplu nu este clar care este parametrul de securitate), iar rezultatele experimentale se bazează pe valori RSA nesigure.

Definiția 1. O secvență $S_k=\{s_0,s_1,\ldots,s_n\}$ împreună cu funcția de temporizare $\tau : S_k \to \mathbb{N}$ și funcția distanță $\delta : S_k \to \mathbb{N}$ formează o secvență de nonce-uri ritmice $RNS = \{S,\tau,\delta\}$ ce are parametrul de securitate k dacă:

1. pentru $\forall s_i, i \in [1..n]$ funcția de temporizare $\tau(s_i)$ va avea o valoare întreagă t, ce reprezintă momentul în care nonce-ul s_i a fost transmis;
2. pentru $\forall s_i, s_j, i, j \in [1..n]$ funcția distanță $\delta(s_i,s_j)$ va avea o valoare întreagă t, ce reprezintă timpul dintre două nonce-uri;
3. pentru orice adversar de timp polinomial ce deține $\{s_0,s_1,\ldots,s_t\}$ nu este fezabil să calculeze s_{t+1}, cu excepția unei probabilități neglijabile $v(k)$, de parametru de securitate k:

$$\Pr[s_{t+1} \to Adv(\{s_0, s_1, \ldots, s_t\})] \le v(k).$$

Observația 1. Orice funcție one-way (neinversabilă) f poate fi folosită pentru generarea secvenței $\{s_0,s_1,\ldots,s_t\}$ în modul următor: $s_i=f(s_{i-1}), i=0..n$, unde s_0 este ales în mod aleator. Definiția 1 este valabilă în acest caz, aceasta se poate demonstra ușor prin reducere la problema inversării funcției one-way.

Observația 2. Funcțiile de temporizare și de distanță pot fi definite folosind timpul transmiterii primului nonce t_0 și timpul dintre generarea a două nonce-uri consecutive: $\tau(s_i)=t_0+i\delta$, $\delta(s_i,s_j)= \delta\ (i-j)$.

Observația 3. O metodă alternativă de generare a unei astfel de secvențe poate fi realizată folosind semnături digitale, totuși astfel de semnături sunt mai solicitante computațional și au o amprentă mai puternică asupra traficului din rețea. Ca și metodă alternativă se pot folosii funcțiile hash, dar acestea au un domeniu de valori limitat. Cea mai bună alternativă o reprezintă funcțiile one-way, folosite în criptografia cu cheie publică, de tipul $f(x)=x^2 \bmod n$, acestea pot genera lanțuri de lungime infinită. Se poate apela și la lanțuri mixte de funcții hash, cu semnături digitale, acestea fiind folosite la reinițializarea lanțului. În plus, în loc de exponentul par 2 se poate folosi orice exponent impar ce este relativ prim la $\varphi(n)$, acești exponenți permițând lanțuri de lungime nelimitată.

Pentru a avea o privire de ansamblu asupra cerințelor computaționale ale metodei de localizare, în Tabelul 4 sunt prezentați timpii de calcul pentru multiplicare și exponențiere modulară; iar în Tabelul 5 timpii de calcul pentru funcțiile hash ce pot fi folosite la localizare.

	Modul			
Funcție	512 bit	1024 bit	2048 bit	4096 bit
$x^2 n$	$62.5\times10^{-6}s$	$117\times10^{-6}s$	$171.5\times10^{-6}s$	$390\times10^{-6}s$
$x^d n$	109.5×10^{-6}	$164\times10^{-6}s$	$429\times10^{-6}s$	$1497.5\times10^{-6}s$

Tabelul 4 Timpii de calcul pentru multiplicare și exponențiere modulară (d=n-1)

Măsurătorile au fost făcute folosind clasa *BigInteger* din limbajul de programare *Java,* iar testele au rulat pe un PC cu un procesor *Intel Quad Core* de 2,4 Ghz, 4GB memorie RAM și cu un sistem de operare *Windows 7 x64.*

	x	
Funcție	128 bit	256 bit
MD5(x)	$30.5\times10^{-6}s$	$31.5\times10^{-6}s$
SHA-256(x)	$39\times10^{-6}s$	$39\times10^{-6}s$

Tabelul 5 Timpii de calcul pentru funcțiile hash folosite în localizare

4.3.3. Algoritmul Vivaldi modificat

În loc să măsoare RTT-ul, un nod nou N_a ce a intrat în rețea va asculta nonce-urile ritmice transmise de serverele de nonce-uri ritmice *(RNS – Rhythmic Nonce Server).* Acestea se află în rețea iar coordonatele și erorile lor locale de localizare sunt disponibile public prin intermediul unui serviciu public de localizare *(PLS – Public Localization Service).* Aceste RNS-uri sunt sincronizate și își actualizează în mod continuu poziția folosind același algoritm Vivaldi.

4.3.3.1. Stabilirea locației

Pentru a-și determina poziția, nodul N_a trimite o *cerere* de localizare de tip broadcast (notată cu $\rightrightarrows$). Când va primi această cerere, *RNS* îi va răspunde cu o fereastră de *w* nonce-uri: s_{i+1}, s_{i+2},…, s_{i+w}. N_a va răspunde, la rândul lui, cu valoarea hash a fiecărui nonce: $h(s_{i+1})$, $h(s_{i+2})$,…, $h(s_{i+w})$. Această comunicare are loc în mod asincron (notată cu $\rightsquigarrow$). Adică *RNS* va transmite nonce-uri cu aceeași periodicitate iar N_a răspunde imediat ce a primit un nonce, fără să aștepte după următoarele. Presupunând că în momentul în care primește cererea de la N_a, cel mai recent nonce transmis de *RNS* este s_i, *RNS* va aștepta până se ajunge la perioada generării nonce-ului următor ca sa-i răspundă la N_a. Acest fapt nu va introduce întârzieri semnificative deoarece nonce-urile sunt transmise cu o periodicitate de ordinul milisecundelor, iar într-o observație următoare este prezentat modul în care acest fapt contribuie la acuratețea localizării.

Când fereastra *w* s-a închis, *RNS* v-a calcula distanța $d(N_a,RNS)$ și o va trimite la N_a. Acest proces este descris mai jos:

1.) $N_a \rightrightarrows RNS : cerere$,

2.) $RNS \rightsquigarrow N_a : s_{i+1}, s_{i+2}, \ldots, s_{i+w}$,

3.) $N_a \rightsquigarrow RNS : h(s_{i+1}), h(s_{i+2}), \ldots, h(s_{i+w})$,

4.) $RNS \rightarrow N_a : d(N_a, RNS), sign_{RNS}(d(N_a, RNS))$.

Observația 4. *RNS* poate face broadcast la nonce-uri, în loc să aștepte după cereri de la N_a. Dar aceasta va conduce la trafic inutil în rețea.

Observația 5. În loc să transmită semnătura digitală $Sign_{RNS}(d(N_a,RNS))$, solicitantă computațional, *RNS* poate folosi un cod MAC, mai puțin intensiv, ce poate folosi un nonce s_j pe post de cheie, acesta urmând a fi transmis ulterior într-o manieră similară protocolului TESLA [82].

Observația 6. Din motive de eficiență computațională RNS-ul poate stoca hash-urile ultimelor nonce-uri (trebuiesc stocate un număr de nonce-uri corespunzător distanței maxime în rețea).

Observația 7. Pentru a-și determina poziția N_a trebuie să obțină de la *PLS* poziția *P(RNS)* și eroarea de localizare e_{RNS} a lui *RNS*. După care va rula algoritmul Vivaldi prezentat mai sus.

Observația 8. Dacă *RNS* îi transmite lui N_a cel mai recent nonce al său s_i, localizarea ar fi mai puțin precisă deoarece s_i a fost deja trimis, și este posibil ca N_a să-l mai fi primit încă odată (ca urmare a unei cereri paralele de localizare). Astfel, RTT-ul poate avea valori în următorul interval:

$$RTT_{s_i} \in \left[\frac{t_{snd}(s_i) - t_{rec}(h(s_i))}{2}, \frac{t_{gen}(s_i) - t_{rec}(h(s_i))}{2} \right).$$

RTT-ul poate fi exprimat cu o precizie de maxim δ (timpul dintre generarea a două nonce-uri consecutive) ce reprezintă lungimea maximă a intervalului. În relația de mai sus $t_{gen}(s_i)$ reprezintă momentul de timp la care nonce-ul s_i a fost generat, iar $t_{snd}(s_i)$ momentul la care a fost transmis; $t_{rec}(h(s_i))$ reprezintă momentul de timp la care hash-ul nonce-ului a fost recepționat de la N_a. Astfel, $RTT=|P(RNS)-P(N_a)|+\varepsilon$, $\varepsilon \in [0,\delta)$. Deoarece pentru o aproximare cât mai precisă a RTT-ului: $\varepsilon \ll RTT_{\min}$, trebuie ca și $\delta \ll RTT_{\min}$ (RTT_{min} reprezintă cel mai mic RTT tolerat în rețea).

Observația 9. Dacă un nod N_a vrea să mintă în legătură cu poziția sa, el poate pretinde că e mai departe, nu și mai aproape, de un *RNS.* Această situație este prezentată în Figura 27. Dacă N_a pretinde că se află în poziția N'_a cu $d'<d$, atunci *RNS* trebuie să primească nonce-ul s_i în momentul în care $s_{i+2d'}$ este generat. Aceasta înseamnă ca N_a va trebui să trimită s_i la momentul $i+2d'-d$, dar N_a nu se află în posesia lui s_i la acel moment de timp, acesta va sosi mai târziu, la momentul $i+d$ (evident $i+d>i+2d'-d$). Datorită condiției din definiția nonce-urilor ritmice, N_a nu îl poate genera pe s_i mai devreme.

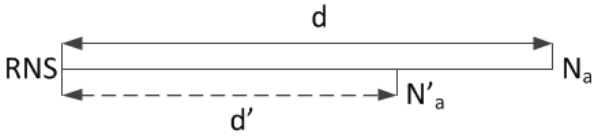


Figura 27 Nodul N_a pretinde sa fie mai aproape (N'_a) de *RNS* [21]

4.3.3.2. Localizare sigură folosind triangulare

Considerând cazul în care un nod N_b dorește să verifice poziția nodului N_a. Pentru aceasta N_b accesează *PLS* pentru a găsi trei RNS-uri (RNS_1, RNS_2, RNS_3) ce poziționează N_a în interiorul triunghiului definit de ele. Apoi N_b îi va cere lui N_a să-și verifice poziția în raport cu cele trei RNS-uri. După cum se poate vedea în Figura 28 dacă N_a pretinde a fi mai departe de unul din cele trei RNS-uri, el va fi automat mai aproape de cel puțin unul din celelalte două RNS-uri. Conform observației 9 acest lucru nu se poate întâmpla.

Arcele din Figura 28 sugerează că nodul N_a poate fi, din punctul de vedere a RNS-ului, în interiorul cercului definit de *RNS* (centrul) și RTT-ul estimat (raza). După verificarea poziției lui N_a, cele trei RNS-uri vor comunica rezultatul verificării ($pos\_valid_{N_a}$) către *PLS.* Ulterior, N_b poate cere rezultatul verificării PLS-ului.

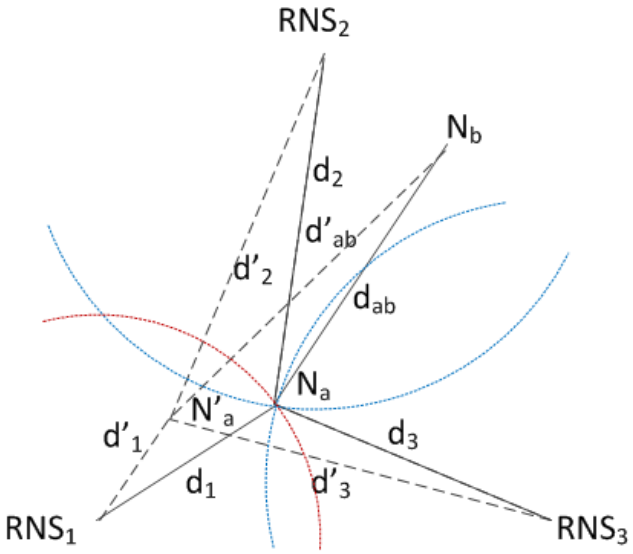


Figura 28 Verificarea locației lui N_a folosind triangulare

4.3.4. Sporirea rezistenței protocolului SSL/TLS la atacuri DoS

După cum s-a văzut anterior [27], [23] protocolul SSL/TLS poate fi protejat de atacuri DoS prin utilizarea puzzle-lor criptografice, dar aceste soluții tratează toți clienții în mod uniform. Adică, în momentul în care un server SSL se află sub atac va cere puzzle-uri, de o dificultate proporțională cu dimensiunea atacului, tuturor clienților (legitimi și nelegitimi). Dificultatea fiind aceeași pentru toți clienții. Pentru a rafina aceste soluții în [21] se prezintă o metodă de determinate a dificultății puzzle-urilor în funcție de locația clientului și de numărul de cereri ce vin de la acel client. Handshake-ul SSL/TLS în acest caz este prezentat în Figura 29.

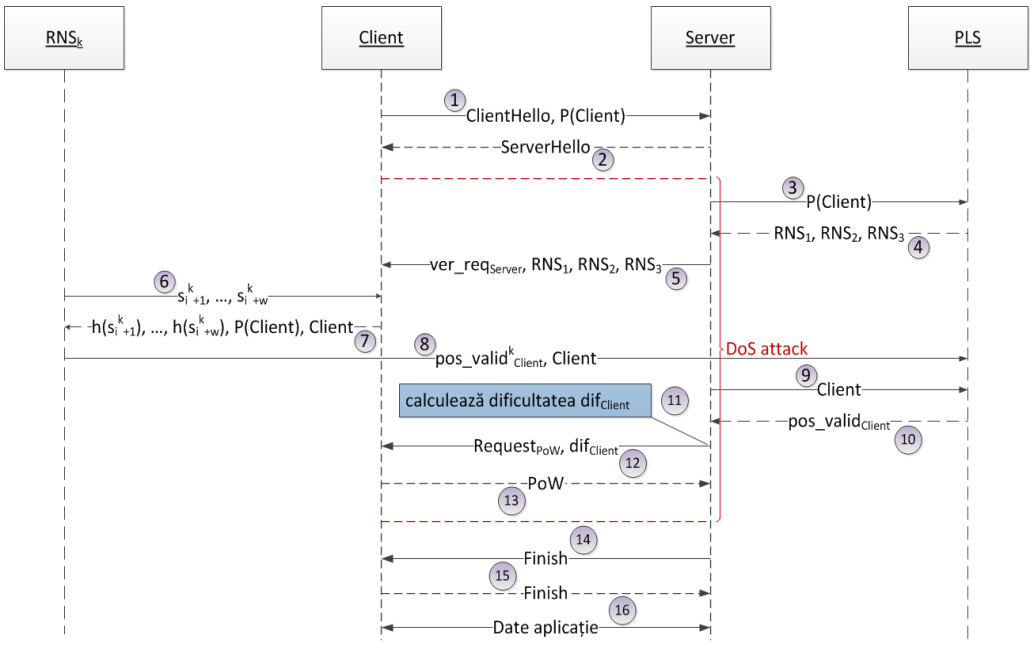


Figura 29 Utilizarea localizării sigure pentru protejarea SSL/TLS în fața atacurilor DoS

Astfel, atunci când un client dorește să se conecteze la un server SSL/TLS, acesta va transmite alături de *ClientHello* și poziția sa *P(Client)* (pasul1). După care serverul răspunde cu *ServerHello* (pasul 2), iar dacă server-ul este foarte încărcat va trimite la *PLS* poziția clientului (pasul 3) pentru a primi de la acesta (pasul 4) o listă de trei RNS-uri (*RNS$_1$, RNS$_2$, RNS$_3$*) ce pot triangula clientul. Serverul va trimite mai departe această listă la client (pasul 5) pentru ca acesta să-și demonstreze locația folosind algoritmul Vivaldi modificat (pașii 6, 7). RNS-urile vor verifica poziția clientului și vor notifica PLS-ul despre corectitudinea acesteia (pasul 8). Serverul SSL/TLS va întreba PLS-ul despre corectitudinea poziției clientului (pasul 9), după primirea răspunsului (pasul 10), serverul va determina dificultatea puzzle-ului. Dacă clientul minte, în legătură cu poziția sa, serverul fie va genera un puzzle de dificultate maximă, fie va bloca accesul clientului la server; în caz contrar, dificultatea puzzle-ului va fi calculată conform formulei:

$$dif_{Client} = \sum_{i=1}^{\log_2 dif_{default}} |Z_i| * \frac{dif_{default}}{2^i}.$$

În relația de mai sus $dif_{default}$ reprezintă nivelul implicit de securitate definit pe server (valorile definite în [27] pot fi folosite pentru acesta), $|Z_i|$ reprezintă numărul de cereri ce provin dintr-o zonă Z_i, unde $Z_i = \{n \mid i-1 \le d(n, Client) < i\}$. Fiecare zonă influențează dificultatea puzzle-ului prin numărul de cereri ce vin din

zona respectivă și prin apropierea față de centrul zonei, după cum se poate vedea în Figura 30, în care cercurile reprezintă zone de o anumită dificultate (dificultatea descrește odată ce ne îndepărtăm de centru – zona de origine a DoS-ului), iar numerele prezente în dreptul nodurilor reprezintă numărul de vecini descoperiși. Pentru a calcula dificultatea puzzle-ului, serverul trebui să mențină o listă cu cele mai recente locații din care a primit cereri.

Odată determinată dificultatea, serverul va trimite puzzle-ul spre rezolvare clientului (pasul 12). După ce rezolvă puzzle-ul clientul trimite soluția către server (pasul 13), iar dacă soluția este corectă protocolul continuă în mod obișnuit (pașii 14, 15, 16).

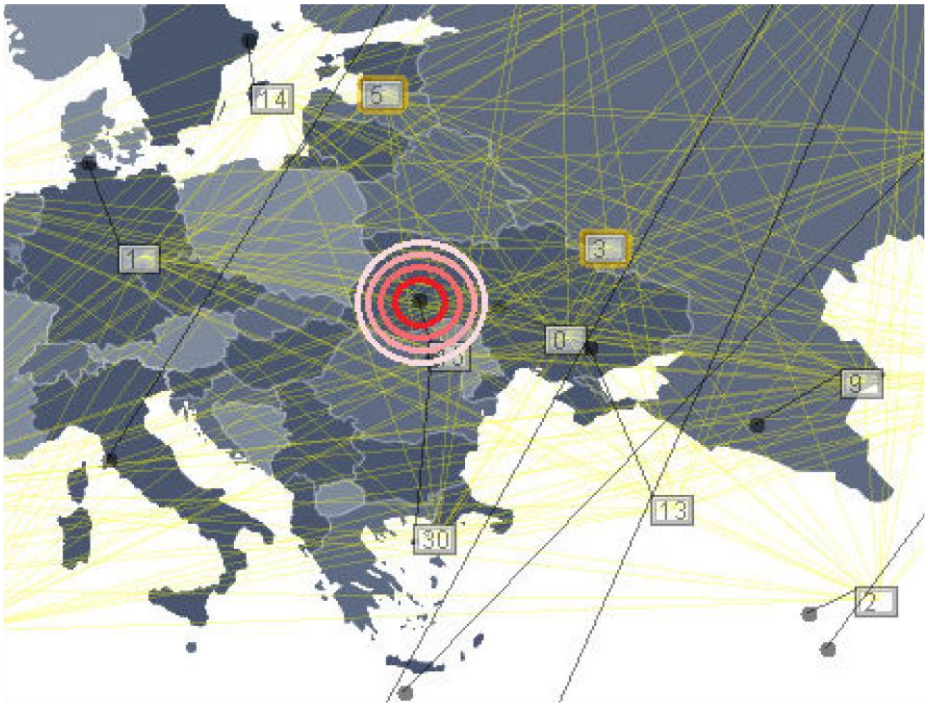


Figura 30 Ajustarea dificultății puzzle-urilor (figură generată de autor cu ajutorul VivaldiMonitor – www.dinesgroup.org)

Totuși noțiunea de zonă are sens doar luând în considerare amplasarea nodurilor în sistemul de coordonate virtual și nu în amplasarea fizică pe glob a nodurilor. Pentru a observa distribuția nodurilor (Figura 31) în sistemul de coordonate Vivaldi s-a realizat un experiment la care au luat parte 20 noduri, distribuite astfel:

- 6 noduri în comuna Bozovici – Caraș-Severin;
- 13 noduri în municipiul Timișoara – Timiș;

- 1 nod în municipiul Zalău – Sălaj.

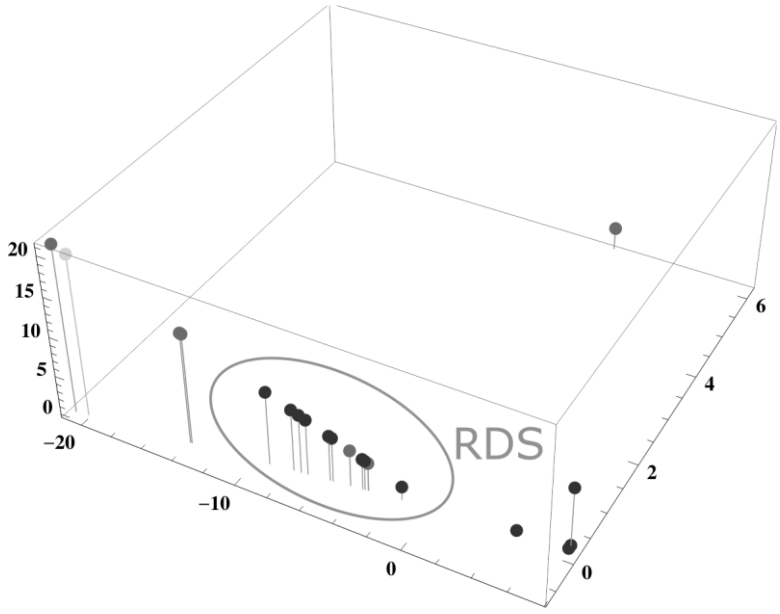


Figura 31 Distribuția nodurilor în sistemul de coordonate Vivaldi

Dintre nodurile alese 11 (2 din Bozovici și 9 din Timișoara) au făcut parte din rețeaua furnizorului de servicii Internet RDS, acestea apărând mai aproape în sistemul de coordonate virtual. De altfel, chiar dacă nodurile sunt foarte apropiate fizic sau chiar dacă sunt parte a aceleași rețea (având valorile RTT-ului apropiate) ele vor ocupa o poziție distinctă în sistemul de coordonate virtual, acest lucru permițând identificarea clară a nodurilor.

Astfel, chiar dacă un nod onest va fi apropiat fizic de un nod malițios, dacă acesta nu va face parte din rețeaua atacatorului, nu va fi afectat de dificultatea puzzle-urilor. Pe de altă parte, dacă nodul onest se nimerește în rețeaua atacatorului, acesta nu va fi afectat de rezolvarea unui număr redus de puzzle-uri, rezolvarea unui puzzle fiind de ordinul 10^{-6}s (Tabelul 3).

4.3.5. Localizarea sursei email-urilor

Localizarea sursei email-urilor poate fi realizată introducând câteva modificări în protocolul SMTP (Figura 32). Când un nod sau un client N_a dorește să trimită un email, el trebuie să adauge și poziția s-a la conținutul email-ului (pasul 1). Când recepționează email-ul, serverul de email (*MS*) va cere (pasul 2) la *PLS* trei RNS-uri (RNS_1, RNS_2, RNS_3) ce pot triangula clientul. După ce primește (pasul 3) cele trei RNS-uri, *MS* îi va cere (pasul 4) lui N_a să-și verifice poziția în raport cu

acestea (pașii $5^k - 6^k, k = \overline{1,3}$). În cele din urmă *MS* va cere la *PLS* rezultatul verificării (pașii 8,9). Dacă N_a minte în legătură cu poziția sa, email-ul său va fi marcat automat de către *MS* ca fiind spam. În caz contrar, dacă nodul provine dintr-o locație cunoscută ca sursă de spam, email-ul său va fi marcat ca spam, sau i se poate cere să rezolve un puzzle, având dificultatea calculată ca în cazul SSL/TLS.

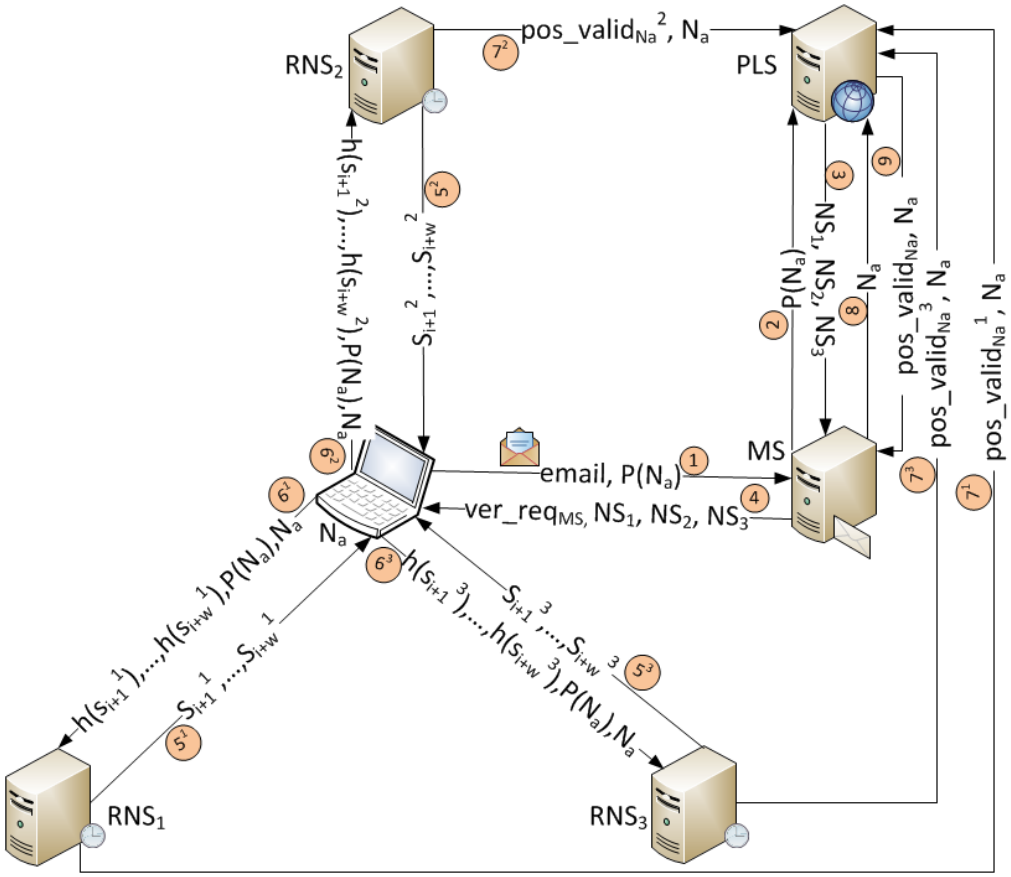


Figura 32 Folosirea algoritmului Vivaldi modificat pentru identificarea sursei email-urilor [21]

Protocolul SMTP modificat este prezentat mai jos:

$1.N_a \rightarrow MS : email, P(N_a);$

$2.MS \rightarrow PLS : P(N_a);$

$3.PLS \rightarrow MS : RNS_1, RNS_2, RNS_3;$

$4.MS \rightarrow N_a : ver\_req_{MS}, RNS_1, RNS_2, RNS_3;$

$5^{k=\overline{1,3}}.RNS_k \rightsquigarrow N_a : s_{i^k+1}, \ldots, s_{i^k+w};$

$6^{k=\overline{1,3}}.N_a \rightsquigarrow RNS_k : h(s_{i^k+1}),...,h(s_{i^k+w}),P(N_a),N_a;$

$7^{k=\overline{1,3}}.RNS_k \rightarrow PLS : pos\_valid^k{}_{N_a},N_a;$

$8.MS \rightarrow PLS : N_a;$

$9.PLS \rightarrow MS : pos\_valid_{N_a},N_a.$

4.4. Amprentarea echipamentelor în rețea folosind timestamp-uri ICMP

Identificarea nodurilor sursă a atacurilor DoS folosind localizarea poate fi îmbunătățită folosind amprentarea, astfel se identifică echipamentul care contribuie la un atac DoS și nu zonele cauzatoare de DoS. Astfel, în [18], se utilizează metoda de amprentare folosită de Kohno, Broido și Claffy [60], dar amprentarea este bazată pe timestamp-urile prezente în pachetele ICMP. Chiar dacă lucrarea utilizează telefoane mobile pentru partea experimentală, acestea nefiind (în general) o sursă de DoS, rezultatele sunt valabile pentru orice tip de nod dintr-o rețea. Telefoanele mobile inteligente de azi fiind, de fapt, mini calculatoare ce dispun de un sistem de operare (Android - http://www.android.com/, un sistem de operare bazat pe Linux și dezvoltat de Google). În plus, rezultatele obținute pe telefoane mobile ne-au permis să observăm că toate telefoanele inteligente actuale pun în pericol intimitatea utilizatorilor și să dezvoltăm o metodă de identificare fizică a telefoanelor mobile foarte folositoare în cazul furtului acestora.

4.4.1. Pachetele ICMP

Pentru a putea realiza amprentarea, identificatorul *C* (partea care se ocupă de amprentare) trebuie să fie capabil să emită cereri ICMP de tip timestamp (ICMP tip 13) către dispozitivul ce se dorește amprentat *R* (partea care răspunde la cererile ICMP), acesta va oferi un răspuns ICMP de tip timestamp (ICMP tip 14). Structura unui pachet ICMP este prezentată în Figura 33. Dintre câmpurile prezente în acest pachet ICMP, cele care prezintă interes pentru amprentare sunt: *tip,* care va avea valoare 13 pentru cerere și valoarea 14 pentru răspuns; *timestamp-ul originar* care este setat de *C* chiar înainte de transmiterea cereri (T_C); *timestamp-ul recepționării*, care este setat de *R* chiar în momentul primirii cererii (T_R); *timestamp-ul de tranzit* care este setat de *R* chiar înainte de transmiterea răspunsului. Timestamp-urile prezente în pachetele ICMP reprezintă milisecundele scurse de la miezul nopții și sunt reprezentate pe 32 biți.

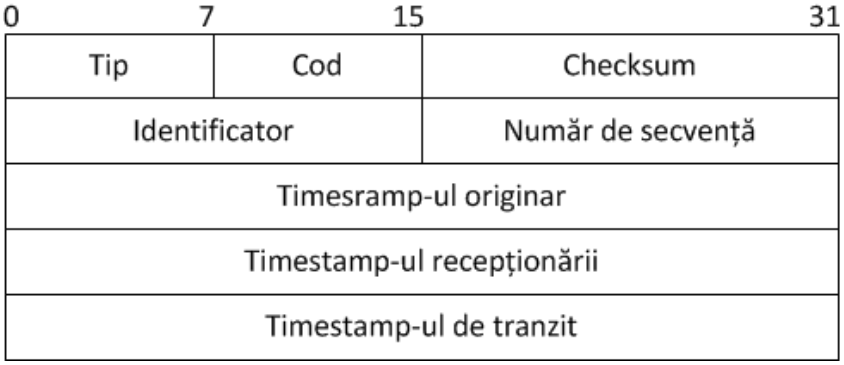


Figura 33 Structura pachetului ICMP

Pentru mesajul *i*, având T_C^i și T_R^i, notăm $x^i=T_C^i-T_C^0$ timpul dintre mesajul *i* și primul mesaj și $y^i=T_R^i-T_R^0-(T_C^i-T_C^0)$ diferența observată dintre cele două ceasuri. Astfel, pentru un echipament *e* ce primește *n* cereri ICMP de tip timestamp notăm $O_e=\left\{\left(x^i,y^i\right),i=\overline{0,n}\right\}$ deviația ceasului lui *e*, iar cu τ_e alunecarea ceasului (skew), aceasta fiind pantă punctelor din O_e. Dacă ceasul lui e ar fi perfect (τ_e=0) atunci O_e ar fi, de fapt, o bandă orizontală.

4.4.2. Determinarea alunecării

Calculul alunecării este bazat pe metoda prezentată în [70]. Pentru aceasta este necesară identificarea dreptei *ax+b* care mărginește inferior O_e (vezi Figura 34 b)). Astfel, $y^i-\left(ax^i+b\right)\geq 0, i=\overline{0,n}$. Pe de altă parte această dreaptă trebuie să fie cât mai apropiată de O_e, adică suma distanțelor de la dreaptă la toate punctele din O_e trebuie să fie minimă: $\min\left\{\sum_{i=0}^{n}y^i-(ax^i+b)\right\}=\min\left\{\sum_{i=0}^{n}y^i-a\sum_{i=0}^{n}x^i-nb\right\}$. Utilizând aceste relații problema determinării alunecării devine o problemă de programare liniară unde inegalitățile lineare reprezintă constrângerile problemei, iar criteriul de minim reprezintă funcția sa obiectiv.

Aceleași rezultate s-ar obține și dacă s-ar folosi regresia liniară pe segmente mici ale setului de date, dar regresia liniară nu are toleranță la zgomot. Deoarece zgomotul este prezent în măsurători (Figura 34 b)) datele ar trebui filtrate mai întâi. Metoda bazată pe programare liniară nu este afectată de zgomot deoarece acesta nu poate fi prezent sub punctele din O_e.

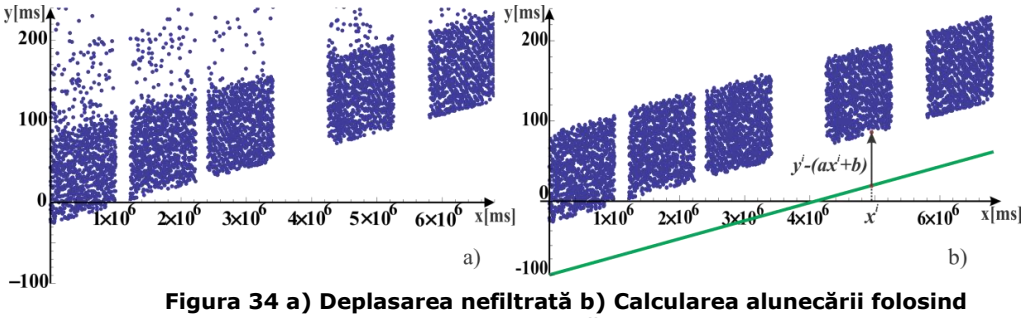


Figura 34 a) Deplasarea nefiltrată b) Calcularea alunecării folosind programare liniară [18]

4.4.3. Rezultate experimentale

Pentru testare s-au amprentat 2 routere wireless și 5 telefoane mobile inteligente. Routerele wireless folosite au fost: D-Link DIR-825 și Linksys WRT54GL. Firmware-ul original a fost înlocuit cu sistemul de operare OpenWrt Backfire (https://openwrt.org), o distribuție de Linux dedicată echipamentelor înglobate (embeded). Telefoanele mobile folosite au fost toate telefoane cu sistem de operare Android, acestea fiind: un Samsung Google Nexus S cu sistem de operare Android Jelly Bean 4.1.2, două Samsung I9000 Galaxy S, un Motorola RAZR XT912 cu sistem de operare Android Gingerbread 2.3.6 și un Samsung Galaxy Mini cu sistem de operare Android Gingerbread 2.3.5.

Astfel, pentru colecția pachetelor ICMP timestamp s-a folosit utilitarul *ping,* disponibil în toate distribuțiile de Linux:

```
ping -i .1 -T tsonly -c 10000 192.168.1.1 >> /tmp/ping.log.
```

Parametrii ce apar în comandă sunt: *-i* care specifică durata (în secunde) dintre două cereri ICMP consecutive; *-T* setează tipul cereri ICMP (13 în cazul nostru); *-c* specifică numărul de cereri transmise. Cu ajutorul comenzii de mai sus se pot captura 10000 de pachete ICMP timestamp în 16.6 minute.

Odată capturate timestamp-urile acestea vor fi analizate în *Mathematica.* Fragmentul de cod folosit pentru rezolvarea problemei de programare liniară, utilizată în calculul alunecării este prezentat în Secvența 3.

```
      offset_e (*deviația echipamentului*)
      x=offset_e[[All,1]] (*timpii cererilor ICMP relative la
prima cerere*)
      y=offset_e[[All,2]] (*deviația observată de emițător*)
      n=Length[offset_e] (*lungimea măsurătorii*)

      constr=y[[1]]-a*x[[1]]-b>=0(*inițializează constrângerile*)
      For[i=2,i<=n,i=i+1,constr=constr&&y[[i]]-a*x[[i]]
         -b>=0](*construiește constrângerile*)

      sumx=Total[x] (*calculează suma timpilor relativi pentru
cererile ICMP*)
      sumy=Total[y] (*calculează suma deviațiilor*)

      NMinimize[{sumy-a*sumx-n*b,constr},{a,b}]
         (*rezolvă problema de programare liniară – b reprezintă
alunecarea ceasului echipamentului*)
```

Secvența 3 Calcularea alunecării prin programare liniară folosind Mathematica

Alunecarea în cazul routerelor wireless este prezentată în Figura 35 iar pentru telefoanele mobile în Figura 36.

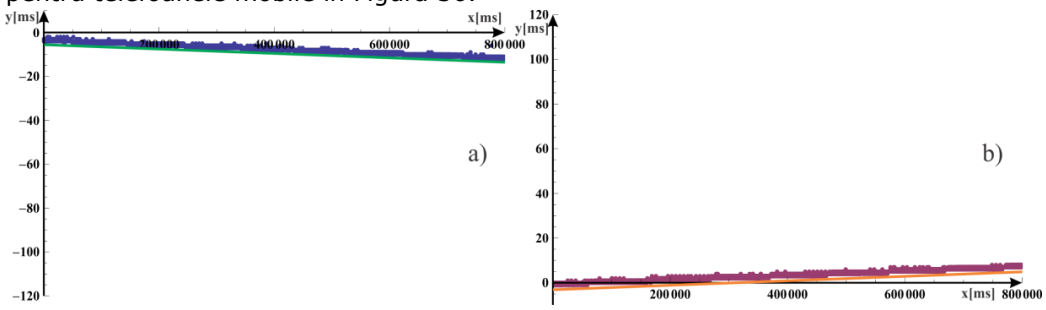


Figura 35 Deplasarea și alunecarea pentru routerele wireless: a) D-Link DIR-825; b) Linksys WRT54GL

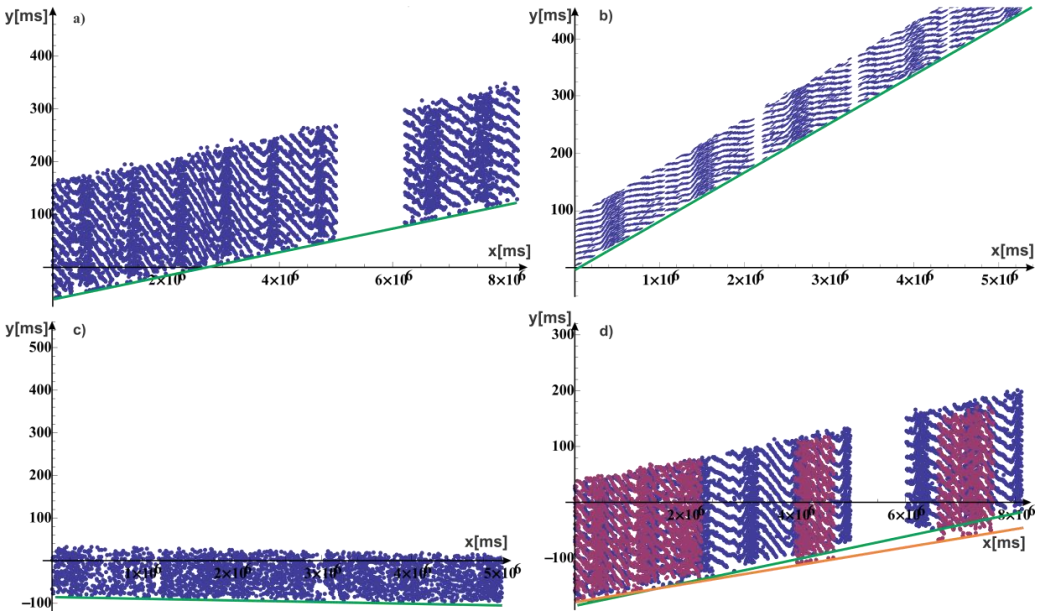


Figura 36 Alunecarea pentru: a) Samsung Google Nexus S; b) Motorola RAZR XT912; c) Samsung Galaxy Mini S5570; d) Samsung I9000 Galaxy S [18]

După cum se poate observa diferențele în ceea ce privește valoarea alunecării sunt semnificative și evidente chiar și în cazul a două echipamente identice (Figura 36 d)). Alunecarea rămâne constantă chiar dacă măsurătorile sunt întrerupte (Figura 37 a)), echipamentul folosește Network Time Protocol (ntp) pentru sincronizare (Figura 37 b)) și chiar dacă amprentarea se face prin intermediul a mai multor noduri (hops) din rețea (Figura 37 c)). Astfel, această metodă se poate dovedi extrem de eficientă pentru detecția adversarilor DoS.

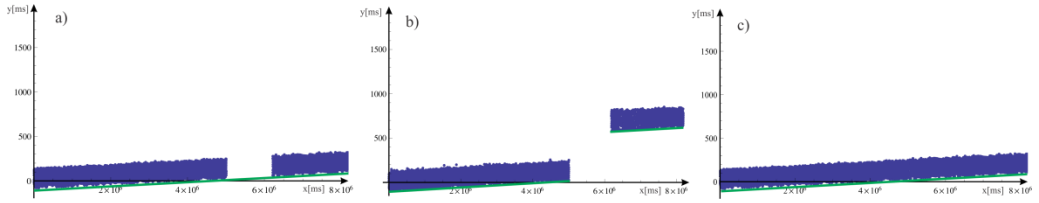


Figura 37 Alunecarea în cazul: a) cu întreruperea măsurătorilor; b) cu sincronizare; c) prin mai multe hopuri

În Tabelul 6 sunt prezentate valorile alunecării, măsurate în ppm (*μs/s*) pentru echipamentele analizate.

Tipul echipamentului	Nr. de timestamp-uri	Durata capturii	Alunecarea
D-Link DIR-825	8000	13.33 min	-10.09 ppm
Linksys WRT54GL	8000	13.33 min	10.62 ppm
Nexus S	7000	11.66 min	22.68 ppm
Motorola RAZR	5000	8.33 min	87.43 ppm
Galaxy Mini	4849	8.08 min	- 3.17 ppm
Galaxy S 1	7097	11.82 min	20.83 ppm
Galaxy S 2	3997	6.66 min	16.26 ppm

Tabelul 6 Valoarea alunecării pentru echipamentele folosite în experiment

4.4.4. Utilizarea amprentării pentru prevenirea furtului soluțiilor PoW

Toate metodele de protecție bazate pe PoW au problema că un adversar, amplasat între entitatea care generează puzzle-ul și cea care îl rezolvă, poate fura soluția de la rezolvitor. Pentru a preveni această situație, trebuie ca soluția puzzle-ului să fie legată de entitatea care îl rezolvă. În acest scop se poate utiliza amprentarea, astfel PoW-ul oferit de rezolvitor va conține pe lângă soluția puzzle-ului și alunecarea echipamentului rezolvitorului. Pentru a realiza asocierea dintre soluția puzzle-ului și amprenta echipamentului se poate utiliza o funcție hash: $\mathrm{PoW\_sol} = h(puzzle\_sol, clk\_skw_e)$, unde *puzzle_sol* este soluția puzzle-ului, iar clk_skw_e reprezintă alunecarea echipamentului care a rezolvat puzzle-ul. Protocoalele bazate pe PoW prezentate în capitolele anterioare rămân valabile în continuare, doar modul de generare și verificare a soluției se modifică prin introducerea alunecării în soluția PoW-ului.

4.4.5. Implicații specifice telefoanelor mobile

Deoarece pentru partea experimentală au fost folosite telefoane mobile inteligente s-au descoperit unele implicații cu privire la securitatea acestora. Pe deoparte amprentarea telefoanelor mobile poate fi folosită pentru marketing sau pentru identificarea telefoanelor furate, pe de altă parte amprentarea pune în pericol intimitatea utilizatorilor.

4.4.5.1. Utilizarea amprentării pentru urmărirea utilizatorilor

Există un risc evident ca această metodă de amprentare să poată fi folosită pentru urmărirea utilizatorilor. Deoarece nu există nici o posibilitate de a dezactiva răspunsurile ICMP timestamp și nici o modalitate de a avertiza utilizatorul că este urmărit, această metodă de amprentare reprezintă o amenințare și mai serioasă la adresa utilizatorilor de telefoane mobile inteligente.

Pentru a evidenția pericolul adus de amprentarea telefoanelor mobile asupra intimității utilizatorilor s-a realizat un scenariu experimental pentru care s-au utilizat 3 puncte de acces distincte și 5 telefoane mobile inteligente. Punctele de acces au fost montate pe 3 etaje diferite ale Universității ”Politehnica” din Timișoara iar telefoanele mobile au fost distribuite la 5 studenți. Participanții la experiment trebuiau să se conecteze în mod aleatoriu la cele 3 puncte de acces și să utilizeze Internetul pentru 15 min. După experiment, participanți au confirmat că nu au observat latențe în conexiunea la internet sau altceva suspicios. Rezultatele experimentului (deviația ceasurilor) sunt prezentate în Figura 38.

După cum se poate vedea, indiferent de punctul de acces, ceasul unui telefon mobil va avea aceeași alunecare, permițând astfel urmărirea lui.

Totuși problema urmării utilizatorilor ar putea fi evitată fie prin dezactivarea răspunsurilor ICMP de tip timestamp fie prin manipularea pantei deviației ceasului. Fiind un sistem de operare bazat pe Linux, Android poate beneficia de puternicul firewall disponibil pe acesta: *iptables*.

Pentru a bloca cererile ICMP de timp timestamp, următoarele reguli trebuie adăugate la iptables:

```
iptables -I INPUT 1 -p ICMP --icmp-type timestamp-request
-j DROP
iptables -I INPUT 1 -p ICMP --icmp-type timestamp-reply -j
DROP
```

De asemenea, iptables poate fi folosit pentru logarea cererilor ICMP timestamp primite, astfel încât o aplicație să poată citi aceste loguri și să notifice utilizatorul despre faptul că poate fi victima unei amprentări. Utilizatorul putând să permită cererile sau să le blocheze.

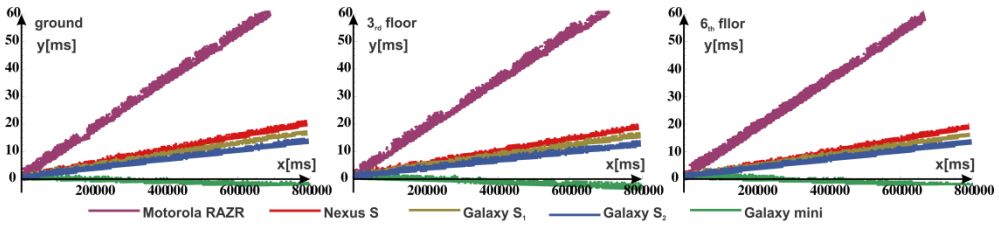


Figura 38 Deplasarea înregistrată pe cele 3 puncte de acces montate în Universitatea ”Politehnica” Timișoara [18]

Dacă nu se dorește dezactivarea totală a cererilor ICMP de tip timestamp (de exemplu utilizatorul dorește să fie identificat doar într-un anumit context pentru o perioadă redusă de timp) se poate apela la manipularea pantei deviației ceasului, aceasta se poate realiza modificând driverul echipamentului wireless integrat în

telefonul mobil. Pentru aceasta fiind necesară modificarea funcției *icmp_input* din *ip_icmp.c:*

```
void icmp_input(struct mbuf *m, int off){
      ...
      case ICMP_TSTAMP:
      ...
      if (iptime() % 360000 == 0){
            /* determină o pantă nouă odată pe oră */
            slope = (float)rand()/(float)RAND_MAX;
      }
      icp->icmp_type = ICMP_TSTAMPREPLY;
      icp->icmp_rtime = slope * iptime();
      ...
      }
```

Secvența 4 Schimbarea alunecării prin modificarea driver-ului plăcii de rețea

4.4.5.2. Utilizarea amprentării ca tehnică anti-furt

Până în prezent majoritatea tehnicilor anti-furt sunt bazate pe coduri IMEI (International Mobile Station Equipment Identity), aceste coduri fiind dificil de falsificat, iar falsificarea lor este considerată infracțiune. Cu toate acestea, codurile IMEI sunt încă falsificate; pentru a le îmbunătăți securitatea s-a propus utilizarea de coduri IMEI nefalsificabile fizic (NF-IMEI). Construcția acestor coduri este bazată pe combinarea codului IMEI a telefonului cu amprenta sa fizică dată de alunecarea ceasului său. În plus, acest NF-IMEI este *verificabil public*, în sensul că orice entitate poate verifica dacă acest NF-IMEI este autentic sau nu. Figura 39 arată că alunecarea mobilului τ_e este semnată digital împreună cu IMEI-ul standard folosind cheia producătorului (Prd.). Concatenarea IMEI-ului cu semnătura digitală formează NF-IMEI-ul:

$$\text{NF-IMEI} = IMEI \,||\, Sig_{\text{Prd}}(IMEI \,||\, \tau_e).$$

Astfel, folosind certificatul digital al producătorului se poate verifica autenticitatea NF-IMEI-ului.

Calcularea corectă a alunecării poate fi realizată folosind un mobil cu un sistem de operare curat (pentru a evita modificarea driver-elor, după cum s-a arătat în secțiunea precedentă), sau se poate utiliza un driver semnat digital, în momentul verificării.

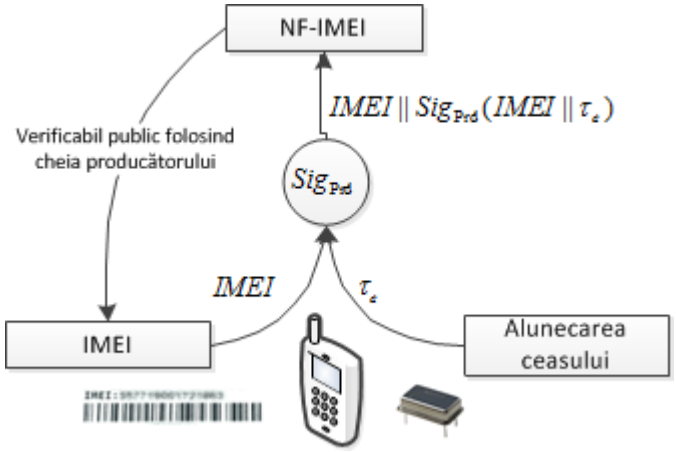


Figura 39 Procedură pentru generarea NF-IMEI-ului

Concatenarea semnăturii cu IMEI-ul duce la o dimensiune mai mare a NF-IMEI-ului (IMEI-ul original are 15 caractere). Totuși, creșterea în dimensiune a IMEI-ului nu ar trebuie să fie problematică, deoarece acest IMEI nu este o parolă ce trebuie reținută de utilizatori. Pe de altă parte, există metode [6] ce pot reduce dimensiunea acestei semnături la doar 160 biți (dacă acest NF-IMEI trebuie imprimat, ar putea fi reprezentat folosind aproximativ 25 caractere, cheia de activare a Windows 7 având cam aceeași dimensiune). Dacă, totuși, dimensiunea reprezintă o problemă, singura alternativă rămâne criptografia simetrică. Astfel, se poate reduce dimensiunea NF-IMEI-ului la 128 biți (valoarea minimă pentru AES) sau chiar mai mică dacă e necesar. Dar, folosind criptografia simetrică, NF-IMEI-ul nu mai poate fi verificabil public, ci doar de producătorul care deține cheia. Aceste verificări pot fi realizate periodic, de exemplu în momentul actualizării sistemului de operare.

5. Concluzii

Atacurile DoS/DDoS reprezintă o problemă pregnantă în securitatea informației. Aceste atacuri se realizează apelând la metode relativ simple de atac (flood, spam, etc) și vizează diferite domenii (politic, financiar, etc.), cauzând pagube însemnate victimelor vizate. În domeniul financiar atacurile DoS dacă sunt combinate cu atacuri de tip inginerie socială pot avea un efect și mai puternic. În domeniul sistemelor de conducere automată, posibilitatea realizării unui atac DoS reprezintă un motiv serios de îngrijorare deoarece pagubele ce pot fi cauzate de un astfel de atac sunt comparabile cu cele ale unui atac armat. De aceea identificarea corectă a surselor de atac DoS și contracararea lor trebuie să fie o prioritate în domeniul securității informatice.

Cercetarea din această teză pornește de la analiza rezistenței la atacuri DoS a echipamentelor profesionale folosite în sistemele moderne de conducere și control. Astfel, studiul nostru arată că și produsele proiectate cu atenție cum sunt echipamentele SCALANCE au probleme, fie datorită implementării (problema de la interfața de autentificare pentru aplicația de configurare a echipamentului) sau fie le-au moștenit de la protocoalele pe care le folosesc (faptul că atacul de de-autentificare este posibil). În cazul echipamentelor SCALANCE problemele ce afectează securitatea protocoalelor de comunicare pot deveni fatale în cazul unui scenariu de control la distanță, după cum se poate vedea în studiul de caz în care un adversar poate ”tăia” canalul de comunicație iar procesul condus iese din regimul normal de funcționare. Pentru a rezolva astfel de probleme este necesar ca inginerii sistemelor de control să colaboreze cu experți în domeniul securității care să cunoască capabilitățile atacatorilor. În general, echipamentele wireless nu pot garanta o comunicație continuă, de aceea este necesară prezența unui controler redundant local care să funcționeze în cazul în care apar probleme pe canalul de comunicație; totuși această soluție va duce la creșterea costului sistemului de comandă și control. Astfel, în aplicațiile critice, cum sunt sistemele de control, echipamentele wireless ar trebui folosite doar acolo unde folosirea unei infrastructuri cablate nu este fezabilă.

Atacurile de tip DoS sunt posibile, după cum s-a văzut, datorită faptului că nu există un echilibru între resursele computaționale necesare unui server pentru a răspunde unei cereri, pe de o parte, și resursele necesare adversarului pentru a realiza atacul, pe de altă parte. Pentru a contrabalansa acest dezechilibru există binecunoscutul mecanism proof-of-work. Deoarece majoritatea atacurilor sunt posibile datorită vulnerabilităților prezente în protocoale sau a aplicațiilor construite pe baza acestor protocoale, lucrarea analizează mecanisme de protecție care au la bază tehnici proof-of-work. Astfel, s-a realizat o extensie a OpenSSL ce oferă suport pentru puzzle-uri criptografice și care este compatibilă cu RFC [5] și RFC [88]. Aceasta s-a folosit cu succes pentru protejarea unui server web.

O altă formă de DoS prin epuizarea resurselor serviciilor de email o reprezintă spam-ul. Și împotriva spam-ului s-au folosi puzzle-uri criptografice de tip

time-lock. Soluția propusă fiind compatibilă cu infrastructura existentă, fiind necesare doar câteva modificări: un server de ticketing pentru distribuirea informației necesare creări și rezolvării corecte a puzzle-lor. Măsurătorile efectuate demonstrează că introducerea puzzle-lor conduce la reducerea profitului pentru spamer, iar impactul puzzle-lor asupra clienților legitimi este insesizabil.

Deoarece spam-ul contribuie și la frauda financiară, prin intermediul ingineriei sociale și deoarece România este una dintre principalele surse ale acestui tip de fraudă pe de o parte și o potențială țintă pe de altă parte; o aplicație relevantă este analiza rezistenței la atacuri mixte DoS și inginerie social a băncilor din România și a gradului de conștientizare, în rândul clienților, al pericolului reprezentat de ingineria socială. Analiza realizată arată că soluțiile folosite de furnizorii de servicii de e-banking din România sunt variate: de la simple parole alese de utilizatori, la unele mai avansate bazate pe token-uri sau pe certificate digitale. Aceste soluții fiind parțial sau total vulnerabile în fața unor atacuri de tip phishing, pharming sau MitB, atacuri ce pot fi și mai puternice în combinație cu atacurile de tip DoS. Pentru a îmbunătăți rezistența în fața unor astfel de atacuri, recomandăm folosirea soluțiilor ce oferă autentificare mutuală între client și site-ul de e-banking, dar aceste soluții nu trebuie să devină greu de folosit pentru utilizatori. Pentru realizarea acestui scop utilizarea de dispozitive portabile, cum ar fi smart-carduri sau telefoane mobile, poate reprezenta o soluție eficientă și ușor de implementat. În prezent, nici o soluție existentă pe piața românească nu este complet sigură în fața unor atacuri moderne, cum ar fi, de exemplu, atacurile de tip MitM.

Nu doar contracararea atacurilor DoS este importantă ci și identificarea sursei acestor atacuri este la fel de importantă, fie pentru a bloca direct sursele care produc DoS, fie în cazul DDoS pentru a îmbunătăți tehnicile existente, dar pentru aceasta e nevoie de o tehnică de localizare sigură. Astfel, algoritmul Vivaldi a fost modificat astfel încât să ofere localizare sigură în sistemul său de coordonate virtuale. De asemenea, deoarece puzzle-urile criptografice nu sunt foarte eficiente când sunt folosite ca unic mecanism de protecție în fața DoS, localizarea a fost folosită și pentru calibrarea dificultății puzzle-lor. Rezultatele experimentale arată că algoritmul Vivaldi modificat are un impact computațional mic asupra clienților ce îl folosesc, iar localizarea sigură poate fi folosită pentru adaptarea dificultății puzzle-lor, folosite în combaterea DoS sau a spam-ului, în funcție de locația atacatorilor.

Identificarea poate fi dusă la un alt nivel, folosind amprentarea fizică. În această teză am demonstrat eficiența acestei tehnici asupra echipamentelor mobile (routere, telefoane mobile inteligente, etc.). Rezultatele sunt în mod evident extensibile către orice echipament digital care deține ceas de timp real.

După cum s-a putut vedea atacurile DoS reprezintă o problemă prezentă în domenii de maximă importanță (reglare automată, e-banking) cu costuri enorme atât pentru furnizorii de servicii cât și pentru consumatorii acestora, dar acordând atenție proiectării protocoalelor, din fericire prin tehnicile menționate în lucrare (PoW, localizare și amprentare), efectul acestor atacuri poate fi redus. Chiar dacă soluțiile propuse în această lucrare sunt necesare, ele nu sunt suficiente pentru a

eradica un fenomen aflat în continuă evoluție. Mecanismele prezentate trebuie să se adapteze și să se dezvolte în funcție de nevoile utilizatorilor și de complexitatea atacurilor.

În încheiere vom prezenta contribuțiile aduse, concretizate sub formă de lucrări științifice, pe perioada stagiului de doctorat și concluziile finale ale cercetării prezentate în această lucrare.

1. **Identificarea vulnerabilității protocoalelor utilizate în practică.** Majoritatea atacurilor DoS sunt posibile, după cum se arată în lucrarea [43], datorită vulnerabilităților existente în protocoalele folosite de diverse aplicații. Studiile de caz pornesc de la identificarea unor astfel de probleme în protocolul SSL/TLS, în secțiunea 3.1 și în protocoalele de autentificare/autorizare folosite de serviciile de e-banking, în secțiunea 3.2. De asemenea, contramăsurile propuse pleacă tot de la vulnerabilitățile existente în diferite protocoale: SSL/TLS în secțiunile 4.1, 4.3 sau SMTP în secțiunile 4.2 și 4.3.
2. **Identificarea problemelor de securitate la echipamentele wireless industrial de tip SCALANCE.** Datorită problemelor de securitate de ultimă oră (virusul Stuxnet) din domeniul reglării automate și a implicațiilor acestea, identificarea problemelor prezente în astfel de echipamente este esențială pentru desfășurarea în siguranță a proceselor industriale complexe. În secțiunea 3.1 este realizată o analiză atât într-un scenariu static, de configurare, cât și într-unul dinamic, de comandă și control. Rezultatele analizei sunt prezentate în lucrarea [22].
3. **Analiza securității principalilor furnizori de servicii de e-banking de pe piața românească.** Datorită dezvoltării serviciilor de e-banking și a creșterii numărului de incidente în acest domeniu, analiza securității este relevantă, atât pentru furnizorii de servicii, care pot găsi diferite zone în care aceasta poate fi îmbunătățită, cât și pentru clienți, care pot descoperi diferitele metode prin care pot fi păcăliți de către infractorii cibernetici. Analiza, prezentă în secțiunea 3.2, este completată de un chestionar aplicat pe un număr de peste 100 participanți, acesta oferind informații despre conștientizarea pericolelor la care sunt expuși utilizatorii în domeniul e-banking, precum și despre preferințele lor în domeniul securității și uzabilității. Rezultatele analizei și ale chestionarului sunt prezentate în lucrarea [19].
4. **Protecția OpenSSL folosind puzzle-uri criptografice.** Majoritatea serverelor web folosesc SSL/TLS pentru a asigura autentificarea clienților sau confidențialitatea datelor, dar datorită efortului computațional indus de folosirea acestui protocol, atacatorii pot realiza, cu ușurință, atacuri de tip DoS asupra serverelor web prin cereri repetate. Pentru a contrabalansa această diferență de efort computațional în secțiunea 4.1 s-a implementat și utilizat cu succes un mecanism bazat pe puzzle-uri criptografice pentru protecția serverelor

web ce folosesc OpenSSL, rezultatele obținute sunt prezentate în lucrarea [23].

5. **Protecția serviciului de email contra spam folosind puzzle-uri criptografice de tip time-lock.** Spam-ul poate fi văzut ca un atac prin epuizarea resurselor serverelor de email, aducând costuri imense furnizorilor acestor servicii. Pentru a reduce profitul realizat de spamer, în secțiunea 4.2, este propusă o metodă de combatere a spam-ului bazată pe puzzle-uri criptografice de tip time-lock. Modul în care spameri sunt afectați de această soluție este prezentat în lucrarea [20].
6. **Calibrarea puzzle-urilor criptografice folosind localizare.** Pentru a spori eficiența puzzle-urilor criptografice în combaterea DoS și spam, în secțiunea 4.3, acestea sunt calibrate după locația adversarului. Metoda de localizare este una sigură, bazată pe o modificare a propusă în lucrare a algoritmul Vivaldi. Impactul computațional asupra clienților legitimi, precum și metoda de localizare sigură au fost prezentate în lucrarea [21].
7. **Identificarea intrușilor folosind amprentare fizică.** O identificare mai rafinată a adversarilor se poate realiza, după cum se poate vedea în secțiunea 4.4, folosind amprentarea fizică. . Acuratețea cu care dispozitivele pot fi amprentate, precum și metoda de amprentare și implicațiile acesteia sunt prezentate în lucrarea [18].

Acknowledgement

Această lucrare a fost parţial susţinută de grantul strategic POSDRU/88/1.5/S/50783, Proiectul ID50783 (2009), co-finanţat de Fondul Social European - Investește în Oameni, în cadrul Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007-2013.

Bibliografie

[1] T. Aura, P. Nikander, and J Leiwo, "Dos-resistant authentication with client puzzles," *Revised Papers from the 8th International Workshop on Security Protocols*, 2001. [Online]. http://research.microsoft.com/en-us/um/people/tuomaura/publications/aura-nikander-leiwo-protocols00.pdf

[2] J. Bakker. (2011) CSO Online. [Online]. http://www.cso-online.com/article/668169/ddos-attack-forces-dutch-bank-offline

[3] D.J. Bernstein. cr.yp.to. [Online]. http://cr.yp.to/syncookies.html

[4] BitDefender. (2011, May) BitDefender. [Online]. http://www.bit-defender.ro/news/27-dintre-utilizatorii-de-internet-nu-au-auzit-de-phish-ing-2110.html

[5] S. Blake-Wilson, M. Nystrom, D. Hopwood, and J. Mikkelsen, "RFC 4366 - Transport Layer Security (TLS) Extensions," *Network Working Group*, 2006.

[6] D. Boneh, B Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, vol. 17, pp. 297-319, 2004.

[7] C. Boyd, J. Gonzalez-Nieto, L. Kuppusamy, H. Narasimhan, C. Rangan, J. Rangasamy, J. Smith, D. Stebila, V. Varadarajan, "Cryptographic Approaches to Denial-of-Service Resistance," in *An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks*.: Springer Link, 2011, ch. 6, pp. 183-238.

[8] S. Brands and D. Chaum, "Distance-Bounding Protocols," *Advances in Cryptology EUROCRYPT '93*, 1993.

[9] C. Brenton. (2006, Apr.) Center, Global Incident analysis. [Online]. http://www.sans.org/reading_room/whitepapers/firewalls/egre-ss-filtering-faq_1059

[10] J.B.D. Cabrera et al., "Proactive detection of Distributed Denial of Service Attacks using MIB traffic variables––a feasibility study," *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management*, 2001.

[11] S. Ĉapkun, L. Buttyán, and J.-P. Hubaux, "SECTOR: secure tracking of node encounters in multi-hop wireless networks," *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, 2003.

[12] CERT Coordination Center. (1998, Jan.) Carnegie Mellon University. [Online]. http://www.cert.org/advisories/CA-1998-01.html

[13] E. M. Chan, C. A. Gunter, S. Jahid, E. Peryshkin, and D. Rebolledo, "Using rhythmic nonces for puzzle-based DoS resistance," *Proceedings of the 2nd ACM workshop on Computer security architectures*, 2008.

[14] Y. Chen et al., "Phoenix: Towards an Accurate, Practical and Decentralized Network Coordinate System," *Proceedings of the 8th International IFIP-TC 6 Networking Conference*, 2009.

[15] Y. Chen, Y. Xiong, X. Shi, B. Deng, and X. Li, "Pharos: A Decentralized and Hierarchical Network Coordinate," *GLOBECOM '07*, 2007.

[16] Cisco. (1997, Dec.) Cisco 7xx password buffer overflow. [Online]. http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-19971216-pw-buffer

[17] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC: Practical Internet Coordinates for Distance Estimation," *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, 2004.

[18] M. Cristea and Groza B., "Fingerprinting Mobile Phones Remotely with ICMP Timestamps," *Draft*, 2013.

[19] M. Cristea and B. Groza, "A Survey on Security Solutions of Top e-Banking Providers from an Eastern European Market," *Draft*, 2012.

[20] M. Cristea and B. Groza, "Augmenting a webmail application with cryptographic puzzles to deflect spam," *Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security, NTMS'2011*, 2011.

[21] M. Cristea and B. Groza, "Provable Synthetic Coordinates for Increasing PoWs Effectiveness Against DoS and Spam," *Proceedings of*

the 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, 2012.

[22] M. Cristea, B. Groza, and M. Iacob, "Some Security Issues In SCALANCEWireless Industrial Networks," *In proceedings of the sixth International Conference on Availability, Reliability and Security*, 2011.

[23] M. Cristea, B. Groza, and N. Robu, "Some vulnerabilities of IP based protocols still persistence in practice," *Workshop nr.1-"Cercetari doctorale în domeniul tehnic", Craiova, România*, 2012.

[24] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: a decentralized network coordinate system," *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004.

[25] Daemon9, Route, and Infinity. (1996, June) IP-Spoofing Demystified. [Online]. http://www.citi.umich.edu/u/provos/security/ph48.txt

[26] D. Davidowicz. (1999) Domain Name System (DNS) Security. [Online]. http://compsec101.antibozo.net/papers/dnssec/dnssec.html

[27] D. Dean and Stubblefield A., "Using Client Puzzles to Protect TLS," *Proceedings of the 10th conference on USENIX Security Symposium*, vol. 10, 2000. [Online]. http://www.csl.sri.com/users/ddean/papers/usenix01b.pdf

[28] O. Demir and B. Khan, "Finding DDoS attack sources: Searchlight localization algorithm for network tomography," *Proceedings of the 7th International Wireless Communications*, 2011.

[29] P. Druschel and G Banga, "LRP: A New Network Subsystem Architecture for Server," *Proceedings of the second USENIX symposium on Operating systems design and implementation*, 1996.

[30] M. Dunlop, Groat S., and Shelly D., "GoldPhish: Using Images for Content-Based Phishing Analysis," *Fifth International Conference on Internet Monitoring and Protection*, 2010.

[31] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, 1992. [Online].

http://www.wisdom.weizmann.ac.il/~naor/PAPERS/pvp.ps

[32] O. Eisen, "Catching the fraudulent Man-in-the-Middle and Man-in-the-Browser," *Network Security*, 2010.

[33] L. Enos. (2011, nov) Ecommerce Times. [Online]. http://www.e-commercetimes.com/story/8867.html

[34] T. Espiner. (2006, Nov) U.K. outlaws denial-of-service attacks. [Online]. http://news.cnet.com/U.K.-outlaws-denial-of-service-attacks/2-100-7348_3-6134472.html

[35] N. Falliere and L. O. Murchu. (2011, Feb) W32.Stuxnet Dossier. [Online]. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[36] P. Ferguson and D. Senie, "RFC 2872 - Network ingress filtering: defeating Denial of Service attacks which employ IP source address spoofing," *Network Working Group*, 2000.

[37] P. Francis et al., "IDMaps: a global internet host distance estimation service," *IEEE/ACM Trans. Netw.*, 2001.

[38] M. K. Franklin and D. Malkhi, "Auditable metering with lightweight security," *Proceedings of the First International Conference on Financial Cryptography*, 1998. [Online]. http://www.springerlink.com/index/3205-h17579123674.pdf

[39] S. Furnell, V. Tsaganidi, and A. Phippen, "Security beliefs and barriers for novice Internet users," *Computers & Security*, vol. 27, 2008.

[40] S. Garera, N. Provos, M. Chew, and A.D. Rubin, "A framework for detection and measurement of phishing attacks.," *In proceedings of the 2007 ACM workshop on Recurring malcodes*, 2007.

[41] T.M Gil and M. Poleto, "MULTOPS: a data-structure for bandwidth attack detection," *Proceedings of 10th Usenix Security Symposium*, 2001.

[42] B. Groza, *Introducere în sistemele de criptografie cu cheie publică*.: Editura Politehnică, 2007.

[43] B. Groza, M. Minea, M. Cristea, P.S. Murvay, and M. Iacob, "Protocol vulnerabilities in practice: causes, modeling and automatic

detection," *Proceedings of the Romanian Academy Series A: Mathematics, Physics, Technical Sciences, Information Science*, vol. 13, no. 2, 2012.

[44] B. Groza and B. Warinschi, "Revisiting difficulty notions for client puzzles and dos resilience," *Proceedings of the 15th international conference on Information Security*, pp. 39-54, 2012.

[45] P. Gühring. (2006, Sep.) CAcert. [Online]. http://www.ca-cert.at/svn/sourcerer/CAcert/SecureClient.pdf

[46] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: estimating latency between arbitrary internet end hosts," *SIGCOMM Comput. Commun. Rev.*, 2002.

[47] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers & Security*, vol. 30, 2011.

[48] A. Herzberg, "Why Johnny can't surf (safely)? Attacks and defenses for web users," *Computers & Security*, vol. 28, 2009.

[49] A. Hiltgen, T. Kramp, and T. Weigold, "Secure Internet Banking Authentication," *IEEE Security and Privacy*, 2006.

[50] C. Hostiuc. (2012, Feb.) Ziarul Financiar. [Online]. http://www.zf.ro/banci-si-asigurari/topul-integral-al-bancilor-pe-2011-ce-active-detin-si-ce-profit-pierdere-au-avut-9241764

[51] K. J. Houle and G. M. Weaver. (2001, Oct.) Carnegie Mellon University. [Online]. www.cert.org/archive/pdf/DoS_trends.pdf

[52] Y. Huang and J.M. Pullen, "Countering Denial of Service attacks using congestion triggered packet sampling and filtering," *Proceedings of the 10th International Conferenceon Computer Communiations and Networks*, 2001.

[53] IBM. IBM Internet Security Systems. [Online]. http://xforce.i-ss.net/xforce/xfdb/47

[54] IBM. IBM Security Services. [Online]. http://xforce.iss.net/x-force/xfdb/464

[55] V. Jacobson, R. Braden, and D. Borman, "RFC 1323 - TCP Extensions for High Performance," *Network Working Group*, 1992.

[56] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008.

[57] A. Jules and J. Brainard, "Client puzzles: A cryptographic defense against connection depletion attacks," *In proceedings of the the sixth Annual Symposium on Network and Distributed System Security (NDSS'99)*, 1999.

[58] M. A. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, "Virtual networks under attack: disrupting internet coordinate systems," *Proceedings of the 2006 ACM CoNEXT conference*, 2006.

[59] D. Karig and R. Lee, "Remote Denial of Service Attacks and Countermeasures," 2001.

[60] T. Kohno, A. Broido, and K. C. Claffy, "Remote Physical Device Fingerprinting," *IEEE Trans. Dependable Secur. Comput.*, 2005.

[61] E. Kritzinger and S. von Solms, "Cyber security for home users: A new way of protection through awareness enforcement," *Computers & Security*, 2010.

[62] B. Laurie and R. Clayton. (2004) "Proof-of-Work" proves not to work. [Online]. www.cl.cam.ac.uk/~rnc1/proofwork2.pdf

[63] W. Lee and S.J. Stolfo, "Data mining approaches for intrusion detection," *Proceedings of the 7th USENIX Security Symposium*, 1998.

[64] W. Lee, S.J. Stolfo, and K.W. Mok, "A data mining framework for building intrusion detection models," *Proceedings of the1999 IEEE Symposium on Security and Privacy*, 1999.

[65] C. Ludl, S. Mcallister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," *In proceedings of the 4th international conference on Detection of Intrusions and Malware*, 2007.

[66] Y. Mao and L. K. Saul, "Modeling distances in large-scale

networks by matrix factorization," *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004.

[67] A.J. Menezes, P.C. Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*.: CRC Press, 1996.

[68] R.C. Merkle, "Secure communications over insecure channels," *Communications of the ACM*, no. ISSN: 0001-0782, 1978.

[69] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," *Proceedings of ICNP*, 2002.

[70] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements," *INFOCOM, 1999*, 1999.

[71] T. Moore and R. Clayton, "The impact of incentives on notice and take-down," *In proceedings of the 7th Workshop on Economics of Information Security*, 2008.

[72] S.J. Murdoch and R. Anderson, "Verified by visa and mastercard securecode: or, how not to design authentication," *Proceedings of the 14th international conference on Financial Cryptography and Data Security*, 2010.

[73] T. E. Ng and H. Zhang, "A network positioning system for the internet," *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2004.

[74] T. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," *Proceedings of the twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002.

[75] D. Oancea. (2012, Jan.) BUSINESS Magazin. [Online]. http://www.businessmagazin.ro/actualitate/epayment-tentativele-de-frauda-in-2011-au-scazut-cu-15-fata-de-2010-9155515

[76] Y. Pan and X. Ding, "Anomaly based web phishing page detection.," *In proceedings of the 22nd Annual Computer Security Applications Conference*, 2006.

[77] K. Park and H. Lee, "On the effectiveness of route-based packet

filtering for distributed DoS attack prevention in power-law internets," *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, 2001.

[78] V. Paxson, "End-to-end Internet packet dynamics," *ACM Transactions on Networking*, vol. 7, 1999.

[79] V. Paxson, "On calibrating measurements of packet transit times," *SIGMETRICS Perform. Eval. Rev.*, 1998.

[80] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from Distributed Denial of Service attack using history-based IP filtering," *ICC 2003*, 2003.

[81] C. Perkins, "RFC 3344 - IP mobility support for IPv4," *Network Working Group*, 2002.

[82] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA CryptoBytes*, 2002.

[83] M. Pias, J. Crowcroft, S. R. Wilbur, T. L. Harris, and S. N. Bhatti, "Lighthouses for Scalable Distributed Location," *Proceedings of the Second International Workshop on Peer-to-Peer Systems*, 2003.

[84] J. Postel, "RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL," *Network Working Group*, 1981.

[85] S.G. Prevost, G.G. Granadillo, and M. Laurent, "A dual approach to detect pharming attacks at the client-side.," *In proceedings of the 4th International Conference on New Technologies, Mobility and Security*, 2011.

[86] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," *Proceedings of the twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002.

[87] R. L. Rivest, A. Shamir, and D. A. Wagner. (1996) Massachusetts Institute of Technology. [Online]. people.csail.mit.edu/rivest/Rivest-ShamirWagner-timelock.ps

[88] S. Santesson, "RFC 4680 - TLS Handshake Message for Supplemental Data," *Network Working Group*, 2006.

[89] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," *Proceedings of the 2nd ACM workshop on Wireless security*, 2003.

[90] R. Segal, J. Crawford, B. L. J. Kephart, and B. Leiba, "SpamGuru: An Enterprise Anti-Spam Filtering System," *In Proceedings of the First Conference on E-mail and Anti-Spam*, 2004.

[91] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in Euclidean space," *IEEE/ACM Trans. Netw.*, 2004.

[92] AG. Siemens. (2006, June) Siemens Automation. [Online]. http://www.automation.siemens.com/doconweb/pdf/SINUMERIK_SINAMICS_04_2010_E/S7_BA31A.pdf?p=1.

[93] AG. Siemens. (2010, Jan.) Siemens Automation. [Online]. http://cache.automation.siemens.com/dnl/zQ/zQxMDE1AAAA_9975764_HB/SYH_IWLAN_76.pdf

[94] M. Stahlberg, "The Trojan Money Spinner," *Proceedings of virus bulletin conference*, 2007.

[95] L.D. Stein and J.N. Stewart. (2002, Feb.) W3C. [Online]. http://www.w3.org/Security/Faq/

[96] P.R. Talpade, G. Kim, and S. Khurana, "NOMAD: Trafficbased network monitoring framework for anomaly detection," *Proceedings of the Fourth IEEE Symposium on Computers and Communications*, 1998.

[97] L. Tang and M. Crovella, "Virtual landmarks for the Internet," *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003.

[98] N. Utakrit, "A Review of Browser Extensions, a Man-in-the-Browser Phishing Techniques Targeting Bank Customers," *Proceedings of the 7th Australian Information Security Management Conference*, 2009.

[99] F. Veysset, O. Courtay, and O. Heen, "New Tool And Technique For Remote Operating System Fingerprinting," 2002.

[100] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten, "New Client Puzzle Outsourcing Techniques for DoS Resistance," *Proceedings of the 11th ACM conference on Computer and communications security*,

2004. [Online]. http://www.cse.umich.edu/~jhalderm/pub/papers/puzzle-ccs04.pdf

[101] T. Weigold et al., "The Zurich Trusted Information Channel --- An Efficient Defence Against Man-in-the-Middle and Malicious Software Attacks," *Proceedings of the 1st international conference on Trusted Computing and Trust in Information Technologies: Trusted Computing - Challenges and Applications*, 2008.

[102] C.S. Weir, G. Douglas, M. Carruthers, and M. Jack, "User perceptions of security, convenience and usability for ebanking authentication tokens," *Computers & Security*, vol. 28, 2009.

[103] Z. Zhang, J.I. Hong, and L.F. Cranor, "Cantina: a content-based approach to detecting phishing web sites.," *In proceedings of the 16th international conference on World Wide Web.*, 2007.

Index

Anexe

A1. Rezultate obținute pe parcursul stagiului doctoral

- **Lucrare publicată în jurnal ISI:**
  - B. Groza, M. Minea, M. Cristea, P.S. Murvay și M. Iacob, "Protocol vulnerabilities in practice: causes, modeling and automatic detection," Proceedings of the Romanian Academy Series A: Mathematics, Physics, Technical Sciences, Information Science, vol. 13, no. 2, 2012 (ISI Web of knowledge).
- **Lucrări publicate la conferințe internaționale indexate BDI:**
  - M. Cristea și B. Groza, "Augmenting a webmail application with cryptographic puzzles to deflect spam," Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security, NTMS'2011, 2011 (IEEE Explore, DBLP, Scopus);
  - M. Cristea, B. Groza, și M. Iacob, "Some Security Issues In SCALANCE Wireless Industrial Networks," In proceedings of the sixth International Conference on Availability, Reliability and Security, 2011 (IEEE Explore, DBLP, Scopus, ACM);
  - M. Cristea și B. Groza, "Provable Synthetic Coordinates for Increasing PoWs Effectiveness Against DoS and Spam," Proceedings of the 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, 2012 (IEEE Explore, DBLP, ACM).
- **Lucrări în workshop-uri organizare în cadrul școlii doctorale:**
  - M. Cristea și N. Robu, "Some vulnerabilities of IP based protocols still persistent in practice"– Workshop: "Cercetari doctorale în domeniul tehnic",Craiova, România, februarie, 2011;
  - M. Cristea, "Cryptographic Solutions for Resilience Against DoS Attacks" – Workshop: "Interdisciplinaritatea si managementul cercetării", Timișoara, Romania, noiembrie 2011.
- **Lucrări în curs de publicare:**
  - M. Cristea și B. Groza, "A Survey on Security Solutions of Top e-Banking Providers from an Eastern European Market", World Wide Web Journal, Springerlink;

- M. Cristea și Groza B., "Fingerprinting Mobile Phones Remotely with ICMP Timestamps", IEEE Communications Letters, IEEE Comm. Soc.

- **Alte activități:**
  - Participare la școala de vara SWING 2011, Bertinoro, Italia;
  - Recenzor pentru: ARES, Viena, Austria, 2012; CRiSIS, Timișoara, România, 2012; ISETC, Timișoara, România, 2012;
  - Membru în grantul DISSIS (grant PN2 2008-2011) – cercetare în proiectarea și implementarea soluțiilor moderne de securitate pentru sisteme distribuite (SCADA, DCS) și control automat, cu aplicații în distribuția gazelor, IDEI, PN2 940/2008;
  - Membru IEEE.