DETECŢIA ATACURILOR INFORMATICE ŞI A ANOMALIILOR DIN REŢELE DE CALCULATOARE PRIN SUPRAVEGHEREA CARACTERISTICILOR DE TRAFIC FOLOSIND TEHNICI DE PRELUCRARE A SEMNALELOR

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea Politehnica Timişoara
în domeniul INGINERIE ELECTRONICĂ
ȘI TELECOMUNICAȚII
de către

ing. Florin Vancea

Conducător ştiinţific:	prof.univ.dr.ing Ioan Naforniță
Referenţi ştiinţifici:	prof.univ.dr. Monica Borda
	conf.univ.dr.ing. Emil Cebuc
	prof.univ.dr.ing. Alexandru Isar

Ziua susţinerii tezei: 10 Ianuarie 2014

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
12. Ingineria sistemelor
13. Inginerie energetică
14. Calculatoare şi tehnologia informaţiei
15. Ingineria materialelor
16. Inginerie şi Management

Universitatea Politehnica 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./fax 0256 403823
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Rețelele de calculatoare moderne transportă informație cu valoare semnificativă astfel încât ele sunt supuse permanent atacurilor informatice. In același timp complexitatea lor tehnică și complexitatea contextului în care funcționează, corelate cu limitările care sunt impuse frecvent asupra resurselor dedicate pentru configurare și întreținere, permit apariția unor situații în care apar anomalii care le impactează funcționarea.

In baza presupunerii verificate că traficul constatat în diverse puncte din rețea poate să fie o sursă valoroasă de informație am analizat metode de a transforma valorile măsurate de trafic în decizii care pot sprijini activitatea de protecție și administrare. Am identificat metode de a colecta și selecta serii de date care să prezinte maximum de relevanță. Am identificat metode de a analiza aceste serii de date folosind tehnici de prelucrare a semnalelor. Am dezvoltat unelte destinate să faciliteze aceste operații și unelte destinate cercetării acestei problematici, în speranța construcției unui sistem complet (NEAR) care să ofere un instrument puternic de detecție și sprijin al deciziei. Procesul de dezvoltare și aplicare a acestor unelte a condus la rezultate noi și la confirmarea unor informații cunoscute dar în moduri inovatoare.

La sfârșitul îndelungatei perioade în care am pregătit această teză am ocazia să privesc în urmă și să ofer câteva cuvinte de mulțumire celor care m-au ajutat să ajung aici.

In primul rând, aș dori să mulțumesc din suflet conducătorului meu științific, domnului Profesor Dr. Ioan Naforniță, pentru priceperea cu care m-a atras să înțeleg semnalele dincolo de matematica lor încă de pe vremea când îi eram student, pentru ideile și sugestiile primite de-a lungul timpului și pentru răbdarea și calmul cu care m-a împins să termin această teză în momentele dificile în care credeam că n-am să ajung niciodată la capăt.

Vreau să mulțumesc doamnei Profesor Dr. Miranda Naforniță pentru căldura cu care m-a încurajat să merg mai departe și pentru îndrumarea sa legată de comportarea statistică a traficului. Vreau să mulțumesc de asemenea domnului Profesor Dr. Alexandru Isar pentru atenția acordată încercărilor mele de a găsi orientarea corectă a acestei teze spre forma curentă și doamnei Profesor Dr. Cornelia Gordan pentru îndemnurile permanente de a duce teza la bun sfârșit.

Mulțumesc domnului Profesor. Dr. Luca Deri (Universitatea din Pisa) pentru că mi-a permis accesul gratuit la componente de circulație limitată a sistemului său de captură și statistică și astfel am putut să înțeleg mai bine problematica specifică legată de captura de trafic.

Mulțumesc tuturor colegilor și prietenilor care au avut răbdare să termin această lucrare și care m-au sprijinit pe durata scrierii ei.

La sfârșitul acestei liste dar nu în ultimul rând, mulțumesc familiei mele care a participat direct la tot drumul lung și nu întotdeauna drept care a fost pregătirea acestei teze. Le mulțumesc pentru răbdare, pentru încurajare și pentru felul în care au suplinit prezența mea la sarcinile mărunte dar necesare, ca să pot să îmi îndrept eforturile spre finalizarea acestei lucrări.

Timişoara, Decembrie 2013 Vancea Florin

Vancea, Florin

Detecţia atacurilor informatice

şi a anomaliilor

din reţele de calculatoare

prin supravegherea caracteristicilor de trafic

folosind tehnici de prelucrare a semnalelor

Teze de doctorat ale UPT, Seria 7, Nr. 66, Editura Politehnica, 2014, 126pagini, 86 figuri, 7 tabele.

ISSN: 1842-7014

ISBN: 978-606-554-752-0

Cuvinte cheie: rețele de calculatoare, securitate, detecția intruziunilor, detecția anomaliilor, descompunere wavelet

Rezumat,
Securitatea și disponibilitatea unei rețele de calculatoare poate fi îmbunătățită prin aplicarea unor metode de detecție a intruziunilor și a anomaliilor în general.
Prin aplicarea unor principii de captură și analiză a traficului bazate pe tehnici de prelucrare a semnalelor autorul introduce un astfel de sistem de detecție.
In lucrare se analizează principii de analiză și detalii de implementare. Se introduc de asemenea direcții noi de dezvoltare pentru astfel de sisteme.

CUPRINS

LISTA DE FIGURI 7
LISTA DE TABELE 12
1. INTRODUCERE 13
1.1. Contribuții personale 14
1.2. Structura tezei 15
2. SISTEME DE DETECȚIE A INTRUZIUNILOR 17
2.1. Monitorizarea rețelelor de calculatoare 17
2.1.1. Scurtă istorie 18
2.2. Mecanisme de captură a traficului 19
2.3. Detecția anomaliilor folosind inspecția pachetelor 20
2.4. Extracția caracteristicilor de trafic pentru analiza 21
2.5. Analiza caracteristicilor de trafic 22
2.5.1. Distribuția traficului de rețea 23
2.5.2. Tehnici de detecție a anomaliilor 24
2.5.3. Transformarea wavelet – unealtă de analiză traficului 25
2.5.4. Metode existente de analiză a traficului 27
2.6. Soluţia propusă – analiza combinată și adaptivă a multiple caracteristici 29
2.6.1. Limitări în soluțiile și analizele anterioare 29
2.6.2. Descriere generală a soluției propuse 31
2.6.3. Aplicabilitate 33
3. CAPTURA ŞI EXTRACŢIA SERIILOR DE DATE 35
3.1. Modelul rețelei supravegheate 35
3.2. Principii de îmbunătățire a ratei de detecţie 37
3.2.1. Principiul de îmbunătățire SNR 37
3.2.2. Principiul darknet 38
3.2.3. Principiul de corelație 38
3.3. Mecanisme de captură și procesare primară 39
3.3.1. Separarea în tipuri de trafic și domenii de autoritate 39
3.3.2. Separarea după rolul partenerilor în relația client-server 42
4. ANALIZA SERIILOR DE DATE 47
4.1. Tipuri de anomalii detectabile în seriile de date 47
4.2. Instrumente de detecţie a anomaliilor singulare. 52
4.2.1. Detecţie cu prag. 52
4.2.2. Evaluarea tehnicilor de detecţie cu prag 55
4.2.3. Detecţie cu procesare timp-frecvenţă 63
4.2.4. Detecţie cu descompunere wavelet şi cumulare de praguri 65
4.2.5. Detecţie cu denoising incomplet pe bază de wavelet 67
5. CONSIDERAȚII PRACTICE. SISTEMUL NEAR 81
5.1. Evaluarea IDS DARPA'98 82
5.2. Evaluarea IDS DARPA'99 84
5.3. Rețele proprii ca sursă de date pentru analiză 86
5.4. Sistemul NEAR – platformă de captură și analiză a anomaliilor din rețea 89
5.5. NEAR-agent - instrument de colectare trafic și extracție serii de date 90
5.6. NEAR-GUI - instrument de vizualizare serii de date 96
5.7. Atacuri prezente în seturile DARPA, analizate cu NEAR 98
5.7.1. arppoison 100

5.7.2. ipsweep 100
5.7.3. mailbomb 103
5.7.4. neptune 104
5.7.5. nmap 105
5.7.6. PoD 105
5.7.7. portsweep 105
5.7.8. resetscan 106
5.7.9. SATAN 106
5.7.10. smurf 106
5.7.11. tcpreset 106
5.8. Secvenţe din reţele proprii, analizate cu NEAR 107
5.8.1. Absenţă trafic în reţeaua EDU 107
5.8.2. Atac fals în reţeaua EDU 108
5.8.3. Trafic periodic normal în reţeaua RD 109
5.8.4. Variaţie a comportării periodice în reţeaua RD 113
5.8.5. Evoluţie normală a protocolului NTP în reţeaua RD 115
6. CONCLUZII ŞI CONTRIBUŢII PERSONALE 117
7. Bibliografie citată şi consultată 119

LISTA DE FIGURI

Figura 1 Lanț de comunicare între două noduri terminale care trece printr-un nod releu. 36

Figura 2 Model de multiple rețele locale cu noduri terminale și noduri releu. 37

Figura 3 Trafic de nivel 2 OSI la interfața de captură. Se evidenţiază cele 4 tipuri de trafic și domeniile de autoritate. 40

Figura 4 Trafic de nivel 3 OSI la interfața de captură. Se evidenţiază cele 6 tipuri de trafic și domeniile de autoritate locale și globale. 42

Figura 5. Anomalie de tip 1: impuls de amplitudine mare și durată scurtă 48

Figura 6. Anomalie de tip 2: componentă de amplitudine moderată, pe o durată mai extinsă de timp 48

Figura 7. Trafic periodic generat de două instrumente de monitorizare și transformarea FFT (doar modul) 49

Figura 8. Serii corelate generate de buclă de build continuu. In zona marcată s-a derulat un build, în exterior se monitorizează doar modificările. 50

Figura 9. Exemple de corelație în trafic stație-router (stânga) și stație-stație (dreapta) 51

Figura 10. Detecția unui impuls perturbator. a – serie inițiala (Hurst=0.1); b – serie cu impuls unitar; c – varianta filtrata cu filtru medie rulantă; d – rezultat după detector (3-sigma); e – serie după aplicare SNEO (Bartlett 9); f – rezultat SNEO după detector (5-sigma) 56

Figura 11. Amplitudinea relativă a impulsului perturbator detectabil cu prag si filtru-fereastra de 32 puncte funcție de durata și factorul Hurst. 57

Figura 12. Amplitudinea relativă a impulsului perturbator detectabil cu prag si filtru-fereastra de 16 puncte funcție de durata și factorul Hurst 58

Figura 13. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO k=1 şi fereastră Bartlett 9. Pragul superior (roşu) este cel pentru falsuri pozitive. 59

Figura 14. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO k=1 şi fereastră Bartlett 17. Pragul superior (roşu) este cel pentru falsuri pozitive. 60

Figura 15. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 33. Pragul superior (albastru) în zona H=0..0.1 este cel pentru falsuri negative.....60

Figura 16. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 65. Pragul superior (albastru) este cel pentru falsuri negative. 61

Figura 17. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=4 şi fereastră Bartlett 9. Pragul superior (albastru) în zona H=0..0.1 este cel pentru falsuri negative.....61

Figura 18. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=16 şi fereastră Bartlett 9. Pragul superior (albastru) este cel pentru falsuri negative. 62

Figura 19. Schema de analiză folosind descompunere wavelet şi cumulare de praguri pe coeficienţi detaliu. 65

Figura 20. Detecţie cu descompunere wavelet, praguri şi cumulare de decizie. a – serie inițiala (Hurst=0.3); b – serie perturbată cu impuls unitar de amplitudine relativă 0.2 și durata 128; c – descompunere wavelet (Meyer, n=7); d – rezultat la ieșirea sumatorului, înainte de prag final. 66

Figura 21. Schema de analiză folosind o variantă de denoising incomplet pe bază de wavelet şi praguri. 68

Figura 22. Detecţie cu denoising incomplet tip wavelet şi praguri. a – serie inițiala (Hurst=0.3); b – serie perturbată cu impuls unitar de amplitudine relativă 0.2 și durata 128; c – descompunere wavelet (Meyer, n=7); d – semnal reconstruit; e – ieşire detector la 6-sigma.....69

Figura 23. Performanţe la metoda cu praguri şi cumulare, wavelet Meyer discret, 7 nivele 71

Figura 24. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-1, 7 nivele 71

Figura 25. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-2, 7 nivele 71

Figura 26. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-3, 7 nivele 71

Figura 27. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-4, 7 nivele 71

Figura 28. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-5, 7 nivele 71

Figura 29. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-1, 7 nivele 72

Figura 30. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-3, 7 nivele 72
Figura 31. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-5, 7 nivele 72
Figura 32. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-1, 7 nivele 72
Figura 33. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-3, 7 nivele 72
Figura 34. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-5, 7 nivele 72
Figura 35. Performanţe la metoda cu praguri şi cumulare, wavelet Biortogonal-3.3, 7 nivele 73
Figura 36. Performanţe la metoda cu praguri şi cumulare, wavelet Biortogonal-5.5, 7 nivele 73
Figura 37. Performanţe la metoda cu denoising parţial, wavelet Meyer discret, 7 nivele, prag 6-sigma.....74
Figura 38. Performanţe la metoda cu denoising parţial, wavelet Daubechies-1, 7 nivele, prag 6-sigma 74
Figura 39. Performanţe la metoda cu denoising parţial, wavelet Daubechies-2, 7 nivele, prag 6-sigma 74
Figura 40. Performanţe la metoda cu denoising parţial, wavelet Daubechies-3, 7 nivele, prag 6-sigma 74
Figura 41. Performanţe la metoda cu denoising parţial, wavelet Daubechies-4, 7 nivele, prag 6-sigma 74
Figura 42. Performanţe la metoda cu denoising parţial, wavelet Daubechies-5, 7 nivele, prag 6-sigma 74
Figura 43. Performanţe la metoda cu denoising parţial, wavelet Symlet-1, 7 nivele, prag 6-sigma.....75
Figura 44. Performanţe la metoda cu denoising parţial, wavelet Symlet-3, 7 nivele, prag 6-sigma.....75
Figura 45. Performanţe la metoda cu denoising parţial, wavelet Symlet-5, 7 nivele, prag 6-sigma.....75
Figura 46. Performanţe la metoda cu denoising parţial, wavelet Coiflet-1, 7 nivele, prag 6-sigma.....75
Figura 47. Performanţe la metoda cu denoising parţial, wavelet Coiflet-3, 7 nivele, prag 6-sigma.....75
Figura 48. Performanţe la metoda cu denoising parţial, wavelet Coiflet-5, 7 nivele, prag 6-sigma.....75
Figura 49. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-3.3, 7 nivele, prag 6-sigma.....76

Figura 50. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-5.5, 7 nivele, prag 6-sigma 76
Figura 51. Performanţe la metoda cu denoising parţial, wavelet Meyer discret, 7 nivele, prag 10-sigma 76
Figura 52. Performanţe la metoda cu denoising parţial, wavelet Daubechies-1, 7 nivele, prag 10-sigma 76
Figura 53. Performanţe la metoda cu denoising parţial, wavelet Daubechies-2, 7 nivele, prag 10-sigma 77
Figura 54. Performanţe la metoda cu denoising parţial, wavelet Daubechies-3, 7 nivele, prag 10-sigma 77
Figura 55. Performanţe la metoda cu denoising parţial, wavelet Daubechies-4, 7 nivele, prag 10-sigma 77
Figura 56. Performanţe la metoda cu denoising parţial, wavelet Daubechies-5, 7 nivele, prag 10-sigma 77
Figura 57. Performanţe la metoda cu denoising parţial, wavelet Symlet-1, 7 nivele, prag 10-sigma 77
Figura 58. Performanţe la metoda cu denoising parţial, wavelet Symlet-3, 7 nivele, prag 10-sigma 77
Figura 59. Performanţe la metoda cu denoising parţial, wavelet Symlet-5, 7 nivele, prag 10-sigma 78
Figura 60. Performanţe la metoda cu denoising parţial, wavelet Coiflet-1, 7 nivele, prag 10-sigma 78
Figura 61. Performanţe la metoda cu denoising parţial, wavelet Coiflet-3, 7 nivele, prag 10-sigma 78
Figura 62. Performanţe la metoda cu denoising parţial, wavelet Coiflet-5, 7 nivele, prag 10-sigma 78
Figura 63. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-3.3, 7 nivele, prag 10-sigma 78
Figura 64. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-5.5, 7 nivele, prag 10-sigma 78
Figura 65. Comparaţie între simulări pentru Symlet-5 şi Biortogonal-3.3. Sus – prag şi cumulare; mijloc – denoising parţial şi prag 6-sigma; jos – denoising parţial şi prag 10-sigma 79
Figura 66 Diagrama reţelei DARPA'98 (conform setului public de date) 83
Figura 67. Diagrama reţelei DARPA’99 (conform setului public de date) 85
Figura 68. Diagrama reţelei private RD 87
Figura 69. Diagrama reţelei private EDU 88

Figura 70. Sistem de detecţie a intruziunilor NEAR 89
Figura 71. Interfaţa aplicaţiei NEAR-GUI 97
Figura 72. Atacuri ipsweep evidenţiate în NEAR-GUI.....101
Figura 73. Detecţia de anomalie pentru seria 50 (ARP) din n99w2d2-darpain. 102
Figura 74. Detecţia de anomalie pentru seria 48 (UNKN PORT) din n98w2d2 103
Figura 75. Atac mailbomb pe seria porturilor de e-mail (25, 110).....104
Figura 76. Atac neptune din exterior şi SATAN din interior pe seriile n99w2d4. Seria de sus este capturată în exterior, seria de jos este capturată în interior. Atacul din stânga pe seria de jos este SATAN. 104
Figura 77. Cădere de tensiune temporară reflectată în seriile de date (reţeaua EDU).....107
Figura 78. Trafic neașteptat pe port necunoscut. Sus: anomalie ARP şi trafic pe porturi neclasificate. Jos: Aceeaşi semnătură de trafic pe porturi neclasificate, fără corespondenţă în SYN-neclasificate, ...108
Figura 79. Trafic periodic între S-CB (jos) şi GW-I (sus). Marcajul indică o discontinuitate posibil interesantă. 109
Figura 80. Transformata STFT pentru seria GW-I, index 19. Reprezentare implicită Matlab, normalizat. 110
Figura 81. Transformata STFT pentru seria GW-I, index 19. A: Trafic apcupsd la 30 s; B: trafic apcupsd la 60 s; C: trafic mrtg la 5 min 111
Figura 82. Transformata STFT pentru seria S-CB, index 3. A: trafic apcupsd la 30 s; B: trafic mrtg la 5 min.....112
Figura 83. Activitate pe seria SYN la server de build periodic S-CB. 113
Figura 84. Transformata STFT pentru seria S-CB/52, 23-Aug-2012. Se observă efectul buclei de build nefolosite. 114
Figura 85. Transformata STFT pentru seria S-CB/52, 24-Aug-2012. Se observă efectul buclei de build active. 114
Figura 86. Transformata STFT pentru seria S-CB index 35 (NTP simetric). Se observă evoluţia gradată şi repetată în timp a protocolului NTP de la poll-interval 256 la poll-interval 1024 115

LISTA DE TABELE

Tabel 1. Regula de corespondență pentru detecția traficului de tip 4, 5, 6 la nivel 3 OSI 42

Tabel 2. Exemplu de grupare a porturilor cu trafic normal 45

Tabel 3. Eficienţa relativă a unor funcţii wavelet, ca timpi de execuţie 80

Tabel 4. Categorii de trafic folosite de NEAR-agent și semnificația lor 94

Tabel 5. Incadrarea porturilor în clase folosită pentru analiza capturilor DARPA 95

Tabel 6. Capturi care depășesc o zi în setul DARPA'98 99

Tabel 7. Capturi care prezintă situații speciale în setul DARPA'99 99

1. INTRODUCERE

Undeva la originea acestei teze stă experiența directă acumulată în postura de administrator neoficial al unor rețele de calculatoare. In această poziție am fost confruntat direct cu probleme de asigurare a securității acestora, de asigurare a funcționalității acestora și de asigurare a performanței acestora, nu neapărat în această ordine. Toți cei care au avut de a face cu așa ceva cred că pot să confirme că nu este o sarcină simplă și că dificultatea crește probabil exponențial cu dimensiunea rețelei și cu nivelul de sofisticare al utilizatorilor acesteia. Experiența mea directă în calitate de administrator cu drepturi depline s-a limitat la rețele de dimensiune cel mult medie dar la capitolul complexitate am avut ocazia să observ și să vizitez rețele de dimensiuni semnificative.

O observație pe care am făcut-o repede este că instrumentele de administrare contează. Fenomenele care au loc în rețea nu doar că sunt invizibile „cu ochiul liber"[1] dar se și petrec la scări de timp greu accesibile observatorului uman. Un instrument de administrare bun trebuie să captureze esența despre comportarea rețelei și să o prezinte într-o formă care să înlesnească decizia de a reacționa proactiv sau corectiv într-un fel sau altul. In această direcție o unealtă ca MRTG, deși la origine o simplă reprezentare grafică a traficului de rețea constatat în timp pe o interfață de router, devine un mijloc prin care se poate arunca o privire asupra a ceea ce se întâmplă (sau mai degrabă ce s-a întâmplat) în rețea. In momentul în care am observat cât de bine se pot corela anumite evenimente care știam că au avut loc cu aspectul traficului în diverse puncte de rețea colectat de MRTG am început să mă gândesc la posibilități de a valorifica această sursă de informație pentru a ușura activitatea celor care administrează rețele.

Odată ce am intrat mai adânc în subiect am constatat că există deja realizări semnificative în acest domeniu dar și că s-ar putea să mai fie unele lucruri de descoperit sau măcar de implementat. Teza mea este (din punctul meu de vedere) doar un început în această direcție pe care intenționez să o urmez.

Așadar, pentru a putea să prelucrăm această categorie de informație va trebui în primul rând să o capturăm și să o transformăm în serii de numere pe care le putem prelucra apoi cu un algoritm oarecare. Există instrumente care fac deja asta, dar din nou din experiența practică am extras ideea că nu ne interesează întotdeauna tot traficul și ca uneori ar fi mult mai interesant să avem doar serii de date referitoare doar la segmentul care este afectat de o problemă anume. Am dezvoltat așadar un mecanism care să descompună traficul pe care îl vedeam agregat în graficele propuse de domnul Oetiker în MRTG și am constatat că o parte de drum a fost deja parcursă de domnul Deri în NTOP. Metoda pe care o propun însă este ușor diferită pentru că aplică un punct de vedere personal, unde „personal" este luat ad-literam: „traficul *meu*" și „traficul *nostru*" spre deosebire de „traficul *străin*" și pentru a obține asta descompune pachetele după criterii cumulate de la nivelele 2, 3 și 4 ISO-OSI.

Odată ce am obținut serii de date, putem aplica metode clasice de prelucrarea semnalelor asupra lor. Putem de exemplu să căutăm impulsuri sau

[1] De fapt, ar trebui să spun *aproape* invizibile. Un administrator cu experiență aplică intuitiv și poate chiar la nivel subconștient ideile pe care le promovez în această lucrare și își dă seama când „clipesc LED-urile altfel"

absența lor, să căutăm serii periodice sau absența lor, să căutăm corelație sau absența ei. Și aici există deja multă experiență adunată de înaintași. Nu trebuie să uităm însă specificul „semnalelor" pe care le analizăm. Ele sunt rezultate dintr-un fenomen particular și complex (traficul de rețea) care încă mai oferă motiv de dispută academică privitor la caracteristicile sale statistice. Mai mult, urmează să prelucrăm serii de date care nici măcar nu reprezintă volum integral de trafic, pentru care există suficientă experiență, ci sub-componente rezultate în urma descompunerii pe criterii neuniforme.

In final, trebuie să recunosc orientarea mea personală ca practician. Toate eforturile pe care le-am depus au o puternică motivație practică, așa încât o bună parte din realizări se sprijină pe implementări efective, prezentate în cele ce urmează. Ambiția de a construi la un moment dat un sistem semiautomat de detecție a anomaliilor și eventual de reacție (semi) automată în mod expert stă la baza eforturilor pe care le-am depus și sper să funcționeze în continuare ca factor motivant de a duce această linie de cercetare mai departe.

1.1. Contribuții personale

Contribuțiile personale pe care le aduce această teză sunt:

- Introducerea unui set de principii de captură și extracție care conduc la îmbunătățirea calității analizei
- Descompunerea traficului după dimensiunea „port server" pentru îmbunătățirea ratei semnal-zgomot la detecția anomaliilor
- Includerea în analiză a informației derivate din trafic non-IP (ARP, 802.3)
- Introducerea anomaliei ca extensie a ideii de atac și țintă a eforturilor de detecție, împreună cu identificarea unui set de anomalii detectabile cu metoda descrisă
- Introducerea corelației cu mărimi non-rețea ca instrument de confirmare a anomaliilor
- Introducerea corelației între serii colectate în puncte diferite ca instrument de confirmare sau rafinare a anomaliilor
- Analiza explicită a componentelor normale cu puternică periodicitate pentru identificarea anomaliilor „prin lipsă"
- Analiza comportării metodelor de detecție cu prag la variația nivelului de auto-similaritate în seria de date și la aspectul impulsului perturbator
- Introducerea unei metode de detecție bazată pe descompunere wavelet și cumulare de decizie
- Introducerea unei metode de detecție bazată pe filtrare cu descompunere-recompunere wavelet și anularea parțială inversă a unor componente în spațiul descompus (denoising invers)
- Analiza comportării metodelor bazate pe wavelet la variația nivelului de auto-similaritate în seria de date și la aspectul impulsului perturbator
- Implementarea principiilor descrise în instrumente funcționale de colectare, extracție și vizualizare
- Aplicarea offline a mecanismelor descrise pe serii de date extrase din seturi binecunoscute, și confirmarea prin analiză detaliată a aplicabilității metodelor descrise în raport cu aceste date binecunoscute
- Investigarea de situații reale pentru confirmarea aplicabilității metodelor descrise în situații neprevăzute

1.2. Structura tezei

Dincolo de această introducere teza începe cu capitolul 2, „SISTEME DE DETECȚIE A INTRUZIUNILOR”, în care se trece în revistă problematica legată de sisteme de detecție a intruziunilor, de mecanisme existente de captură a traficului, de tehnici alternative de detecție bazată pe inspecție a pachetelor.

Se expun argumente pentru care am ales analiza traficului pe baze cantitative în detrimentul tehnicii de inspecție a pachetelor. Se prezintă pe scurt aspecte teoretice necesare înțelegerii deciziilor luate în implementarea componentelor sistemului NEAR. Se discută probleme legate de distribuția traficului, de instrumente ale analizei timp-frecvență bazate pe wavelet și de alte mecanisme de analiză și detecție propuse de alți autori.

In continuare se prezintă limitări pe care consider că le au alte abordări și puncte în care există posibilitatea de îmbunătățire apoi se face o descriere generală a unei posibile soluții bazate pe descompunerea traficului în fluxuri și se evaluează aplicabilitatea unei astfel de soluții.

In următorul capitol (3, „CAPTURA ŞI EXTRACŢIA SERIILOR DE DATE”) se tratează problematica legată de modul în care se transformă traficul de rețea în serii de date pregătite pentru a fi analizate. Pentru aceasta se definește un model al rețelei atât din punct de vedere al nivelelor cât și din punct de vedere al topologiei considerate. Se descriu și se motivează trei principii prin aplicarea cărora se poate crește eficiența detecției aplicate ulterior: îmbunătățirea SNR, principiul darknet/greyspace, principiul corelației inter-flux.

Capitolul continuă prin prezentarea formalizată a tipurilor de trafic considerate care conduce la o arhitectură specifică a componentei de extracție a seriilor de date. Se descrie metoda de identificare practică bazată pe port binecunoscut și se argumentează eficiența ei. Se dau exemple generice de aplicare a acestei metode.

Capitolul care urmează (4, „ANALIZA SERIILOR DE DATE”) se ocupă de descrierea și evaluarea mecanismelor folosite pentru analiza seriilor de date obținute cu metodele descrise mai sus. Pentru delimitarea problemei se enumeră tipurile de anomalii constatate și se discută cauzele potențiale ale apariției lor. Se dau exemple și se comentează asupra caracteristicilor particulare ale prezentării anomaliilor care poate avea relevanță asupra metodelor de detecție.

Se prezintă apoi metode de detecție pentru o clasă de anomalii și anume impulsuri singulare. Prima dată se tratează detecția simplă, bazată pe praguri și pe metode orientate către energia impulsului. Eficiența acestor metode este evaluată și susținută cu simulări în Matlab care se concentrează pe caracterul specific al seriilor considerate, dat de originea lor din algoritmii de descompunere a traficului în fluxuri.

Din rezultatele obținute la simularea detecției cu metode simple se susține rezultatul teoretic deja cunoscut care promovează o analiză timp-frecvență, așadar în continuare se prezintă două mecanisme de detecție care folosesc transformarea wavelet discretă și respectiv ideea de filtrare cu transformare wavelet. Cele două metode sunt evaluate folosind simulări în aceleași condiții ca și metodele simple. Se extrag concluzii cu privire la eficiența metodelor și a diverselor funcții wavelet la aplicarea în acest caz particular al seriilor obținute din trafic de rețea.

Obiectivul parțial al acestei teze este de a consolida pași spre un sistem integrat de detecție a intruziunilor (NEAR) util atât ca platformă de cercetare în domeniu cât și ca instrument practic folosit ca atare în rețele reale. Capitolul 5 („CONSIDERAȚII PRACTICE. SISTEMUL NEAR”) se ocupă de elemente componente

ale acestui sistem, implementarea lor practică și confirmarea ideilor teoretice din capitolele anterioare prin aplicarea pe cazuri reale.

Se prezintă sursele de date de test: evaluarea DARPA 1998, evaluarea DARPA 1999 și două exemple de rețele reale la care am avut acces. Se prezintă apoi structura propusă a sistemului NEAR și modul în care urmează să interacționeze elementele sale. Se descriu apoi mai detaliat două componente: agentul de captură și extracție NEAR-agent respectiv interfața grafică de prezentare și analiză ad-hoc NEAR-GUI. Cele două instrumente sunt apoi folosite pentru a analiza sursele de date DARPA, cu o prezentare orientată spre tipurile de atacuri detectabile folosind metodele deja descrise. In acest scop s-a parcurs integral setul de date DARPA offline cu NEAR-agent și am căutat apoi exemple relevante care să fie bine ilustrate în NEAR-GUI și corelate cu poziții documentate din setul original de date.

In încheierea capitolului am aplicat instrumentele de analiză pe trafic real capturat în timp real cu NEAR-agent și analizat apoi manual offline cu NEAR-GUI. Se prezintă câteva exemple care ilustrează potențialul celor două instrumente de a detecta anomalii sau pur-și-simplu de a confirma fenomene cunoscute teoretic dar pentru care nu am găsit până acum instrumente de vizualizare.

Lucrarea se încheie cu capitolul de concluzii și cu lista bibliografică.

2. SISTEME DE DETECȚIE A INTRUZIUNILOR

2.1. Monitorizarea rețelelor de calculatoare

De la apariția lor până azi rețelele de calculatoare au evoluat semnificativ în complexitate și importanță. Din această cauză monitorizarea automată sau semiautomată a modului în care operează a devenit o funcționalitate indispensabilă pentru rețelele mari și din ce în ce mai necesară pentru toate rețelele.

Pentru a pune în context prezentarea care urmează trebuie să facem o scurtă descriere informală a rețelei de calculatoare ca obiect de monitorizare. O rețea de calculatoare este compusă din *noduri* cu diverse roluri (client, server, nod intermediar) și *echipamente de comunicație* care conectează nodurile între ele. De asemenea, în marea majoritate a cazurilor rețeaua de calculatoare nu funcționează autonom, independent, izolat. Ea este în permanentă interacțiune cu *actori umani*. Interacțiunea între aceste componente diverse devine complexă și punctele de vedere din care se manifestă interesul pentru monitorizare sunt multiple.

Nodurile client sunt tipic folosite de utilizatori umani pentru a îndeplini funcții independente de rețeaua în sine (adica rețeaua și nodul client sunt doar instrumente de lucru). Gama de competență referitoare la calculatoare și înțelegerea specificului rețelelor variaza mult pentru această categorie de utilizatori. Din punctul lor de vedere importantă este disponibilitatea și calitatea serviciilor percepute global (în principal viteză de transfer și timp de răspuns). Există de asemenea utilizatori pentru care sunt importante și alte caracteristici cum ar fi confidențialitatea traficului, fiabilitatea sau controlul accesului la informație sau servicii. Trebuie să observăm ca majoritatea criteriilor după care judecăm diferitele elemente componente ale rețelei sunt raportate finalmente la experiența utilizatorului care se află în spatele nodurilor client. De remarcat natura specială a acestor noduri și anume că ele în principal emit solicitări care sunt rezolvate de alte noduri.

Nodurile server oferă servicii pentru alte noduri (deci indirect pentru utilizatori de la distanță) și interacțiunea lor directă cu actori umani este limitată la personalul de întreținere/administrare. In mod tipic accesul acestor actori de administrare la nodul server se face tot prin intermediul rețelei. Pentru nodurile server importantă este disponibilitatea serviciilor oferite și accesibilitatea lor în rețea. De asemenea prezintă importanță controlul accesului (atât la servicii și cât și la funcțiile de administrare) respectiv integritatea datelor pe care le procesează. De remarcat că uzual aceste noduri emit solicitări către alte noduri dar într-o măsură mai redusă comparativ cu volumul de solicitări pe care le rezolvă.

Nodurile intermediare conduc, controlează și transformă traficul între nodurile client și nodurile server. În această categorie intră de exemplu routere, firewall-uri, gateway-uri. Pentru aceste noduri importantă este disponibilitatea, performanța (exprimată în termeni de bandă generalizată adică spre exemplu throughput de pachete) dar și calitatea controlului aplicat. Nodurile intermediare nu interacționează decat cu actori umani cu rol de administrare. Accesul acestora la nod este similar cu cazul de nod server. Din punct de vedere al solicitărilor utilizatorilor aceste noduri sunt fie neutre fie au comportament de server.

Echipamentele de comunicație conduc trafic de nivel fizic sau nivel legătură de date între nodurile descrise mai sus. Performanța lor este legată de disponibilitate, fiabilitate, performanță dar pot sa manifeste și funcții de control al

accesului. Echipamentele de comunicație de calitate superioară pot fi administrate și urmărite de către actori umani cu rol de administrare, tot prin intermediul rețelei. Din punct de vedere al solicitărilor aceste noduri sunt uzual quasi-neutre (nu produc și nu rezolvă solicitări) dar pot adopta o poziție de server cu privire la traficul de monitorizare.

Monitorizarea rețelei presupune urmărirea unor parametri pentru anumite componente (noduri, echipamente) sau urmărirea și eventual analiza traficului constatat la nivelul componentelor și prezentarea informației într-o formă agregată spre utilizatorul uman sau extragerea unor concluzii automate. Monitorizarea poate fi făcută pentru scopurile unui utilizator izolat sau pentru scopul de urmărire a unei întregi rețele. Domeniul de interes pe care se concentrează monitorizarea poate fi de exemplu disponibilitatea serviciilor, previzionarea necesarului de capacitate de transport, detecția atacurilor informatice, detecția timpurie sau în timp real a defectelor în echipamente. Subiectul principal pe care ne vom concentra este detecția atacurilor dar păstrăm în atenție și posibilele aplicații ale unor soluții în celelalte domenii de interes ale monitorizării.

Sistemele de monitorizare se pot încadra în două categorii majore: sisteme care încearcă să detecteze structuri anormale direct în pachetele de date sau în relația dintre ele folosind experiența din atacuri anterioare (să le numim *sisteme bazate pe inspecția pachetelor*) și sisteme care consideră că evenimentele anormale din rețea se reflectă asupra traficului și încearcă să detecteze variații de aspect în mărimi sintetice care caracterizează traficul (să le numim *sisteme bazate pe analiza traficului*). Prima categorie poate fi mai precisă în detecție dar nu poate detecta decât situații anormale cunoscute. A doua categorie are o sarcină mult mai dificilă și poate să sufere de alarme false dar este principial capabilă să detecteze situații anormale noi. Interesul acestei lucrări se îndreaptă în principal spre a doua categorie şi reflectă preocuparea autorului pentru această abordare [1].

Ambele categorii de sisteme necesită capturarea în diverse forme a traficului. Sistemele bazate pe inspecția pachetelor capturează pachete întregi iar sistemele bazate pe analiza traficului capturează caracteristici *despre* trafic. In cazul sistemelor bazate pe analiza traficului după colectarea acestor caracteristici se aplică procesări de agregare, transformare și extracție a punctelor de interes.

2.1.1. Scurtă istorie

Monitorizarea rețelelor de calculatoare are radăcini istorice în monitorizarea rețelelor de telecomunicații generale. In spațiul mai recent odată cu apariția Internet-ului s-au desprins două direcții majore și anume monitorizarea orientată spre buna funcționare (disponibilitate, performanță) și monitorizarea orientată spre securitate (sisteme de detecție a intruziunilor - IDS).

Un exemplu (atât istoric cât și curent) este rețeaua ESnet care oferă conectivitate pentru diferite entități de cercetare în domeniul energiei din SUA. In această rețea s-a constituit încă din 1994 un grup de lucru pentru monitorizarea rețelei cu scop de diagnosticare și planning [2]. In spațiul aplicațiilor militare accentul cade pe securitate și preocuparea pentru monitorizare apare chiar mai devreme (1988, în urma viermelui Morris) în activități de cercetare la UC Davis [3] și ulterior în structuri organizatorice de tip CERT (Computer Emergency Response Team) aliniate la cerințele militare (AFCERT, 1992) [4].

Nevoia de monitorizare a rețelelor de calculatoare a condus de asemenea la produse software și hardware care acoperă diverse domenii de interes folosind multiple tehnici, inclusiv unele din cele descrise în detaliu în continuare. Există spre

exemplu sisteme de monitorizare comerciale orientate spre diagnostic sau planning [5], sisteme hardware de colectare de statistici despre trafic [6][7] și programe open-source pentru detecția intruziunilor [8] iar lista pachetelor software și a produselor hardware este realmente lungă și deschisă [9].

2.2. Mecanisme de captură a traficului

O parte importantă din sistemele de monitorizare analizează traficul constatat în rețea și extrag informații statistice sau caută anumite pattern-uri în interiorul pachetelor de date. Problema achiziţiei de informaţie referitoare la trafic la un nivel de detaliu care să permită o analiză eficientă este non-trivială. Ratele de bit ridicate disponibile actualmente, gradul de detaliere a parametrilor necesari, limitările inevitable hardware şi software impun aplicarea de tehnici speciale [10].

Fiecare sistem de operare capabil de comunicație în rețea oferă o unealtă proprie de captură și analiză a pachetelor în scop de diagnostic și analiză de protocol (Windows oferă Microsoft Network Monitor, Solaris oferă snoop, AIX oferă iptrace, Linux oferă tcpdump extins cu Ethereal/Wireshark). Aceste unelte specifice nu sunt însă potrivite pentru scopuri de monitorizare datorită modului lor de organizare orientat spre pachete izolate sau fragmente limitate de dialog orientat pe un singur protocol.

O evoluție semnificativă pentru domeniul de analiză automată a traficului a constituit-o decuplarea funcționalității de captură din utilitarul tcpdump într-o bibliotecă separată care se ocupa exact cu captura pachetelor, bibliotecă numită libpcap. Perechea tcpdump/libpcap dezvoltată inițial de Van Jacobson, Craig Leres și Steven McCanne a cunoscut un succes semnificativ, devenind în timp disponibilă pe multe sisteme de operare și un standard de facto pentru captura de pachete.

Evoluția performanțelor echipamentelor de rețea a adus însă în timp o problemă nouă pentru captură datorată abilității limitate a sistemelor de calcul generice de a prelua și procesa pachetele în timp real. Problema se manifestă atât la nivel hardware cât și la nivel software și se manifestă prin pierdere de pachete în sensul că nu toate pachetele care tranzitează linia de comunicație pot fi capturate.

Pentru a evita pierderea de pachete pe linia hardware e nevoie de o arhitectură specială capabilă să execute operații I/O la viteze susținute de ordin 1-10 Gbps (soluții FPGA, memorii FIFO) [11]. Pe linia software au fost necesare de exemplu adaptări ale structurilor din nucleul sistemului de operare pentru a evita copieri și comutări de context inutile (soluția PF_RING [12] și TNAPI [13]). Actualmente există soluții bazate pe sisteme de calcul uzuale echipate cu interfețe de rețea specializate, cu adaptări ale sistemului de operare și cu tehnici de paralelism multi-nucleu astfel încat să poată atinge captură și analiză continuă la trafic Ethernet de 1 Gbps și chiar 10 Gbps [14]

Există și posibilitatea de a delega funcțiile de captură și analiză statistică de prim nivel completamente către dispozitive hardware dedicate, asociate echipamentelor de rețea care pot de fapt să transporte volumele semnificative de trafic menționate mai sus. Un exemplu notabil este soluția NetFlow propusă inițial de CISCO. Echipamentele CISCO capabile de NetFlow echipate cu opțiunea de procesare hardware pot să extragă informație despre organizarea traficului în fluxuri de pachete la rata de trafic a interfeței (care poate fi zeci-sute de Gbps). In acest caz există totuși limitări intrinseci cu privire la numarul de fluxuri monitorizate simultan dar acesta este semnificativ de mare (echipamentele de categorie superioara suporta 250.000 fluxuri simultane, spre exemplu Cisco 7600 Series

Ethernet Services Plus 40-Gbps LineCard). Dincolo de aceste limite, colectarea fluxurilor poate fi făcută doar statistic prin eșantionarea traficului.

2.3. Detecția anomaliilor folosind inspecția pachetelor

Una din tehnicile larg folosite pentru detecția anomaliilor este inspecția pachetelor. Această metodă presupune că fiecare pachet este capturat și structura lui este analizată în detaliu. Captura și analiza se face fie în tranzit prin sistemul care face și monitorizarea, fie pe un sistem dedicat conectat la sursa de trafic printr-o interfață specială tip oglindă („*mirror*"). Sistemul de analiză aplică un set de reguli care încearcă să detecteze pachete anormale structural, pachete suspecte a fi parte dintr-un atac sau să colecteze informații statistice despre trafic și să detecteze anomalii.

Regulile pot fi simple („la offset X în interiorul pachetului se află valoarea Y") sau complicate („pachetul este parte dintr-o secvență de pachete care asamblate conțin textul NNNNN"). Sistemul aplică pe rând regulile și încearcă să găsească una sau mai multe care să fie îndeplinite. Dacă una din reguli este îndeplinită se declanșează o acțiune care poate fi discretă (de exemplu avansează un contor) până la agresivă (alertează administratorul sau blochează traficul sau ambele).

Evaluarea regulilor pentru fiecare pachet este o operație costisitoare în termeni de putere de procesare și de timp. Depășirea capacității de procesare conduce la riscul de ne-evaluare a unor pachete deci la riscul de nedetectare a anomaliilor căutate. Sistemele care folosesc metoda de inspecție a pachetelor trebuie să minimizeze acest risc prin formularea și aplicarea eficientă a regulilor sau prin tehnici de paralelizare.

Formularea setului de reguli este importantă pentru eficiența detecției. Fiecare regulă este destinată să detecteze un anumit tip de atac sau situație anormală. In timp apar atacuri noi sau situații anormale noi care necesită adaugarea unor noi reguli. Din această cauză există cel puțin două probleme practice semnificative pentru un sistem de detecție a anomaliilor care folosește inspecția pachetelor: 1. regulile trebuie întreținute și 2. volumul de reguli crește în permanență. In plus, un sistem care folosește inspecția pachetelor cu reguli poate detecta doar anomaliile și atacurile deja cunoscute. Trebuie să observăm de asemenea că actualmente există o tendinţă semnificativă de a proteja prin criptare o mare parte din volumul de trafic [15], astfel încât analiza conținutului dincolo de nivelul superficial al protocoalelor de rețea devine imposibilă.

Cel mai cunoscut sistem de detecție a intruziunilor care se bazează pe inspecția pachetelor este probabil Snort. Snort a fost creat inițial în noiembrie 1998 de Martin Roesch [16] pentru a complementa domeniul de acoperire al altor sisteme de detecție (SHADOW, NFR) pentru cazul sistemelor de trafic relativ scăzut. Intre timp Snort a evoluat și actualmente este disponibil ca sistem IDS open-source dar există oferte comerciale de actualizare a regulilor de detecție similar cu modelul de actualizare al produselor antivirus.

Snort are multipli concurenți, fie cu origine complet independentă, fie derivați original din acesta. Aproximativ în aceeași perioada (1998-1999) a fost independent prezentat Bro [17]. Bro are o structură mai bine definită, organizată pe nivele captură-evenimente-analiză. Lucrarea originală care introduce Bro analizează și contextul în care functionează un monitor de rețea destinat detecției intruziunilor. Din categoria sistemelor derivate din Snort putem aminti Suricata care aduce analiză mai detaliată a traficului HTTP și paralelizare agresivă pentru a exploata

procesoarele moderne multi-core și respectiv BotHunter [18] care urmărește un nivel superior de abstractizare și anume secvențele de trafic induse de propagarea tipică a unui vierme-bot.

2.4. Extracția caracteristicilor de trafic pentru analiza

Traficul de rețea nu este amorf. Datorită organizării sale structurate este posibil să îl defalcăm în fluxuri de date folosind diverse criterii, la diferite nivele ale modelului de rețea. Detecția anomaliilor din rețea pe baza analizei seriilor de date presupune aplicarea unor metode de analiză de natură statistică asupra seriilor de date agregate colectate de agenți cu privire la fluxurile de date care se regăsesc în traficul de rețea. Scopul este detectarea unor modificări cum ar fi: evoluții anormale ale mărimilor colectate în domeniul timp, corelație sau lipsă de corelație anormale între fluxurile detectate, schimbări de rang între nodurile implicate. Aceste modificări pot să sugereze existența unor defecte sau apariția unor tipuri nedorite de trafic (atacuri informatice sau doar trafic de overhead datorat unor erori tranzitorii de configurare).

Separarea traficului în fluxuri, preprocesarea informației despre fluxuri pentru a produce serii de date și analiza lor efectivă sunt procese consumatoare de resurse. Pentru a face posibilă analiza s-au dezvoltat mai multe modele de extragere, colectare și analiză dar ideea de bază este că unul sau mai mulți agenți colectează și agregă informația, apoi un nivel superior continuă efortul de agregare și analiză. Modul practic în care se face acest lucru și scopul pe care se concentrează fac diferența între metodele descrise în continuare.

Prima metodă larg folosită pentru analiza traficului orientată spre fluxuri a fost RMON (Remote network MONitoring). RMON a fost introdus în 1991 ca o extensie a SNMP MIB (Management Information Base) prin intermediul RFC 1271 [19]. Elementele originale de monitorizare din SNMP permit doar o monitorizare brută la nivel de interfață. RMON introduce detaliere a traficului pe dimensiunea timp, alarme la depășirea unor limite prestabilite, defalcarea traficului pe noduri descoperite, liste ordonate cu noduri pe criterii de trafic, matrice de conversație host-host, filtre de pachete, captură de pachete, jurnalizare de evenimente. Facilitatea de matrice de conversații este de fapt un tip de analiză pe fluxuri.

Unul din principiile notabile care stă la baza RMON (indus de fapt de descendența sa din SNMP) este faptul că agentul RMON trebuie să poată opera autonom, chiar fără existența unei stații de management. Setul complet de funcționalitate enunțat mai sus permite ca una sau mai multe stații de gestiune a monitorizării să interacționeze cu multipli agenți de monitorizare prezenți în echipamentele de rețea și care implementează RMON. Implementarea completă a unui agent este costisitoare însă în termeni de resurse așa încat multe implementări sunt doar parțiale, lucru permis de specificațiile RMON. Definiția MIB RMON a fost amendată de-a lungul timpului de RFC1757 (versiunea finală) și RFC2819 (reformularea pe bază de SMIv2) și RMON este prezent frecvent în echipamentele de tip switch cu capabilitate de management la distanță.

Standardul RMON acoperă nivelele fizic și legatură de date pentru rețele Ethernet și Token Ring. Pentru a putea extinde monitorizarea și la nivelele ierarhice superioare în 1997 a fost definit RMON2 prin intermediul RFC2021 [20], cu primitive de operare destinate monitorizării de nivel rețea (3) și aplicație. In terminologia RMON2 aplicație înseamnă de fapt orice nivel deasupra nivelului rețea, inclusiv nivelul transport. RMON2 se ocupă de aceleași aspecte ca și RMON dar adaugă

tratarea de noduri la nivel rețea și aplicație, matrici de conectare între astfel de noduri, adresare și filtrare pe protocoale.

O tehnică înrudită de monitorizare a traficului de rețea este NetFlow, introdusă de CISCO aproximativ în 1990 pe echipamentele proprii. Spre deosebire de RMON care lucrează la nivel fizic și legătură de date, respectiv RMON2 care urmărește și nivelele superioare, NetFlow face strict analiza fluxurilor constituite ca urmare a dialogurilor client-server în rețea derulate peste protocol IP. Pachetele TCP și/sau UDP care trec prin interfața urmărită sunt analizate sumar și se face identificarea pe bază de 7 atribute: host/port sursă, host/port destinație, protocol nivel rețea, clasă de serviciu, interfață. Pachetele sunt încadrate în fluxuri de date identificate pe baza acestor criterii și prezența lor actualizează contoare de pachete și octeți. La terminarea conexiunii (detectată pe bază de flag TCP-FIN) sau după un timeout fluxul este considerat închis și informația este exportată spre un nod în care se face colectare și analiză. Datorită acestui comportament de agregare per conexiune analiza fluxurilor se plasează într-o perspectivă mai apropiată de nivelul sesiune. Diferența semnificativă față de RMON este că NetFlow contează explicit pe prezența unui nod de colectare spre care se descarcă informația despre fluxuri.

Versiunile ulterioare de NetFlow au extins gama de informație procesată și colectează informații suplimentare (de exemplu număr AS sursă, număr AS destinație, nexthop IP, timestamps, TCP flags). Aceste câmpuri pot fi exportate împreună cu datele originale de flow [6]. Pe baza acestor informații suplimentare sistemele de analiză pot să ia decizii cu granularitate și acurateţe sporite.

NetFlow a cunoscut versiuni interne succesive până în 2004 (versiunea 9). In paralel, IETF a constituit un comitet pentru dezvoltarea unui standard de colectare a fluxurilor de rețea (2001) care a preluat conceptele NetFlow din versiunea 9 și le-a ajustat spre analiză și standardizare în RFC3954 [21]. Ca urmare a acestui demers în 2008 s-a definit IPFIX (IP Flow Information eXport) [22][23][24] cunoscut si ca NetFlow versiunea 10.

Tot la nivel de agregare a informației de trafic în fluxuri funcționează și principiul de colectare sFlow descris original de InMon Corporation in RFC3176 [25]. Principiul sFlow presupune eșantionarea traficului și colectarea selectivă doar a antetului unor pachete. Aceste antete sunt agregate în pachete UDP și trimise spre noduri colectoare pentru stocare și analiză. Achiziția se face tipic în echipamente de nivel 2 (switch-uri) și elementele colectate sunt configurabile. Din această cauză sFlow poate fi aplicat și pentru trafic non-IP, similar cu RMON. Diferențele majore sunt eșantionarea și descărcarea imediată a datelor către noduri colectoare, ceea ce permite operarea la rate de bit mult mai mari fără presiune semnificativă asupra echipamentelor de rețea. Principiul de eșantionare însă reduce acuratețea și introduce la limită posibilitatea obținerii de informații denaturate.

2.5. Analiza caracteristicilor de trafic

Informațiile referitoare la un anume flux de date se reduc finalmente la o serie de valori desfășurate peste domeniul timp. Putem să luăm uzual în considerare numărul de pachete pe unitatea de timp sau numărul de octeți pe unitatea de timp dar pot exista scheme în care analizăm numărul de fluxuri sau numărul de noduri implicate. De remarcat că nu toate metodele de agregare sursă oferă același gen de informație. Citirea periodică SNMP sau citirea RMON sau agregarea sFlow pot să ofere valori eșantionate uniform. Pentru NetFlow însă ciclul de viață și implicit de

raportare a fluxurilor este decis nu numai de expirarea periodică ci și de terminarea explicită a fluxului, ceea ce este un eveniment asincron cu raportare sincronă.

2.5.1. Distribuția traficului de rețea

Nevoia de estimare a caracteristicilor traficului de reţea în vederea dimensionării elementelor componente este evidentă şi iniţial tehnicile de estimare aplicate au fost preluate din experienţa existentă cu sistemele de telefonie clasică. In scurt timp (aprox. 1994) studiile au arătat că traficul de reţea constatat nu respectă distribuţia Poisson care a fost aplicabilă traficului din centralele telefonice ci mai degrabă are o caracteristică auto-similară [26]. Această constatare are implicaţii în domeniul predicţiei volumului de trafic cu scop de planificare şi cu scop de control al congestiei dar constatarea a deschis drum şi pentru o metodă nouă de apreciere calitativă a traficului cu implicaţii în detecţia anomaliilor din reţele.

Seriile cu distribuţie auto-similară sunt din punct de vedere strict matematic o sub-clasă a seriilor cu dependenţă pe durată lungă (LRD-Long Range Dependent) [27] dar pentru scopurile practice ale acestei prezentări cele două pot fi considerate ca echivalente. In condiţiile în care considerăm o serie $Y = \{Y(k)\}_{k \in Z}$ putem să construim versiuni agregate ale aceleiaşi serii prin însumarea repetată în blocuri $Y_n(k) = Y^{(n)}(k) = \sum_{i=1}^{n} Y(nk+i)$. De observat că însumarea are o caracteristică de scalare în timp de tip „zoom out”. Dacă seria originală are o distribuţie auto-similară atunci toate seriile astfel obţinute au aceeaşi distribuţie în limita unei constante

$$\left\{Y_n^{(m)}(k), k \in \mathbf{Z}\right\} =_d \left\{m^H Y_n(k), k \in \mathbf{Z}\right\}\Big|_{\forall m \in \mathbf{N}, n \to \infty} \quad (1)$$

In relaţia de mai sus H este parametrul Hurst (exponentul Hurst) şi indică gradul de auto-similaritate sau dependenţa pe durată lungă. Valoarea lui H = 0.5 indică o serie care prezintă caracteristici stricte de proces aleator de tipul mişcare Browniană. Valori ale parametrului 0.5 < H < 1 indică o distribuţie auto-similară, cu grad crescând de similaritate pe măsură ce H se apropie de 1. Valorile 0 < H < 0.5 indică un proces anti-persistent. Intr-o serie care are H > 0.5 prezența unei creșteri între două valori succesive este o indicație că următoarea treaptă va fi tot crescătoare. Intr-o serie care are H < 0.5 prezența unei creșteri între două valori succesive este o indicație că următoarea variație va fi descrescătoare. De observat că parametrul Hurst nu poate fi calculat ci doar estimat și estimarea sa este impactată de erori dificil de calculat.

Procesele care generează serii cu comportament de auto-similaritate indică faptul că un eveniment produs la momentul *k* are influență asupra valorii constatate peste *n* pași succesivi și această influență scade lent cu creșterea lui *n*. Procesul prezintă o dependență „long-tail” spre deosebire de cazul proceselor cu distribuție aleatoare la care influența scade exponențial cu creșterea lui *n*.

O proprietate semnificativă a seriilor auto-similare este cea legată de însumarea seriilor din această categorie. Seriile generate de procese stochastice prezintă o tendință de „nivelare a zgomotului” prin insumare sau prin scalare în timp. Seriile auto-similare își păstrează caracteristicile indiferent de scara de agregare în timp și indiferent de faptul că sunt compuse din suma a mai multe serii afluente. Proprietatea este importantă la studiul traficului de rețea unde traficul

agregat din multiple surse este constatat (și trebuie transportat, analizat, controlat) la granița între rețele.

In urma acestei caracterizări a traficului de rețea ca un proces auto-similar se poate face presupunerea că parametrul Hurst este caracteristic unei anumite agregari de trafic tipic pentru o anumită locație, cel puțin pentru o anumita perioadă finită de timp. Modificarea acestui parametru poate indica o anomalie deci tehnicile de estimare trebuie să poată detecta eficient modificarea parametrului și să localizeze cât mai exact în timp momentul modificării.

Trebuie să observăm că există și păreri diferite cu privire la natura auto-similară a traficului de rețea. Unele studii [28] arată că specificul protocoalelor de rețea și mai exact al protocoalelor care fac retransmisie și în special cele care rezolvă congestia în medii cu coliziune pot să inducă un caracter auto-similar care nu este intrinsec traficului util transportat. In evaluarile noastre ulterioare vom ține seamă și de acest punct de vedere.

2.5.2. Tehnici de detecție a anomaliilor

In condițiile în care traficul de rețea de la momentul *n* are un grad semnificativ de corelație cu traficul anterior de la momentul *n-k* se poate construi o metodă de detecție a anomaliilor bazată pe predicție și comparație [29]. Traficul se consideră că respectă un model oarecare de dependență a valorii *n* de valorile anterioare. Pe baza modelului considerat se calculează o predicție a valorii la *n*+1 și un interval de confidență. Dacă valoarea efectiv constatată la momentul *n*+1 cade în afara intervalului de confidență se înregistrează o anomalie. De remarcat că algoritmul descris mai sus este doar un principiu. O implementare realistă conține detalii suplimentare. De exemplu declararea unei anomalii se face doar după o secvență de valori ieșite din intervalul de confidență. De asemenea, prezența unei anomalii puternice va influența predicțiile următoare și va altera nivelul de bază (nivelul considerat normal).

Cel mai simplu model (dar și mai puțin conform cu realitatea, așa cum am văzut mai sus) este dependența cu scădere exponențială a influenței cu distanța (deci presupunerea unei distribuții Poisson). O variantă ceva mai realistă este modelul de predicție Holt-Winters care consideră că traficul are trei componente: o componentă de bază, o componentă cu tendință liniară, o componentă cu tendință periodică. In fapt, cele trei componente încearcă să modeleze comportamentul de lungă, medie si scurtă durată a traficului.

Defectul unei astfel de abordări este că încearcă să captureze dimensiuni diferite prin declarația fixă a unui model care încearcă să se adapteze la mai multe scări de timp. Așa cum am văzut mai sus, nici presupunerea că traficul poate fi modelat cu o sumă de comportamente gaussiane nu este suficient de potrivită cu structura traficului de rețea. Din fericire, structura unei serii auto-similare poate fi analizată și sintetizată eficient în scop de predicție folosind transformarea wavelet. In [30] Reidi et.al. arată că traficul de rețea poate fi simulat folosind o construcție pe bază de transformare wavelet dar scopul lor este construcția unui model de trafic realist pentru estimarea încărcării de rețea (evaluarea cozilor). Acest lucru este normal pentru că studiul lor datează din 2000, perioadă în care interesul studiilor era îndreptat în particular spre latura de simulare a traficului în scop de dimensionare. Ideea de a aplica transformarea wavelet pentru a studia eficient caracteristicile unei serii auto-similare rămâne valabilă și își găsește aplicare în domeniul detecției de anomalii în lucrarea lui Barford et.al. [31].

2.5.3. Transformarea wavelet – unealtă de analiză traficului

Pentru a putea continua discuția referitoare la analiza traficului cu tehnici multirezoluție trebuie să trecem în revistă principiile care stau la baza transformării wavelet.

Așa cum am evidențiat mai sus, caracteristicile despre trafic sunt reprezentate ca serii de valori în timp. Din punct de vedere al analizei putem să facem (deocamdată) abstracție de natura caracteristicii și să considerăm aceste serii ca un semnal generic. Ne interesează să identificăm în acest semnal componente de diferite frecvențe (corespunzătoare unor variații mai rapide sau mai lente a mărimii urmărite) și în același timp să plasăm cât mai exact în timp apariția acestor componente.

Descompunerea pe baza transformării Fourier poate să identifice componentele de diferite frecvențe dar nu poate să poziționeze apariția acestora în timp. Baza exponențială complexă folosită de transformarea Fourier este definită pe intervalul (-∞,∞) deci localizarea în timp este pierdută. O variantă a transformării Fourier care este capabilă să dea o poziționare a componentelor în domeniul timp este transformarea STFT propusă de Gabor. Transformarea STFT este de fapt o grupare de transformări Fourier executate pe segmente ale semnalului complet. Rezultatul este plasat într-un plan timp-frecvență sub forma unor petice dreptunghiulare de dimensiune constantă. Din variația dimensiunii segmentelor se poate obține o precizie mai mare în domeniul timp sau în domeniul frecvență dar există o limită superioară (similară cu principiul lui Heisenberg) care ne împiedică să avem rezoluție maximă atât în domeniul timp cât și în domeniul frecvență [32][33].

Dacă preluăm însă din transformarea Fourier ideea de convoluție care evidențiază prezența unei anumite componente în semnalul original și folosim drept bază de descompunere un set de funcții care să acopere spațiul timp-frecvență într-un mod neregulat obținem ideea de bază a transformării wavelet. O funcție de bază (*mother wavelet*, notată cu ψ) este scalată și translatată în timp pentru a produce setul de funcții care constituie baza de descompunere. Transformarea obținută este transformarea wavelet continuă (CWT) descrisă de relația

$$C_x(a,b) = \frac{1}{\sqrt{a}} \int_{\mathbf{R}} x(t)\psi(\frac{t-b}{a})dt, a \in \mathbf{R}_+ - 0, b \in \mathbf{R} \qquad (2)$$

unde $x(t)$ este funcția pe care o analizăm, a este factorul de scară și b este factorul de translație.

Ca să putem face integrarea funcția generatoare ψ trebuie să satisfacă o condiție de bază și anume să aibă media zero. Asta o face să fie oscilatorie și de aici vine denumirea transformării (wavelet – undișoară). Pentru a avea o rezoluție bună în timp și în frecvență funcția generatoare este preferabil să aibă majoritatea energiei concentrată într-o bandă limitată de frecvențe și un interval limitat de timp. Chiar dacă funcția nu este compactă (nenulă într-un interval limitat de timp) ea este tipic foarte aproape de zero în afara unui interval limitat de timp. In aceste condiții funcția are un comportament de filtru trece-bandă ceea ce justifică eficiența analizei.

Prima funcție de aceste gen a fost propusă de Haar în 1909 pentru analiză timp-frecvență dar formularea curentă a transformării datează din 1984 (Morlet și Grossmann). In fapt domeniul transformării wavelet și a problemelor matematice asociate a fost intens studiat începând cu cea de-a doua jumătate a secolului XX și

au fost dezvoltate aplicații în foarte multe domenii ale științei. Notabilă pentru interesul nostru este existența a mai multor funcții de bază descoperite de diverși cercetători în domeniu (Morlet, Meyer, Daubechies, Mallat, Coifman). Fiecare din aceste funcții de bază prezintă carcterisitici particulare care le pot face mai potrivite sau mai puțin potrivite pentru analiza diverselor procese.

In sens strict matematic setul de funcții obținut din funcția de bază ar trebui obținut prin scalare pe un numar infinit de nivele și translatare pe toată axa reală. Seriile pe care dorim să le analizăm sunt de fapt eșantioane ale unui semnal și procesările programatice pe care le putem face trebuie să se limiteze la un număr finit de scale *a* respectiv la un număr finit de translații *b*. Din această cauză o implementare a transformării va fi în mod necesar limitată la un număr finit de pași în timp și în scalare (intuitiv inversul frecvenței). In acest caz are sens să redefinim transformarea în termeni discreți.

Pentru a defini transformarea wavelet discretă (DWT) vom avea nevoie de asemenea de un set de funcții de analiză dar acestea sunt definite în termeni discreți față de funcția generatoare de bază $\psi(t)$. Obținem așadar o familie de funcții după scara j și translația k:

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k), \quad j \in \mathbf{N} - 0, k \in \mathbf{Z} \tag{3}$$

Se observă că în cazul familiei de funcții de mai sus scalarea în timp se face după o serie de puteri a lui 2 așa încât în fereastra de timp a unei funcții de scară j încap exact două funcții de scară imediat inferioară j-1. Această alegere nu este singura posibilă dar permite unele optimizări în algoritmul de implementare. Dacă setul de funcții este o bază ortonormală pentru spațiul funcțiilor $g(t)$ pe care le analizăm (funcții de pătrat integrabil) atunci putem să găsim un set de coeficienți $d_{j,k}(g)$ care să satisfacă relația de descompunere

$$g(t) = \sum_{j \in \mathbf{N}^*} \sum_{k \in \mathbf{Z}} d_{j,k}(g)\psi_{j,k}(t) \tag{4}$$

Aceşti coeficienţi sunt coeficienţii de transformare wavelet discretă. Din condiţia că funcţia analizată $g(t)$ este reprezentată de seria noastră care se întinde pe un interval finit rezultă că scara j este de fapt limitată superior la J după relația $L >= 2^J$ unde L este lungimea seriei considerate.

Scopul nostru este însă descompunerea semnalului reprezentat de seria originală în sumă de componente pe benzi de frecvenţă. Deoarece descompunerea este finită (până la pasul J) pentru a obţine o versiune practic utilizabilă a transformatei avem nevoie de o funcţie suplimentară, funcţia de scalare $\varphi(t)$. Această funcție este pereche cu funcția generatoare $\psi(t)$ și poate da naștere unei familii similare de funcții Prin introducerea acestei funcţii înlocuim partea din suma cu $j>J$ cu o sumă de funcții $\varphi_{J,k}(t)$ ponderate cu un set de coeficienți $a_{J,k}(g)$.

$$g(t) = \sum_k a_{J,k}(g)\varphi_{J,k}(t) + \sum_{j=1}^{J} \sum_k d_{j,k}(g)\psi_{j,k}(t) \tag{5}$$

Primul termen reprezintă o aproximare a semnalului original la nivel de scară J iar al doilea termen este o sumă de semnale-detaliu extrase din semnalul

original, pentru toate scările de la 1 la *J*. Coeficienții $a_{J,k}(g)$ descriu aproximarea de cea mai joasă frecvență a semnalului. Deoarece funcțiile de bază sunt definite pe interval compact în realitate *k* nu trebuie să varieze decât pe un interval limitat. Dacă luăm în considerare și decimarea descrisă în continuare, la nivelul *J* (cel mai adânc) există un singur coeficient de aproximare relevant și anume chiar valoarea medie a semnalului original.

Transformarea DWT poate fi așadar privită ca o descompunere succesivă a semnalului original în componenta detaliu și componenta aproximare, urmată de o nouă descompunere a aproximării in detaliu de nivel j+1 și aproximare de nivel j+1. Deoarece la fiecare pas banda efectivă a semnalului rămas este redusă la jumătate numărul de eșantioane poate fi și el redus la jumătate astfel încât la fiecare pas numărul de coeficienți și de eșantioane se reduce succesiv. Structura acestei transformări este consistentă cu principiul „la frecvențe mari (scară mică) avem un număr mare de coeficienți deci poziționare precisă în timp" si „la frecvențe mici (scară mare) avem un număr mic de coeficienți deci poziționare mai puțin precisă în timp". Pe linia frecvență însă benzile de frecvență sunt înjumătățite la fiecare pas astfel încât rezoluția maximă se va obține la scara cea mai mare (deci la frecvența cea mai redusă).

Deși DWT are proprietăți bune de eficiență și poate fi un instrument util în analiza semnalelor cu caracteristici auto-similare există și dezavantaje induse tocmai de caracterul eficient al decimării introduse la fiecare pas de descompunere. Detecția caracteristicilor de tip LRD necesită o rezoluție sporită și la scări mari pentru că tocmai acolo se manifestă dependența de durată mare. Pentru a compensa lipsa de informație există variante ale transformării bazate pe descompunerea simetrică pe ambele ramuri în detaliu+aproximare (WPT – Wavelet Packet Transform), păstrarea numărului de eșantioane la fiecare pas (SWT – Stationary Wavelet Transform), întreteserea coeficienților într-o schemă DWT pseudo-complexă (DTWT – Dual Tree Wavelet Transform) [34] sau aplicarea transformării asupra unei versiuni complexe de semnal, adaptate din semnalul original (ADWT – Analytical Discrete Wavelet Transform) [37].

2.5.4. Metode existente de analiză a traficului

Pentru analiza seriei de valori în timp putem așadar să aplicăm o transformare wavelet potrivită și să obținem un set de coeficienți. Modul în care sunt exploatați acești coeficienți poate fi diferit și multiple abordări au fost încercate de-a lungul timpului.

Abordarea propusă de Barford et.al. [31] descompune seria și agregă seriile de coeficienți în trei benzi: high, medium, low. Valorile rezultate din agregare sunt analizate cu un algoritm de tip Holt-Winters. Benzile high si medium conduc la detecția anomaliilor de tip DoS sau atacuri de scurtă durată iar banda low permite detecția fenomenelor de lungă durată (flash-crowds, adică intensificări ale traficului de durată mai lungă cauzate de evenimente externe rețelei). De remarcat că în această lucrare apare o idee interesantă dar care nu face obiectul analizei autorilor. Ei sugerează că plasarea senzorului de colectare mai aproape de nodurile interne poate să mărească șansa de detecție a anomaliei dar și că în același timp acest lucru ar produce o variabilitate puternică a traficului constatat care mărește rata de falsuri pozitive. Se poate deduce că ar exista un punct optim în raportul între traficul normal și traficul anormal, punct optim în care detecția are șane maxime de funcționare fiabilă.

O altă abordare este cea propusă de Stoev et.al. care încearcă să identifice modificări în trafic prin evaluarea parametrului Hurst folosind spectrul wavelet [27]. Spectrul wavelet este construit prin estimarea energiei conținută în fiecare bandă (definită de scară) folosind suma pătratelor coeficienților corectată logaritmic. Setul de valori S_j (j=1..J) constituie spectrul wavelet ale semnalului analizat și se poate demonstra că la scări mari panta spectrului este chiar 2H-1. De fapt aceasta este o metodă simplă de estimare a parametrului Hurst, și autorii fac o comparație foarte favorabilă cu metoda clasică bazată pe estimator Whittle. In cazul traficului de rețea real însă, structura clară a spectrului wavelet este deformată datorită perturbațiilor (anomaliilor). Metoda nu permite identificarea temporală (dimensiunea temporală se pierde odata cu însumarea) dar poate constitui o bază de *detecție* a prezenței anomaliilor. Cuplată cu o tehnică de segmentare similară cu STFT poate să fie chiar o metodă de delimitare a anomaliilor în timp.

O altă metodă de detecție este cea propusă de Gao et.al. [35]. Presupunerea pe care se bazează este că anomalia în trafic are o distribuție detectabilă în domeniul frecvență în comparație cu traficul normal care are avea o distribuție mai uniformă. Pentru a putea detecta cu acuratețe anomalia e nevoie de rezoluție bună în frecvență dar tehnica DWT nu asigură rezoluție bună în frecvență decât la scări mari (frecvență joasă). Asta face ca anomaliile „rapide” sa fie greu de detectat. Tehnica propusă foloseste WPT cu o variantă accelerată de evaluare tip „sliding-window”. O data detectată creșterea unei benzi de frecvență peste pragul ales, se face o reconstrucție parțială numai din componenta respectivă pentru a regenera componenta anormală și a putea marca mai bine momentul de timp al acesteia.

Revenim cu seria de exemple de metode care folosesc transformatrea wavelet pentru analiza traficului la ideea de predicție-detecție propusă de Wei Lu și Ali Ghorbani [36]. Metoda este asemănătoare cea descrisă mai sus și aplicată direct pe un model simplu dar de data aceasta predicția se face pe setul de coeficienți rezultați din transformarea wavelet. Se folosesc doua secțiuni din setul de trafic considerat normal pentru pregătirea unui model de predicție ARX. Odată modelul de predicție pregătit, traficul suspect este supus predicției și se constată diferența față de traficul real. Evident că diferențe vor exista intotdeauna dar pentru detecția deviațiilor anormale se folosesc tehnici de recunoașterea formelor care identifică valorile de vârf (outliers). Tehnica poate fi aplicată și folosind alte tipuri de transformare [38].

De observat că la transformarea wavelet, în oricare din tehnicile de analiză de mai sus se poate folosi orice funcție generatoare. Aparent nu toate funcțiile au aceeași eficiență și apreciem că eficiența unei familii de funcții față de altă familie de funcții este dependentă de tehnica aplicată și de specificul traficului pe care îl analizăm Există unele păreri care susțin că funcțiile generatoare Coiflet și Paul au rezultate bune [39] dar și altele care constată că performanța depinde de tipul de date analizat și ca Haar/Daubechies1 are totuși cele mai bune performanțe per total [36].

Analiza traficului poate fi făcută și cu alte tehnici, independente de teoria procesării semnalelor și transformarea wavelet. Evident, au existat metode bazate pe prelucrări statistice înainte ca tehnicile de analiză derivate din transformarea wavelet să capete o largă răspândire. Nu detaliem și nici nu enumerăm aici aceste tehnici dar trebuie să remarcăm că ele există. O idee care merită însă menționată aici pentru că are legătură cu tehnicile propuse în continuare este descrisă de Kopylova et.al. [40]. Metoda propune evaluarea entropiei ca instrument de constatare a anomaliei. Traficul este caracterizat cu o matrice de conexiuni

sursă/destinație care are drept valori intensitatea traficului constatat. Structura internă a matricii și relația între matrici calculate la momente de timp consecutive poate fi evaluată calitativ și agregată întro valoare de tip entropie. Valori mari ale mărimii agregate indică un atac sau o anomalie. Dacă e să dăm o interpretare fizică a conceptului ar trebui să constatăm că traficul normal este „concentrat" iar traficul anormal (portscan, propagare de vierme/virus), este „împrăștiat" deci implică mai mulți parteneti simultani adică o entropie mai mare.

2.6. Soluţia propusă – analiza combinată și adaptivă a multiple caracteristici

2.6.1. Limitări în soluțiile și analizele anterioare

In urma evaluării tehnicilor descrise mai sus și datorită experienței practice în domeniu am ajuns la o serie de păreri personale care constituie baza pachetului de măsuri și soluții propuse în continuare.

In primul rând constat o separație semnificativă între lucrările cu fundament matematic și lucrările cu orientare practică. In fapt, puține lucrări publicate au o abordare eminamente practică asupra subiectului și însoțită în același timp și de noutate sau rezultate semnificative. Merită totuși amintite eforturile descrise de Maselli/Deri/Suin pentru construcția unui sistem real de detecție a anomaliilor la Universitatea din Pisa [41]. Pe de altă parte multe din lucrările care au o componentă extrem de interesantă din punct de vedere algoritmic sunt limitate în modul de raportare la realitatea practică pentru care este construită soluția.

O altă constatare este că există o propagare fără verificare a unor surse de informație relativ vechi, care conduc la definirea unor false probleme referitor la atacuri care nu mai constituie o amenințare de securitate. Este adevărat, capabilitatea principială de detecție este importantă și poate fi demonstrată chiar pe cazuri care nu mai sunt de actualitate dar anumite condiții tehnologice pur-și-simplu nu se mai aplică. Spre exemplu atacul PoD (Ping-of-Death) este un atac de reasamblare al unui singur pachet ICMP de dimensiune exagerat de mare. Este suficient un singur pachet malformat pentru a avea un atac de succes către o gazdă vulnerabilă. Toate sistemele de operare însă au fost de mult adaptate pentru a evita acest atac. Incidental însă ca principiu PoD este un exemplu valoros pentru o altă idee, dezvoltată mai jos. Datorită specificului său de un singur pachet dar cu o dimensiune anormal de mare atacul PoD va fi invizibil pe o caracteristică de tip „număr de pachete" în schimb va fi mai vizibil pe o caracteristică de tip „număr de octeți" și foarte vizibil pe o caracteristică „dimensiune maximă a pachetelor de date".

Un alt exemplu de atac învechit este atacul „smurf". Principiul său este de amplificare a numărului de pachete ICMP datorită efectului de ping broadcast. Din epoca apariției acestui tip de atac până azi s-au luat însă măsuri eficiente de blocare a răspunsului la ping broadcast atât la nivelul nodurilor simple cât și la nivelul routerelor care leagă între ele segmentele de rețea. Actualmente el are doar o importanță istorică. Incidental este și ușor de detectat datorită volumului ridicat de trafic și vitezei rapide de creștere a acestuia.

Tot pe această linie putem să remarcăm că evoluția rapidă a tehnologiei invalidează unele modele sau le face mai puțin interesante. Spre exemplu seturile de date DARPA presupun o structură de rețea bazată pe hub unde tot traficul este vizibil pentru toate nodurile. Nivelul de „zgomot" (trafic neinteresant) perceput de o

sondă de tip netflow plasată într-un nod simplu este semnificativ mai mare decât în cazul unei rețele tipice actuale bazate pe switch-uri.

Foarte multe din soluțiile propuse de-a lungul timpului se auto-limitează sever în faza de captură și agregare. Este de înțeles că sursa tipică de informație ușor accesibilă o constituie colectoarele de flux de date și primele mărimi interesante sunt numărul total de pachete sau numărul total de octeți dar detecția eficientă necesită aplicarea tehnicilor puternice de analiză descrise mai sus asupra celor mai potrivite mărimi pentru a evidenția anomaliile căutate. In acest sens abordarea multi-caracteristică este semnificativă. Spre exemplu mărimea caracteristică „număr de fluxuri pe unitatea de timp" este o caracteristică mai puternică pentru detecția de anomalii din categoria portscan decât „număr de pachete pe unitatea de timp". Chiar dacă este mai costisitor de colectat, avantajul de calitate trebuie luat în considerare.

Pentru a susține următoarea idee trebuie să fac o analogie. In efortul de a interacționa cu succes cu mediul care ne înconjoară ne folosim de toate simțurile. Multiple surse de informație sunt agregate și corelate pentru a avea cea mai bună imagine posibilă asupra lumii din jur. Eliminarea chiar și a unui singur simț limitează abilitățile noastre iar limitarea la un singur simț reduce aproape de zero abilitatea de a ne descurca în lumea înconjurătoare. Dacă preluăm această analogie devine natural să ne întrebăm „de ce să folosesc o singură caracteristică de trafic pentru a lua o decizie?". Putem să punem această întrebare la toate nivelele de abstractizare și să mergem de la „de ce să consider doar traficul IP?" până la „de ce să mă limitez doar la caracteristici ale traficului de rețea?". Analiza simultană a mai multor caracteristici, cu aceeași metodă sau cu metode concurente are trebui să conducă la rezultate mult mai bune atât din punct de vedere al sensibilității cât și din punct de vedere al calității alarmelor.

Așa cum menționam mai sus, în mod surprinzător multe soluții îngustează fără justificare sursa de informație. Susțin că trebuie analizat tot traficul, inclusiv cel non-IP, chiar dacă nu este de volum comparabil cu cel IP (sau tocmai pentru că nu este de volum comparabil cu cel IP). Susțin că ar trebui folosite multiple surse de informație (în măsura în care sunt disponibile), inclusiv non-rețea, cum ar fi temperatura nodului sau încărcarea procesorului. De asemenea, în măsura posibilităților, colectarea de caracteristici în multiple noduri și nu numai în nodurile de acces tip router poate să construiască o imagine mai bună. In acest sens o abordare de tip matrice de trafic între perechi sursa/destinație este valoroasă dar va necesita eforturi sporite de procesare.

O altă idee care consider că merită explorată este îmbunătățirea calității sursei primare de informație. Principiul este exact cel care a condus inițial la extragerea de fluxuri de dat din traficul global. Contoarele globale întreținute de sistemele de management SNMP au fost înlocuiți de informațiile mai detaliate din RMON și ulterior de informațiile și mai detaliate din NetFlow. Dacă analizăm semnalul generat de traficul global trebuie să detectăm un semnal care poate fi foarte slab (anomalia, atacul) îngropat în zgomot deosebit de intens (traficul normal). Dacă folosim instrumentele colectoare de caracteristici despre trafic pentru a analiza doar o parte din trafic putem să schimbăm raportul semnal-zgomot în favoarea noastră. Așadar dacă reușim să eliminăm traficul normal sau o parte semnificativă din el prin alegeri optimale înainte de analiză putem să îmbunătățim calitatea detecției. Principiul este de fapt folosit la soluțiile predicție/detecție dar pe o dimensiune globală.

Pe aceeași linie a îmbunătățirii raportului semnal-zgomot (chiar dacă nu este enunțată în acești termeni) se înscrie de fapt și o idee prezentată de Jain și Patil

[42]. Soluția prezentată de ei este o variație a ideii de „darknet” (în sens de subspațiu de adrese-capcană pentru trafic neautorizat) și presupune segregarea spațiului IP intern în adrese folosite și adrese nefolosite. Adresele nefolosite ar trebui să nu înregistreze trafic dar unele anomalii sunt distribuite peste tot spațiul de adrese și deci pot fi detectate mai ușor în domeniul celor temporar nefolosite. In cele ce urmează voi refolosi ideea de bază de tip „darknet” dar aplicată la spațiul porturilor (adrese de nivel transport) cu rolul de a imbunătăți reportul semnal-zgomot.

Din parcurgerea diferitelor soluții am mai desprins și alte idei care pot fi valorificate în tandem cu cele de mai sus. Este vorba de exemplu despre posibila aliniere specifică a distribuției dimensiunii pachetelor care se manifestă fie per aplicație fie per stil de exploatare a unui protocol. Distribuția statistică a dimensiunii pachetelor este folosită în unele abordări pentru detecția aplicațiilor care folosesc porturi atipice [43][44][45] dar poate fi și bază de analiză pentru detecția intruziunilor în condițiile în care distribuția se schimbă dacă traficul este anormal [46]. Analiza distribuției statistice a dimensiunii pachetelor poate fi un candidat serios la poziția de caracteristică meritorie.

In final, o observație pe care am găsit-o menționată în unele din lucrările menționate mai sus este confirmată de experiența practică: în anumite cazuri particulare dar nu neapărat rare traficul are o foarte puternică componentă periodică. Exemple de astfel de trafic pot fi multiple: colectarea periodică de valori din chiar agenții sistemelor de monitorizare, mecanismele de actualizare a rutelor, mecanismele de negociere și întreținere spanning-tree, procese de actualizare de nivel aplicație cu caracter periodic, actualizări de domenii DNS. Prezența corectă și neperturbată a componentelor de frecvență corectă în traficul constatat este o mărturie a funcționării normale a respectivului mecanism și poate fi foarte ușor detectată folosind tehnicile pe care le-am descris deja. Funcționarea anormală poate să însemne întreruperea periodicității sau înnecarea componentei periodice în trafic neașteptat, ambele fiind situații ușor de detectat.

2.6.2. Descriere generală a soluției propuse

Principiul nou pe care îl introduc în spiritul observațiilor de mai sus este descompunerea traficului pe dimensiunea port sursă - port destinație cu orientare client - server. Descompunerea uzuală în fluxuri identifică portul sursă și portul destinație care sunt parte din cheia de identificare a fluxului. Din păcate însă nu am găsit exemple de procesare care să acorde atenție apartenenței partenerilor de comunicație la o clasă sau alta (client sau server). Fluxul odată identificat cheia de 5 componente devine oarecum anonimă sau este exploatată pe linia adreselor de nod dar nu pe linia porturilor.

In fapt, traficul TCP și UDP (dar și alte tipuri de trafic) sunt aliniate principiului client-server. La nivel transport (in cazul TCP) sau la un nivel superior (în cazul UDP) se manifestă ideea de dialog: nodul client solicită o acțiune sau o informație, apoi nodul server răspunde cu informația solicitată sau confirmarea acțiunii. Secvența poate fi repetată sau poate fi mai complicată dar esența comportamentului este aceeași: se constituie o sesiune de comunicare asimetrică din punct de vedere al rolurilor asumate de parteneri.

In momentul când acceptăm acest model ca bază de analiză următoarea observație logică este că la primul contact clientul trebuie să cunoască portul server-ului (adresa SAP de nivel transport) și că pentru a asigura acest lucru aproape toate protocoalele uzuale folosesc porturi binecunoscute. Asta înseamnă că fluxurile pot fi

segregate după portul „server”. Merită să facem o observație: la prima vedere în modelul de captură pe fluxuri acesta este portul destinație dar în realitate pot exista excepții semnificative. NetFlow elimină din cache fluxurile pe bază de timeout și la recaptură e posibil să apară o inversiune (adică primul pachet să fie emis de server). Detecția portului server se poate face însă cu o rată bună de succes pe bază de porturi binecunoscute.

Colectarea datelor despre trafic se poate face în mai multe puncte. Presupunând că am ales un punct de colectare (adică o interfață a unui nod la nivelul căruia facem captură) se creează imediat o tăietură. Din perspectiva client-server rezultă că unele fluxuri au server-ul „la dreapta” punctului de captură și altele au server-ul „la stânga” punctului de captură. Dacă acest punct de captură este la granița între o rețea internă și Internet, segregarea asta se interpretează „clienți interni accesează servere externe” și „clienți externi accesează servere interne”. Dacă punctul de captură este pe interfața de acces la un nod de interes, segregarea se interpretează „trafic pentru care sunt client” respectiv „trafic pentru care sunt server”. Ambele exemple descriu situații importante din punct de vedere al administrării de rețea deci consider că abordarea poate fi utilă.

A doua componentă a analizei este maximizarea raportului semnal-zgomot pe baza segregării de fluxuri pe dimensiunea porturi-server și pe dimensiunea „dreapta” - „stânga”. In mod natural, traficul normal „spre dreapta” va avea anumite porturi-server preferate și invers. Marea masă a traficului este generată direct sau indirect de activitatea utilizatorilor și prezintă caracteristicile de auto-similaritate descrise mai sus. Dacă desprindem însă din traficul normal traficul intens spre porturile server preferate pe ambele direcții va rămâne trafic „excepțional” care se încadrează fie în categoria „activitate normală de fundal” și care de multe ori este periodică (deci ușor de recunoscut) sau anormală. In fapt, prin discriminarea aplicată am creat un echivalent al spațiului „darknet”[2] folosit deja ca tehnică de detecție și analiză a atacurilor de tip vierme/virus [47].

După aplicarea acestor tehnici rămâne în continuare trafic anonim, cel puțin din punct de vedere al criteriilor de mai sus. Acest trafic este încadrat în una din categoriile: trafic la care ambele porturi sunt porturi binecunoscute, trafic la care nici unul din porturi nu este incadrat ca port binecunoscut, trafic de altă categorie (care nu este TCP sau UDP deci nu cunoaște ideea de port). Exemple notabile ar putea fi traficul NetBIOS pe port 137 (care uzual are si sursa si destinația pe 137 și mai mult decât atât este uzual UDP deci nu se poate determina ușor server-ul) sau traficul de tip peer-to-peer care poate să se deruleze pe porturi arbitrare (non-standard). La categoria „trafic non-UDP și non-TCP” se încadrează de exemplu traficul ICMP (care l-am găsit luat în considerare în [36]) și traficul ARP care îl consider foarte important dar pe care nu l-am găsit analizat deși este recunoscut ca vector semnificativ de atac.

După separarea făcută mai sus se pot colecta caracteristici de trafic pe grupuri de fluxuri, alese astfel încât să se maximizeze șansele de detecție. Procesul de grupare este necesar pentru a reduce numărul seriilor de valori care trebuie

[2] Termenul de „darknet” este folosit de Team Cymru. Aceeași idee se mai întâlnește sub numele de „Network Telescope” (CAIDA) sau „Internet Motion Sensor” (University of Michigan). Am folosit termenul de „darknet” pentru că sugerează cel mai bine ideea pe care o exploatez dar trebuie remarcat că are și o a doua conotație, mai larg răspândită, legată de Digital Rights Management și datorată unui articol originar de la Microsoft dar care nu are legătură cu scopurile acestei lucrări.

analizate și modul în care se grupează categoriile de trafic descrise mai sus. Pentru fiecare grupare se colectează una sau mai multe caracteristici, nu neapărat aceleași pentru toate grupările. Recunosc că mecanismul de grupare este o componentă sensibilă a soluției propuse și că o abordare complet automată este dificil de obținut dar cred că pot exista tehnici de tip inteligență artificială și sistem expert care pot fi folosite aici pentru a obține configurații rezonabile.

In măsura în care există disponibilitatea tehnologică (vezi mai jos ideea de agenți multipli) odată cu colectarea acestor mărimi se colectează și mărimi non-rețea (temperatura CPU sau direct grad de încărcare CPU). Seriile rezultate din aceste mărimi pot fi folosite direct la analiza automată sau pot fi doar reținute ca sprijin în analiza umană post-mortem.

Ultimul nivel de perfecționare este colectarea multiplă. Aceleași reguli de colectare și agregare se aplică la nivelul mai multor agenți de colectare care furnizează date unui sistem central de evaluare. La nivel minim absolut agenții ar trebui să fie plasați în nodurile de graniță ale rețelei dar o configurație mai eficientă necesită colectarea și la nivelul unor noduri-cheie. Prin nod-cheie se înțelege pe de o parte un nod important în funcționarea rețelei (server intern de fișiere, server de aplicație, server DNS) dar în egală măsură și un nod absolut „inutil” (în baza principiului de detecție darknet redus la un singur nod).

Faza finală a analizei presupune aplicarea tehnicilor de analiză a semnalelor pe seriile de valori obținute din colectarea descrisă mai sus. Estimez că o anomalie reală se va regăsi în cel puțin două serii aflate la nivelul unor noduri diferite deci se poate folosi acest lucru ca tehnică de reducere a falsurilor pozitive.

2.6.3. Aplicabilitate

O soluție de tipul celei descrise poate fi implementată teoretic în orice rețea. Există o serie de presupuneri care trebuie subliniate pentru a delimita posibila aplicabilitate dar care se verifică pe multe rețele existente.

- Traficul este majoritar de tip TCP și UDP cu o distribuție tipică și relativ stabilă ca servicii folosite
- Traficul este transportat pe suport Ethernet
- Este acceptabilă instalarea de agenți de captură pe o parte din nodurile rețelei

Presupunerea referitoare la Ethernet este mai puțin importantă dar o facem pentru că Ethernet este un mediu frecvent întâlnit și permite detecția unei clase semnificative de atacuri care încep cu impersonare ARP. Principiul funcționează de fapt și fără ea.

Presupunerea referitoare la agenți poate fi relaxată și o parte din date pot fi colectate de la surse existente de tip NetFlow sau sFlow (routere sau switch-uri). Pentru că nu am avut acces la astfel de echipamente aflate în funcțiune într-un mediu realist de test am folosit implementări de agenți specializați.

Ca să demonstrez viabilitatea principiului am implementat un agent de colectare (instalat în multiple instanțe) și o platformă de agregare și analiză a datelor cu tehnici din familia procesărilor de semnal. Testarea lor a fost făcută în condiții de trafic real pe doua segmente Ethernet de dimensiuni moderate și am făcut unele evaluări pe serii de date derivate din seturile de date DARPA.

3. CAPTURA ŞI EXTRACŢIA SERIILOR DE DATE

Sursa principală de informație pentru detecția intruziunilor și anomaliilor în modelul propus de această lucrare este traficul de rețea, chiar dacă principiile generale se pot aplica şi intenţionăm să le aplicăm şi pentru alte tipuri de surse de date.

Aşa cum am prezentat anterior, detecţia anomaliilor se face prin aplicarea unor tehnici de procesare a semnalelor asupra unor serii de date reprezentând evoluția în timp a unor mărimi caracteristice. In cele ce urmează vom prezenta modelul simplificat al rețelei pe care o supraveghem, vom descrie principiile pe care intenționăm să le aplicăm pentru a îmbunătăți performanța sistemului de detecție și vom descrie modul în care se pot transforma aceste principii într-un sistem de achiziție a seriilor de date destinate procesului de analiză. Aceste principii şi mecanismele de aplicare a lor au fost introduse de autor în [48].

3.1. Modelul rețelei supravegheate

Rețeaua pe care o supraveghem este compusă din noduri care sunt conectate între ele prin intermediul unei rețele capabile să transporte trafic IP sau chiar de altă natură. Nodurile se separă în două categorii:

- noduri terminale
- noduri releu

Nodurile terminale nu efectuează trafic decât pentru uz propriu, dar pot sa facă asta în calitate de consumator sau furnizor de servicii (adică în calitate de client sau de server într-un model client-server). Nodurile terminale mai pot de asemenea să participe pasiv sau activ la trafic secundar, legat de mecanisme interne de funcționare a rețelei și care nu are relevanță pentru entitățile de nivel superior care folosesc aceste noduri pentru a transporta informație. Nodurile terminale sunt conectate la rețea printr-o singură interfață. Există noduri terminale multi-homed (conectate simultan la două sau mai multe rețele) dar atât timp cât nu au rolul de releu le vom considera ca o colecție de noduri terminale simple.

Nodurile releu efectuează trafic pentru a deservi în ultimă instanță alte noduri terminale cu care sunt în directă legătură sau care sunt indirect conectate prin intermediul altor noduri releu. Și aceste noduri participă pasiv sau activ la trafic secundar, la fel ca și nodurile terminale. Prin natura lor nodurile releu sunt conectate simultan la mai multe rețele prin multiple interfețe.

In fapt, nodurile releu sunt de obicei și noduri terminale, pentru că uzual există servicii oferite de aceste noduri releu și la rândul lor nodurile releu sunt consumatoare de servicii.

Din punct de vedere funcțional, această discriminare este reprezentată în Figura 1, conform modelului clasic ISO-OSI [49][50].

Se observă din figură cum caracterul de terminal sau releu poate să difere după nivelul de abstractizare la care ne concentrăm atenția (vezi nivelele diferite de releu în accepțiunea arhitecturală ISO-OSI). Nu ne vom ocupa în mod special cu această diferențiere și ne vom concentra pe nivelul 3 (nivel rețea sau IP), adică releele despre care discutăm în continuare sunt routere.

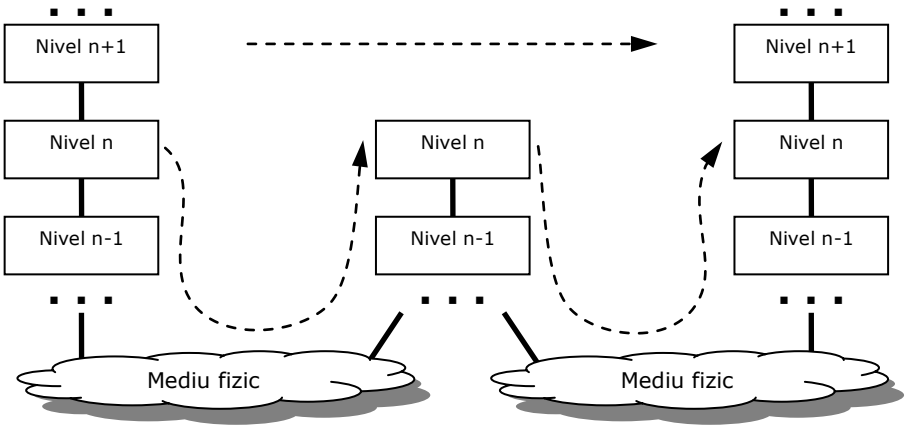


Figura 1 Lanț de comunicare între două noduri terminale care trece printr-un nod releu.

Structura descrisă mai sus este ilustrată topologic în Figura 2 care prezintă un model cu multiple rețele locale conectate între ele prin noduri releu și care conțien fiecare noduri terminale. Calificativul de local aplicat rețelelor nu este neapărat în spiritul clasificării general acceptate. Fiecare rețea din cele prezentate este locală din punctul de vederea al analizei pe care o vom face. Tehnologia nu este neapărat relevantă deși mecanismele de captură implementate se bazează totuși pe tehnologii Ethernet (tehnologii de rețea locală – de data asta termenul fiind folosit în accepțiunea sa uzuală).

In fiecare nod putem să avem un agent care poate să intercepteze traficul de rețea și să extragă din el serii de date pe care să le putem folosi ulterior la analiză. Din motive pe care le prezentăm mai jos, un astfel de agent produce multiple serii de date și nu doar o serie simplă de valori corespunzătoare nivelului traficului în fiecare cuantă de timp. Aceste serii de valori sunt transportate la un nod central de analiză cu un mecanism de tip pull (agenții oferă permanent seriile de date colectate iar server-ul de analiză extrage seriile necesare și aplică procesul de analiză). In acest fel de distribuie încărcarea de achiziție și analiză între multiple elemente ale rețelei pentru un impact minim de performanță.

Nu toate nodurile trebuie să fie instrumentate. La limită, încărcarea suplimentară introdusă de instrumentele de achiziție dezvoltate este nesemnificativă și toate nodurile ar putea avea un agent de achiziție dar volumul de date rezultat ar crește inutil complexitatea analizei. Pentru a obține rezultate bune de detecție, doar un număr limitat de noduri semnificative trebuie supravegheat. Este vorba tipic de nodurile releu unde se poate constata traficul global al rețelei (sau rețelelor) care folosesc acest nod releu ca punct de tranzit respectiv de unele noduri terminale cu semnificație speciala (servere, stații normale sau eventual noduri-capcană).

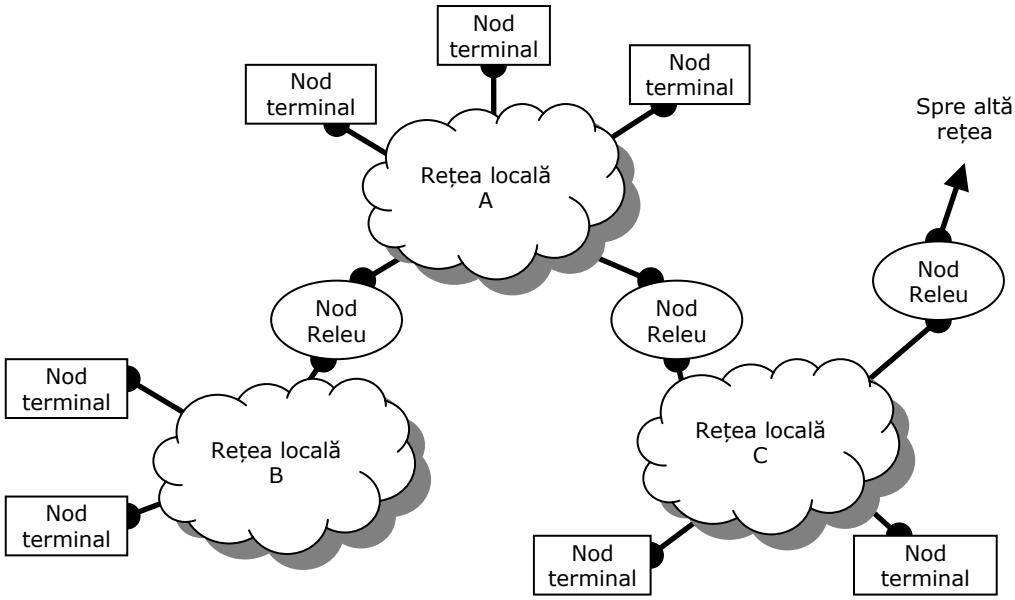


Figura 2 Model de multiple rețele locale cu noduri terminale și noduri releu.

3.2. Principii de îmbunătățire a ratei de detecţie

Putem explica mai clar modul în care procesul de achiziție a seriilor de date poate fi îmbunătățit prin enumerarea unor principii de detecție a anomaliilor. Aplicarea acestora permite obținerea unei performanțe globale mai bune a întregului sistem. Este vorba despre:

- Principiul de îmbunătățire a raportului semnal-zgomot (SNR)
- Principiul „darknet” (spațiu de adresare nefolosit sau puțin folosit)
- Principiul corelației între serii

3.2.1. Principiul de îmbunătățire SNR

Tratăm elementele de trafic indicatoare de anomalii din seriile achiziționate ca semnale înecate în zgomot unde zgomotul este produs de valorile aferente traficului normal. In lumina acestei afirmații, pentru a maximiza performanța oricărui fel de detector este de dorit sa îmbunătățim raportul semnal-zgomot.

In procesarea de semnale, dacă se cunoaște spectrul în care ne așteptăm să găsim semnalul dorit se pot aplica filtre care să reducă energia semnalului cumulat de intrare în zonele de spectru care nu prezintă interes. Prin această operație se reduce efectiv o parte din puterea zgomotului și se ușurează sarcina unui detector.

In cazul achiziției de serii relevante pentru traficul de rețea putem să descompunem traficul în componente după criterii de discriminare naturale (de exemplu tip de trafic sau game de adrese) care se comportă echivalent cu discriminarea după gama de frecvență operată de un filtru clasic. Dacă anomalia se manifestă doar în una din clasele de trafic segregate, raportul semnal-zgomot va fi mult îmbunătățit și șansele de detecție vor fi mult mai mari.

3.2.2. Principiul darknet

Traficul de rețea este condus la destinația corectă folosind adrese specifice fiecărui nivel de abstractizare. Spre exemplu la nivelul legătură de date din modelul ISO-OSI avem adrese fizice (MAC), la nivelul rețea din modelul TCP/IP avem adrese IP iar la nivelul transport avem porturi. Din întreg spațiul de adrese cu valori legale la nivelul entității locale (fie rețea locală, fie nod) doar o parte sunt folosite în traficul normal. Prin urmare, orice trafic constatat în spațiile de adresare nefolosite sau rar folosite (spații întunecate sau „*dark*”) poate fi folosit ca o indicație a unei anomalii.

Eforturile principale în această direcție au fost îndreptate spre spațiile de adresă IP și aici putem enumera: [47][51][52][53]. Denumirea „darknet” care sugerează foarte bine principiul este împrumutată de la Team Cymru [47].

Trebuie să observăm că:

- principiul darknet se poate aplica și asupra altor spații de adresă (în particular asupra spațiului de adrese de transport – porturi)
- spațiile de adresă „*dark*” nu trebuie sa fie complet nefolosite (complet „întunecate”). Putem să avem spații „gri” care au în mod normal un oarecare nivel de trafic dar mult mai redus decât alte spații. Prin compunere cu principiul de îmbunătățire SNR se pot extrage din astfel de spații „gri” informații valoroase despre apariția unor anomalii.

O metodă de exploatare evidentă a principiului darknet pe care n-am gasit-o explicit prezentată în alte lucrări este bazată pe diferențierea între capetele unei conexiuni client-server. Dacă stabilim o relație de direcție pe un flux de trafic unde elementul originator este clientul și elementul destinație este server-ul putem să introducem un nou tip de segmentare a spațiului total de trafic pe baza observației că în relație cu un tip dat de trafic anumite noduri joacă rol numai de server sau respectiv numai de client. Putem introduce așadar o dimensiune suplimentară în spațiile de adrese la nivelele rețea și transport care să creeze segmente „*dark*” utile detecției de anomalii.

Prin extensie, separația client-server se poate face și la nivel de rețea. O rețea „terminală” (care nu are decât un singur punct de conectare cu lumea exterioară) se poate considera ca un nod compus și poate avea un rol predominant sau chiar exclusiv de client în relație cu anumite protocoale sau spații de adrese de transport.

3.2.3. Principiul de corelație

Pentru cele două principii de mai sus am presupus că anomaliile se manifestă concentrat pe anumite subspații în care putem să împărțim traficul global. O parte din anomalii se manifestă într-adevăr în acest mod, dar există suficiente cazuri în care anomaliile au efect asupra mai multor segmente de trafic.

Această observație deschide posibilitatea detecției unor anomalii care se manifestă simultan pe multiple subspații transformate în serii de date, chiar dacă intensitatea manifestată pe fiecare din serii este prea mică pentru a fi detectată cu mijloace directe.

Cel mai simplu exemplu este legătura între ICMP și ARP. Un network-scan bazat pe ICMP destinat să identifice nodurile active din rețeaua locală va produce automat și trafic ARP. Dacă rețeaua în discuție are un nivel relativ ridicat de trafic normal ICMP respectiv ARP față de intensitatea atacului detecția va fi mai dificil de realizat. Prin corelarea seriei de date aferentă ICMP respectiv a seriei de date ARP se

va putea identifica corelația. Traficul normal de ICMP este însoțit de alte tipuri de trafic și nevoia de a rezolva adrese MAC este mai scăzută datorită cache-urilor. Traficul generat de network-scan va încerca multe adrese rar folosite și corelația ICMP – ARP va fi mai mare.

Un alt exemplu este traficul direcționat constatat între două noduri unde unul este client și celălalt este server. Pe baza acestei corelații putem să identificăm în unele cazuri sursa unui atac.

Corelația se poate manifesta și la nivele arhitecturale deasupra nivelului rețea analizat uzual. Spre exemplu o pereche de noduri server (server de aplicație, server bază de date) va prezenta corelație între traficul de tip HTTP care conține solicitări venite de la clienți spre server-ul de aplicație și trafic de tip bază de date între server-ul de aplicație și server-ul bază de date. Modificarea caracteristicilor acestei corelații poate sa indice un atac de tip SQL-injection sau alte anomalii în funcționarea însăși a aplicației, independente de funcționalitatea rețelei dar constatate prin intermediul acesteia.

3.3. Mecanisme de captură și procesare primară

Putem să colectăm relativ simplu trafic de rețea dar volumul acestuia ne obligă să găsim metode de a reprezenta eficient doar anumite caracteristici ale sale, relevante pentru analiza pe care intenționăm să o facem. Asta ne obligă să avem un model simplificat al rețelei în relație cu care să putem simplifica și organiza informația brută din traficul de rețea capturat.

Considerăm că avem un agent de captură care funcționează în interiorul unui nod de rețea și care capturează trafic pe una din interfețele de rețea a acelui nod. Agentul aplică reguli de clasificare asupra traficului și colectează multiple contoare de evenimente organizați pe criterii alese astfel încât să putem face o analiză eficientă. Un sistem centralizat colectează datele de la multipli agenți și aplică algoritmi de analiză pentru a detecta eventualele anomalii.

Presupunerea de la care am pornit în construcția mecanismelor de clasificare este că rețeaua pe care o supraveghem este o rețea Ethernet. Presupunerea nu este extrem de restrictivă pentru că analiza noastră este orientată spre rețelele mici sau medii și încă din 2002 se aprecia că 90% din stațiile desktop sunt conectate folosind Ethernet [54]. In plus, Ethernet este folosit frecvent și ca soluție de nivel 2 OSI pentru conexiuni de categorie MAN sau chiar WAN pe interfețele routerelor, chiar dacă ulterior se face o conversie transparentă de tip tunel la alte protocoale pentru acoperirea distanțelor mai mari.

3.3.1. Separarea în tipuri de trafic și domenii de autoritate

Captura se realizează la nivelul unei interfețe a unui nod din rețea. Presupunem că acest nod este dotat cu o adresă de nivel 2 OSI (Ethernet MAC) și în consecința la nivel 2 OSI vom categoriza următoarele tipuri de trafic [Figura 3]:

1. trafic unicast, având *destinație* adresa MAC a interfeței monitorizate
2. trafic unicast, având *origine* adresa MAC a interfeței monitorizate
3. trafic multicast sau broadcast, având *origine diferită* de interfața monitorizată
4. trafic unicast, *fără relație directă* cu interfața monitorizată. Acest tip de trafic apare în capturi în cazul implementărilor pe bază de hub sau ocazional la implementările comutate (switch) în faza de MAC-location-discovery

Nodurile în care facem captură sunt noduri cu capabilități de nivel 3 OSI sau mai mare (calculatoare gazdă, gateway-uri, routere dar nu bridge-uri, nu repetoare) așadar traficul de nivel 2 OSI este local interfeței pe care se face captura. Putem în acest sens să desemnăm doar două domenii de autoritate față de unitățile de date constatate la nivel 2 pe interfața de captură, folosind chiar interfața de captură ca graniță și anume:

- domeniul intern („inside", „dreapta"), aflat în interiorul nodului
- domeniul extern („outside", „stânga"), aflat în exteriorul nodului

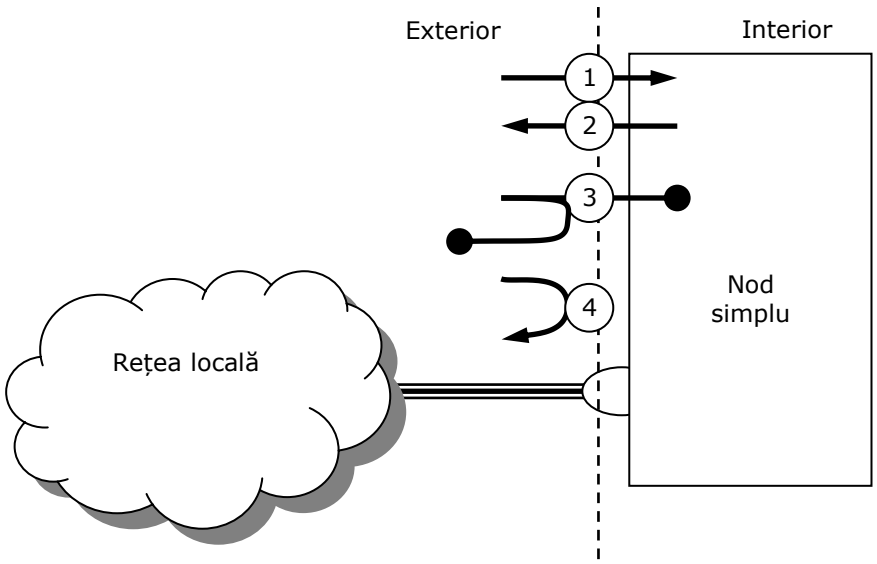


Figura 3 Trafic de nivel 2 OSI la interfaţa de captură. Se evidenţiază cele 4 tipuri de trafic și domeniile de autoritate.

Traficul capturat conţine în majoritatea sa pachete de date (adică unităţi de date de nivel 3 OSI). Acestea sunt dotate direct sau indirect cu adrese de nivel 3 OSI pentru sursă şi pentru destinaţie. In cazul traficului IP adresele sursă şi destinaţie sunt explicit prezente în fiecare pachet deoarece IP este un protocol fără conexiune. In cazul unor protocoale bazate pe conexiune adresele sursei şi destinaţie sunt înlocuite cu mecanisme de identificare a conexiunii dar adresele sursă şi destinaţie ar putea fi determinate prin monitorizarea integrală a întregului trafic, inclusiv secvenţa de stabilire a conexiunii. Pentru simplitate, în cele ce urmează ne vom referi la aceste adrese cu termenul de adresă IP deoarece protocolul IP este extrem de larg răspândit dar principiile rămân valabile și pentru alte protocoale de nivel 3 OSI. Pentru cazul general în care nodul și interfața în discuție suportă mai multe adrese ale aceluiași protocol sau multiple protocoale algoritmul de discriminare se repetă pentru fiecare protocol și fiecare adresă.

Așadar nodul în care facem captură are cel puțin o adresă IP asociată interfeței pe care facem captura. Putem deci să împarţim traficul capturat în tipuri de trafic cu relație la aceasta adresă IP, similar cu mecanismul descris la nivel 2 OSI dar la nivel 3 OSI există mai multe situații posibile. Nodul poate să fie un nod gazdă simplu, care nu suportă tranzit de trafic (chiar dacă este multi-homed) sau poate fi un nod cu capabilități de tranzit (router sau router cu servicii proprii).

In cazul în care nodul este un nod simplu, situația este foarte asemănătoare cu cea descrisă mai sus la nivel 2 OSI, cu deosebirea că în loc de adrese MAC vom avea adrese IP. Putem așadar distinge următoarele tipuri de trafic [Figura 4]:

1. trafic unicast, având *destinație* adresa IP a interfeței monitorizate
2. trafic unicast, având *origine* adresa IP a interfeței monitorizate
3. trafic multicast sau broadcast, având *origine diferită* de interfața monitorizată dar cu destinația în aceeași subrețea
4. trafic unicast, *fără relație directă* cu interfața monitorizată. Acest tip de trafic apare în capturi în cazul implementărilor pe bază de hub sau ocazional la implementările comutate (switch)[3].

In cazul în care un nod este un nod cu capabilități de tranzit, la categoriile de mai sus se adaugă și traficul de tranzit. Pentru că orice nod care face routing are de fapt servicii proprii, char dacă pentru scopuri de întreținere, vom lua în considerare cazul maximal în care putem avea atât trafic terminat în acest nod cât și trafic de tranzit. La un astfel de nod mai constatăm încă două tipuri de trafic și anume

1. trafic unicast, cu adresă de origine în rețeaua locală și adresa de destinație în rețeaua externă
2. trafic unicast, cu adresă de origine în rețeaua externă și adresa de destinație în rețeaua locală

Trebuie să observăm că noțiunea de „rețea locală” este mai sus definită în relație cu interfața la nivelul căreia facem captura. De asemenea, există două tipuri de trafic suplimentare, extensii ale tipurilor 5 și 6, unde înlocuim „rețeaua locală” cu „altă rețea, accesibilă prin rețeaua locală”

Din punct de vedere al autorității vom avea o segregare în domenii dependentă de spațiul pe care dorim să îl protejăm. In cazul unui nod simplu (sau a serviciilor oferite de un nod router) avem:

- domeniul intern local („inside”, „dreapta”), aflat în interiorul nodului
- domeniul extern local („outside”, „stânga”), aflat în exteriorul nodului

In cazul funcționalității de router, pentru demarcarea domeniilor de autoritate folosim însăși router-ul ca și graniță între lumea interioară și cea exterioară iar criteriul interior/exterior se aplică după cum dorim să protejăm sau nu respectivul domeniu. Așadar avem:

- domeniul intern global, care cuprinde toate rețelele proprii
- domeniul extern global, care cuprinde toate rețelele exterioare

[3] Incidența acestor apariții ocazionale la implementările comutate este neglijabilă pentru traficul IP deoarece pachetele IP necesita resolving IP-MAC. Această mapare se face în baza unui cache cu retenție foarte limitată față de retenția tabelelor MAC interne switch-ului așadar dacă e nevoie de interogare explicită ARP aceasta va fi broadcast și răspunsul va actualiza tabelele MAC ale switch-urilor. Există totuși situații în care apare astfel de trafic și anume la restart-ul unui switch din domeniu.

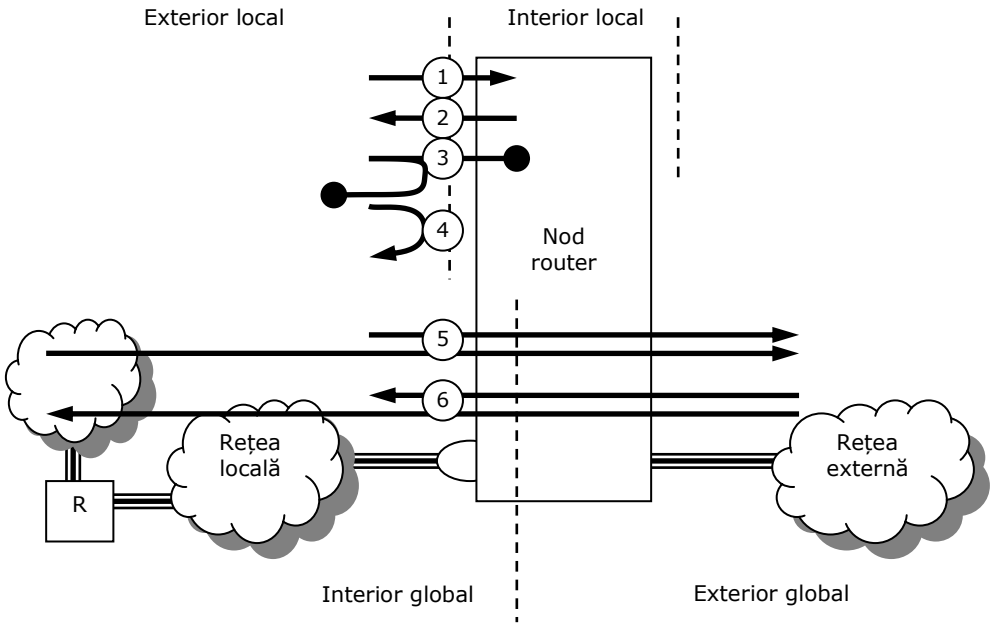


Figura 4 Trafic de nivel 3 OSI la interfaţa de captură. Se evidenţiază cele 6 tipuri de trafic și domeniile de autoritate locale și globale.

Din perspectivă practică, observăm de asemenea că la nivel 3 OSI dispunem de o pereche de adrese IP sursă/destinație ca mijloc de identificare. Dacă din această pereche de adrese nu putem să identificăm trafic de tip 1, 2 sau 3 atunci distincția între traficul din de tip 4, 5 sau 6 nu poate fi făcută decât în anumite cazuri particulare (partener aflat strict în rețeaua locală direct accesibilă). Toate aceste tipuri de trafic (4, 5, 6) au adrese IP care diferă de adresa IP a interfeței considerate a nodului router și nu avem o metodă simplă de a determina apartenența celor două capete participante la trafic la domeniile „intern global" respectiv „extern global". In acest punct devine importantă corelarea cu informația obținută la nivel 2 OSI, ilustrată în tabelul Tabel 1

Incadrare nivel 2 OSI	Incadrare nivel 3 OSI
1	5
2	6
4	4

Tabel 1. Regula de corespondență pentru detecția traficului de tip 4, 5, 6 la nivel 3 OSI

3.3.2. Separarea după rolul partenerilor în relația client-server

In modelul de mai sus nu am făcut referință la funcționalitatea asumată de partenerii de comunicație. O mare parte din traficul de rețea se supune la nivele superioare de abstractizare modelului client-server. Pentru a extrage caracteristici ale traficului de rețea semnificative pentru detecția de anomalii trebuie să

introducem și această dimensiune în modelul pe care vom adopta. Dacă luăm în considerare dimensiunea „client-server” și o suprapunem peste dimensiunea „interior-exterior” obținem cazuri semnificative de interacțiune pe care ar trebui să le reprezentăm în caracteristicile colectate:

- elementul interior (nod sau rețea) este client. Semnificația pe care o atașăm unui astfel de pachet este „un utilizator intern sau un proces automat intern are inițiativa de a face comunicație”
- elementul interior (nod sau rețea) este server. Semnificația pe care o atașăm unui astfel de pachet este „un serviciu intern accepta inițiativa altora de a face comunicație”

Ambele situații pot ilustra cazuri legitime de exploatare sau situații anormale. Ipoteza pe care ne bazăm această lucrare este că situațiile anormale (atacuri sau alte anomalii de rețea) se traduc prin comportamente diferite și detectabile ale celor două categorii majore de trafic de mai sus. La limita extremă se află cazuri evidente de tipul celor enumerate mai jos dar manifestările reale pot fi mult mai nuanțate:

- stație client nu trebuie să manifeste deloc anumite tipuri de trafic „interior este server”
- stație client are un set limitat de scenarii în care își manifestă legitim calitatea de client
- stație server nu are motive normale să manifeste anumite tipuri de trafic „interior este client”
- stație server este destinată unui numar limitat de servicii

Din păcate, dacă localizarea „interior/exterior” este relativ simplu de determinat conform modelului de mai sus, orientarea partenerilor sursă/destinație dintr-un pachet singular pe axa client-server necesită eforturi semnificative (analiza stării protocoalelor de nivel transport purtătoare de stare sau analiza detaliată a conținutului pachetelor din punct de vedere al protocoalelor de nivel superior).

Spre exemplu, dacă pachetul de rețea este parte dintr-un dialog TCP ar trebui să urmărim asocierea între pachetele de inițiere a conexiunii (SYN) și toate pachetele următoare. O astfel de abordare presupune să păstrăm informație despre toate conexiunile TCP care tranzitează interfața pe care facem captură, cel puțin până la închiderea acestora. Acest lucru este fezabil pentru o interfață cu trafic moderat dar devine rapid prohibitiv din punct de vedere al consumului de resurse pentru un trafic intens (cum ar fi un router de graniță, exact unul din punctele în care am dori să facem monitorizare). Există sisteme care sunt nevoite să facă această urmărire a stării (firewall-uri stateful, routere care fac translație de adresă pe bază de port) dar în cazul lor aceasta este o funcționalitate principală care justifică resursele consumate. Sistemul pe care intenționăm să îl descriem trebuie să aibă un impact minimal asupra resurselor rețelei și din această cauză urmărirea stării conexiunilor TCP nu este folosită pentru scopurile analizei noastre.

Dacă traficul incapsulat în pachet nu este TCP, identificarea client/server este și mai dificilă. Traficul UDP nu este purtător de stare la nivelul transport și din analiza unui singur pachet UDP este imposibil de spus cu certitudine dacă sursa este client sau server fără să facem analiza detaliată a conținutului său. Spre exemplu, la captura unui pachet singular care transportă o solicitare DNS încapsulată într-un pachet UDP se poate identifica server-ul și clientul în perechea sursă/destinație dacă identificăm faptul că e vorba de un pachet DNS și analizăm conținutul său in calitate de pachet DNS.

Soluția pe care o adoptăm pentru analiză stă în formularea „identificăm faptul că e vorba de un pachet DNS”. Pentru scopurile funcționale această identificare se face în interiorul server-ului respectiv al clientului pe baza portului destinație la care sosește pachetul. Procesul server DNS ascultă pe portul UDP 53 alocat pentru protocolul DNS și răspunde tipic folosind același port ca sursă. Așadar putem să folosim ca și criteriu euristic portul binecunoscut pentru a identifica tipul de trafic și corespondența client/server peste relația sursă/destinație.

Deoarece serviciile legitime funcționează în marea majoritate a situațiilor folosind porturi binecunoscute putem să clasificăm traficul încadrând portul sursă respectiv destinație în grupe de porturi binecunoscute. Traficul cu sursa binecunoscută va fi considerat „server-to-client”iar traficul cu destinația binecunoscută va fi considerat „client-to-server”. Există situația specială în care atât portul sursă cât și portul destinație sunt binecunoscute (exemple notabile sunt traficul NETBIOS pe porturile 137,138,139 și traficul NTP pe portul 123). Pentru ca în această situație nu se poate identifica server-ul și clientul pachetele aferente sunt încadrate într-o categorie separată.

Am observat anterior că anumite anomalii pot fi pierdute in volumul traficului global și ar fi imposibil de detectat fără un mecanism adițional de îmbunătățire a raportului semnal-zgomot. Un astfel de mecanism este separarea traficului în categorii și analiza separată a semnalului echivalent produs de fiecare categorie.

La primul nivel de separare se face combinarea direcției de trafic cu rolul client sau server derivat din traficul TCP/UDP și obținem trei categorii:

- trafic TCP/UDP, entitatea locală este client
- trafic TCP/UDP, entitatea locală este server
- trafic TCP/UDP, nu pot determina rolul entității locale

In clasificarea de mai sus „entitatea” poate fi nodul în care funcționează agentul de captură sau rețeaua locală din spatele nodului.

Pentru fiecare categorie putem aplica al doilea nivel de separare, pe baza portului binecunoscut care a stat la baza identificării. Din păcate există 65535 porturi posibile și păstrarea de contoare pentru fiecare ar necesita cantități de memorie și eforturi de procesare nerealiste. Putem profita însă de constatarea că într-o rețea dată setul tipic de porturi folosite este limitat și anumite categorii de trafic sunt similare chiar daca au porturi binecunoscute diferite deci pot fi încadrate în aceeași grupă. Separarea traficului la acest nivel nu este unic determinată și pentru a obține performanțe maxime trebuie adaptată la rețeaua aflată în supraveghere. Un exemplu de delimitare a grupurilor de porturi este dat în tabelul Tabel 2 care ilustrează o rețea cu stații Windows care este conectată la servicii uzuale de tip Internet dar care are și servicii de tip aplicație web, baze de date pe sisteme de tip Unix și conţine noduri interconectate printr-o soluție VPN cu alte rețele.

Pentru fiecare grup de porturi aflat în fiecare din cele trei categorii de mai sus se va întreține o serie de contoare de pachete organizate istoric pe intervale de N secunde. Dimensiunea N a intervalelor trebuie corelată cu volumul de trafic din rețea și cu întinderea așteptată a anomaliilor. Intervalul folosit în experiențele efectuate este de 10 secunde.

Nu toate pachetele capturate se vor încadra în una din categoriile descrise mai sus pentru că acestea se referă doar la pachete de protocol TCP sau UDP care au cel puțin portul sursă sau portul destinație în unul din grupurile predefinite. Pentru a completa spațiul pachetelor TCP/UDP mai trebuie să adaugăm o serie de contoare pentru pachetele cu porturi nerecunoscute. Aceste pachete pot fi normale

în rețea dar neprevăzute în schema de grupuri aleasă, pot fi normale dar parte a unui protocol bazat pe porturi dinamice (de exemplu trafic RPC) sau pot fi pachete anormale, parte a unui trafic neașteptat. Pentru claritatea exprimărilor în continuare, ne vom referi la această categorie de porturi și la seriile de contoare asociate cu denumirea OTHER-PORTS sau O-P.

Index	**Porturi**	**Descriere**
1	20,21,25,53,80,110,137,138,443,445	Trafic trivial de utilizator (FTP-DATA, FTP, SMTP, DNS, HTTP, POP, NETBIOS, HTTPS, CIFS)
2	22,3389	Trafic de administrare (SSH, RDP)
3	123,161,514,3551	Trafic generat de sisteme automate (NTP, SNMP, syslog, UPS-manager)
4	1521,2401,3306	Trafic de tip bază de date (Oracle, CVS, MySQL)
5	4500	VPN încapsulat UDP
6	8080, 8443, 9090	Web-application
7	8088	Proxy-server

Tabel 2. Exemplu de grupare a porturilor cu trafic normal

O problemă care trebuie rezolvată într-o implementare reală este alegerea corectă a grupurilor de porturi așa încât să acoperim tot traficul normal cu grupuri și să maximizăm capabilitățile de detecție a anomaliilor. Deocamdată alegerea grupurilor trebuie făcută manual dar poate fi asistată prin identificarea statistică a porturilor din categoria O-P. Așa cum am stabilit deja, nu putem să întreținem serii de contoare pentru fiecare port dar putem să întreținem o listă a celor mai frecvent întâlnite porturi. După o perioadă rezonabilă de monitorizare, prin analiza acestei liste se pot extrage porturi care ar trebui să fie încadrate în grupurile predefinite dar au fost omise la configurarea inițială.

Categoriile descrise mai sus nu acoperă tot traficul de rețea. Există cadre Ethernet care transportă alt gen de informație decât pachete TCP sau UDP. In baza experienței de administrare rețele am adăugat serii de contoare pentru câteva categorii suplimentare de trafic. Există atacuri cunoscute care folosesc aceste tipuri de pachete și e util să avem contoare separate pentru ele:

- cadre Ethernet 802.3 original (de exemplu spanning-tree incapsulat in 802.3 LLC)
- pachete ARP
- pachete ICMP
- pachete de inițializare a sesiunii (deocamdata pachete TCP cu flag SYN)
- pachete de inițializare a sesiunii cu port destinație O-P (deocamdată doar pachete TCP cu flag SYN)
- pachete cu protocol netratat (nici TCP, nici UDP și una din categoriile de mai sus)

4. ANALIZA SERIILOR DE DATE

In urma operaţiilor de captură şi procesare primară prin agregare se obţin serii de valori în domeniul timp. Seriile de valori reprezintă mărimi specifice colectate într-un anume nod conform principiilor de captură descrise deja. Acestea pot fi privite ca semnale şi asupra lor se pot aplica tehnici cunoscute de procesare a semnalelor pentru a obţine informaţii suplimentare şi pentru a detecta situaţiile anormale. Analiza se poate face pentru fiecare semnal în parte sau pentru seturi de două sau mai multe semnale, provenind din acelaşi nod sau din noduri distincte. La modul general, seriile-semnale sunt purtătoare de informație din diverse surse și nu reprezintă neapărat trafic de rețea dar pentru a putea formula simplu anumite constatări vom folosi în continuare termenul de trafic în relație cu seriile de valori pentru a reprezenta intensitatea mărimii constatate.

4.1. Tipuri de anomalii detectabile în seriile de date

Obiectivul analizei este de a detecta comportamente neaşteptate în reţea prin detecţia unor comportamente neaşteptate ale semnalelor reprezentate de seriile colectate. In contextul dat, comportament neaşteptat poate fi:

1. apariţia unei componente suplimentare, intense, de scurtă durată
2. apariţia unei componente suplimentare, de intensitate redusă sau moderată, întinsă pe o durata semnificativ mai lungă de timp
3. apariţia unei componente atipice pentru seria considerată (spre exemplu comportare periodică pe o serie cu distribuţie normal aleatoare auto-similară)
4. lipsa unor caracteristici periodice specifice al anumitor serii, pentru care periodicitatea este indicator al funcţionării normale
5. apariţia de valori semnificative pe serii care ar trebui să conţină valori zero
6. valori zero sau nesemnificative pe serii care ar trebui să conţină valori semnificative
7. prezenţa unor corelaţii nejustificate între serii obţinute în acelaşi nod
8. absenţa corelaţiilor justificate între serii obţinute în acelaşi nod
9. prezenţa unor corelaţii nejustificate între serii obţinute în noduri diferite
10. absenţa unor corelaţii justificate între serii obţinute în noduri diferite

Anomaliile nu sunt exclusiv reprezentate de categoriile de mai sus sau încadrate în una din ele. Acceptăm posibilitatea ca o anomalie să poată fi încadrată la mai multe categorii sau să existe fără sa poată fi încadrată la nici una. Existența enumerării de mai sus este justificată de posibilitatea construcției unor metode de procesare care să detecteze aceste situații.

Apariţia unui volum de trafic suplimentar neaşteptat poate fi datorată unor evenimente externe justificate, unor defecţiuni, unor probleme de configurare sau unor atacuri. Această situaţie se reflectă în comportament de tip 1 sau 2, după durata anomaliei și după intensitatea acesteia. Cauzele pot fi diferite și metodele de detecție pot fi diferite dar după aplicarea unor operații de scalare în timp și amplitudine cele două pot fi considerate echivalente. O schemă simplă de detecție poate folosi o transformare wavelet urmată de o simplă detecție cu prag.

Exemple pentru anomalie de tip 1: transfer legitim de fișier mare dintr-o sursă apropiată și capabilă de bandă largă, defecțiune software care generează un număr mare de solicitări într-o buclă cu repetare rapidă sau portscan agresiv. Trebuie observat că fiecare caz se manifestă pe serii diferite. Primele două exemple se pot detecta pe serii generate din grupe de porturi (fie seria client fie seria server) iar atacul de tip portscan se poate detecta pe seria pachetelor ARP sau pe seria pachetelor cu porturi neîncadrate.

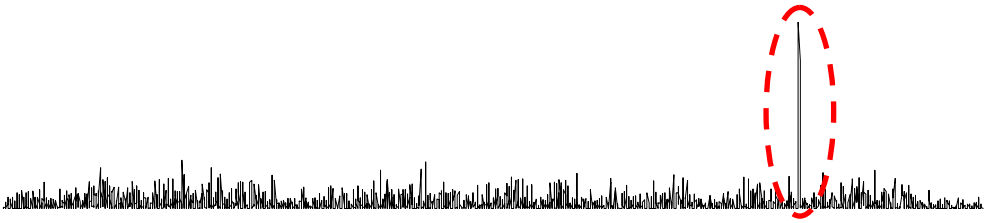


Figura 5. Anomalie de tip 1: impuls de amplitudine mare și durată scurtă

Exemple pentru anomalie de tip 2: trafic de tip flash-crowd (sesiuni multiple de download declanșate de un eveniment extern), rută incorectă generatoare de multiple pachete ICMP-REDIRECT sau atac portscan cu comportare discretă (stealthy).

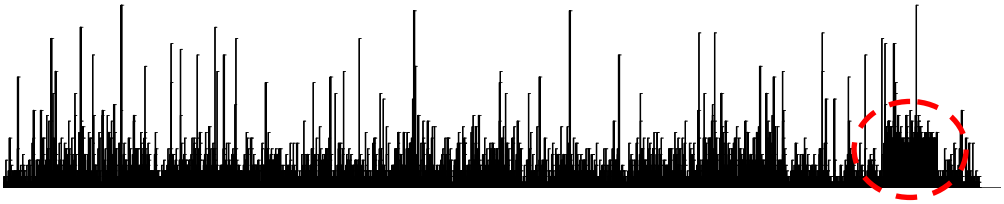


Figura 6. Anomalie de tip 2: componentă de amplitudine moderată, pe o durată mai extinsă de timp

Anomaliile de tip 3 si 4 se bazează pe faptul că anumite categorii de trafic au o caracteristică specifică, diferită de caracterul aleator auto-similar tipic pentru traficul general de rețea. O situație generatoare de anomalie de tip 3 este spre exemplu un atac de tip portscan derulat cu intensitate redusă pentru a evita detecția dar fără introducerea unei componente aleatoare în intervalul de explorare. Tot o anomalie de tip 3 poate fi considerată funcționarea unui sistem neautorizat de monitorizare care folosește polling (inversul situației descrise mai jos).

Mai ușor se pot găsi exemple pentru anomalii de tip 4, cum ar fi traficul generat de diferite instrumente de monitorizare legitimă cu caracter periodic (sau de fapt lipsa lui). Dacă ne îndreptăm atenția spre serii colectate din pachete UDP găsim situații în care se folosește polling periodic chiar la colectarea de date pentru monitorizare prin mecanisme SNMP sau găsim mecanismul folosit de pachetul software apcupsd pentru distribuția notificărilor de închidere automată la toate calculatoarele deservite de aceeași sursă de alimentare neinteruptibilă (UPS). Și la nivel legătură de date există astfel de componente periodice - spre exemplu cadrele

BPDU din protocolul STP (spanning-tree, 802.1d) care anunță nodul rădăcină. Intr-o rețea în care se cunoaște existența unor astfel de componente de trafic periodic, absența lor este o anomalie care poate fi un simptom al unei defecțiuni sau al unui atac. Figura 7 prezintă un exemplu de trafic generat de două procese de monitorizare cu perioade distincte și mai jos transformata FFT a seriei (doar in modul).

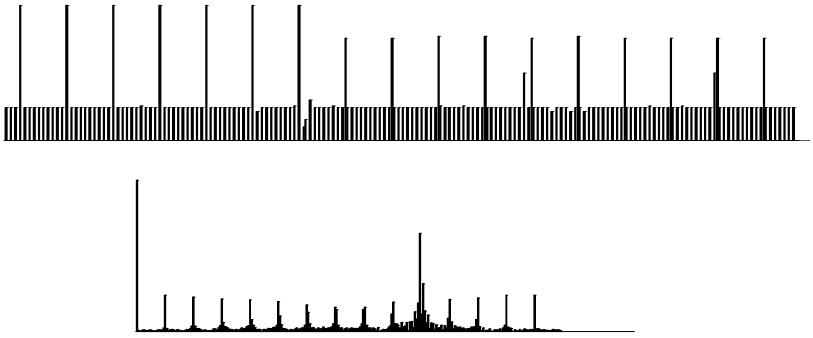


Figura 7. Trafic periodic generat de două instrumente de monitorizare și transformarea FFT (doar modul)

Trebuie să observăm că absența aparentă în semnal a componentelor periodice cunoscute nu este neapărat cauzată de întreruperea acelui tip de trafic. Există posibilitatea ca traficul periodic binecunoscut să fie îngropat în „zgomot" cu evoluție lentă, generat de trafic neașteptat, ceea ce este tot o anomalie dar poate avea altă cauză, diferită de buna funcționare a interogărilor periodice. In acest caz traficul periodic este folosit ca o referință la care se raportează traficul global constatat pe seria în discuție. Anomalia în acest caz poate fi considerată ca una de tip 2 dar cu o evoluție așa de lentă încât ar fi dificil de detectat în absența „semnalului" periodic de referință.

Anomaliile de tip 5 și 6 sunt evidente și nu necesită prelucrări semnificative. Importantă în acest caz este abilitatea de a separa spațiul de trafic global în secțiuni normal populate și secțiuni normal nepopulate, plus transformarea lor în serii. Subliniem că această discriminare nu înseamnă că o anumită serie va avea în mod normal doar valori diferite de zero sau doar valori zero. Orice logică de detecție trebuie să facă o decizie puțin mai nuanțată: vom căuta de fapt valori semnificativ de mari în serii în care valorile medii sunt foarte scăzute și invers.

In încheierea setului de comportamente neașteptate, indicație despre posibile anomalii, includem cazurile 7, 8, 9 și 10 obținute prin urmărirea existenței sau absenței unor corelații între seriile obținute în același nod sau în noduri diferite. In multe situații activitatea din rețea produce efecte pe mai multe serii chiar dacă am depus eforturi ca acestea să reprezinte secțiuni diferite din trafic. Un exemplu de corelație justificată între serii din același nod este corelația între seria aferentă sesiunilor utilizator linie de comandă și seria aferentă traficului de poștă electronică (porturi 22,23 respectiv porturi 25,110). Corelația este justificată de situația reală în care utilizatorii își accesează ocazional contul pe o mașină Unix pentru a verifica poșta electronică. Exemplul este extras din capturile DARPA-98 și actualmente este perimat deoarece cazul tipic de exploatare din care derivă nu mai este folosit. Un exemplu însă de corelație normală mai frecvent întâlnită în rețelele de astăzi este

corelația între traficul de tip HTTP și traficul de tip bază de date. In situația în care mașina care funcționează ca server HTTP este diferită de mașina care funcționează ca bază de date solicitările de pagini dinamice sunt transformate în solicitări SQL către baza de date, ceea ce se reflectă într-o puternică corelație între cele două serii de date.

Există o altă situație care poate să conducă la corelație între serii de date ale aceluiași nod, diferită de exemplul de mai sus. In cazul HTTP/bază de date corelația este imediată și nu implică nici o decalare temporală între cele două evenimente corelate (și seriile lor asociate). Dacă vom considera însă cazul unui nod care funcționează ca mașină de build automat într-un mediu profesional de dezvoltare de software vom constata că un astfel de nod parcurge o serie de pași care se reflectă în comportamentul seriilor sale atașate și generează o corelație cu offset temporal care poate fi chiar variabil de la un caz la altul.

O buclă tipică de build continuu urmărește permanent eventualele modificări în sursă și în cazul în care apar astfel de modificări parcurge următoarea serie de pași:

- descarcă modificările de sursă de pe server-ul de versionare (CVS, SVN, ...)
- procesează local codul sursă (de exemplu generare de cod, compilare)
- execută testelor automate
- salvează codul rezultat într-un spațiu comun din care poate fi utilizat de dezvoltatori ai altor module

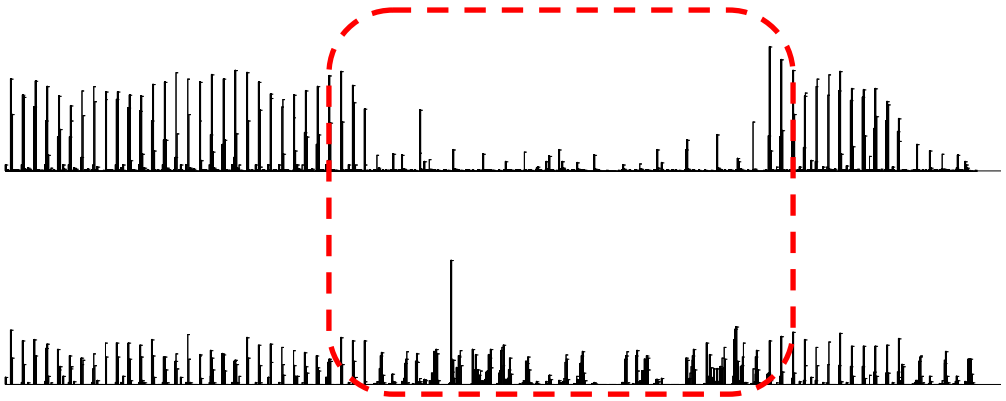


Figura 8. Serii corelate generate de buclă de build continuu. In zona marcată s-a derulat un build, în exterior se monitorizează doar modificările.

Fiecare pas poate să producă trafic detectabil pe serii diferite. Consultarea server-ului de versionare se desfășoară pe porturi specifice, ușor de identificat în serie separată. Procesarea surselor poate să necesite descărcarea versiunilor recente ale bibliotecilor de care depinde modulul curent (trafic FTP sau HTTP). Execuția testelor automate poate să implice accesarea unor baze de date. Salvarea rezultatului finit este din nou trafic FTP sau HTTP. Maximele locale pe seriile amintite apar corelat dar nu în același moment de timp ci decalate și decalajul nu este constant ci depinde de conținutul efectiv al build-ului. Implicația acestui fapt este că

vom avea nevoie de o tehnică de corelație locală care să identifice similarități pe o întindere variabilă în timp.

Pentru cazul mașinii de build continuu se poate constata un alt aspect interesant. Procesul de verificare a apariției modificărilor este periodic și se poate încadra la situația 4 de anomalie dar periodicitatea sa este întreruptă de secvențe de corelație locală între seriile descrise mai sus. Din nou constatăm așadar că este nevoie de un instrument cu care să putem evalua corelația pe intervale variabile de timp.

Dacă extindem spațiul de analiză în cautarea de corelații de la seriile unui singur nod la seriile colectate de la multiple noduri vom găsi și mai ușor exemple pentru că acolo unde există un server trebuie să avem și un client și invers. Majoritatea tipurilor de trafic sunt client-server așa încât putem să identificăm corelație datorată acestui tip de relație. Există însă și corelații non-triviale care implică noduri diferite. Acestea implică undeva o cauză comună și un bun exemplu poate fi tot cel al buclei de build continuu dar aplicat pe seriile de trafic colectate la server-ul de versionare și la server-ul de date care aparent nu au nici un motiv să manifeste trafic corelat. Dacă ne raportăm la acest model, chiar și în condițiile în care nu facem monitorizare pe nodul în care funcționează bucla de build efectele acesteia pot fi constatate indirect.

Tot în cazul seriilor colectate în noduri diferite (sau pe interfețe diferite ae aceluiași nod) putem să găsim ușor exemple de corelație în cazul în care traficul generat de nodul interior A spre o rețea exterioară se regăsește în traficul spre exterior generat de nodul router de la granița spre rețeaua exterioară. Tipul de trafic al celor două serii nu e necesar să fie identic. Exemple notabile: trafic interior pe port 3128 (http proxy) se va găsi reflectat în trafic pe port 80 și/sau 443 spre exterior respectiv trafic interior pe diverse porturi (cu destinație o rețea distantă) se va regăsi în trafic de tip ESP (Encapsulating Security Payload) sau ESP încapsulat în UDP (tipic port 4500). Exemplul se regăsește în Figura 9 stânga.

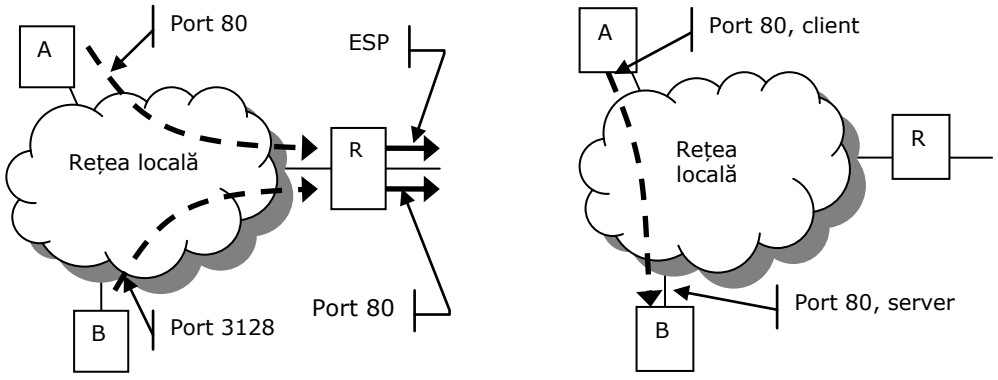


Figura 9. Exemple de corelație în trafic stație-router (stânga) și stație-stație (dreapta)

Exemplele de mai sus se referă la situații normale în care există corelație datorită unor procese legitime derulate de utilizator. Absența unor astfel de corelații generează anomalii de tip 8 sau 10. Putem însă să avem și corelație între serii

cauzată de situații anormale și în acest caz anomalia detectată este de tip 7 sau 9. Spre exemplu, traficul generat de un atac portscan extrem de agresiv (care se consumă pe durata a câteva eșantioane) se va regăsi practic pe toate seriile unui nod. Dacă atacul este generalizat la multiple noduri corelația va fi regăsită și între serii din noduri diferite, chiar pentru porturi diferite.

Un caz special de corelație neașteptată este cel în care se desfășoară un atac derulat de pe stația A împotriva stației B, ambele fiind în mod normal necorelate (stații client banale). Intre seria corespunzătoare „client" colectată pe stația A și seria corespunzătoare „server" colectată pe stația B va apare o corelație care permite nu doar detectarea anomaliei ci și identificarea sursei chiar dacă mecanismele de colectare nu conservă informația detaliată despre partenerii care au generat traficul înregistrat în serii (Figura 9 dreapta).

4.2. Instrumente de detecţie a anomaliilor singulare.

Considerăm anomalii singulare anomaliile de tip 1 respectiv 2 prezentate mai sus. Tot în aceasta categorie se pot încadra anomaliile de tip 5 respectiv 6. Detecția anomaliilor de tip 1 și 2 (componente suplimentare în serii de trafic normal) va fi mai greu de realizat decât detecția anomaliilor de tip 5 unde nu există deloc trafic de fond respectiv 6 unde anomalia este dată de absența completă a traficului normal.

Pentru detecția acestor anomalii vom investiga în continuare detecția simplă cu prag, aplicarea unor operatori de preprocesare, tehnici de procesare timp-frecvență și compunerea acestora.

Deoarece componenta suplimentară poate să aibă diverse durate și aspecte (impuls cu fronturi ideale, impuls trapezoidal) vom analiza comportamentul tehnicilor de detecție în relație cu aceste caracteristici ale impulsului perturbator.

Aspectul traficului (predictibilitatea acestuia sau caracteristica de dependență pe durate lungi de timp) este de asemenea important și vom lua în considerare parametrul Hurst al seriilor de date implicate.

Modelul de test pe care urmează să îl folosim presupune un segment inițial martor neafectat urmat de o zonă perturbată și apoi din nou o zonă neafectată. Această construcție este aleasă astfel pentru a face simulările să fie mai ușor de realizat. În cazul real de detecție avem de-a face doar cu o zonă neafectată urmată de perturbație și procesările descrise trebuie aplicate în regim de fereastră glisantă (ideal cu pas 1) pentru a minimiza întârzierea între momentul apariției anomaliei și momentul detecției. Așadar performanța algoritmului de detecție va fi importantă pentru o implementare practică.

4.2.1. Detecţie cu prag.

Pentru detecţia anomaliilor singulare (tip 1 şi 2) cea mai simplă schemă este aplicarea unui prag. Mecanismele tipice descrise în alte lucrări [55][56][57] presupun o preprocesare care să accentueze impulsurile interesante, urmată de aplicarea pragului de decizie. In fapt, pentru extracţia unor decizii de tip boolean cu privire la apariţia oricaror anomalii, inclusiv pentru fazele finale de detecţie ale celorlalte tipuri de anomalii, trebuie să folosim metode similare, aplicate pe serii derivate din seriile originale.

Prima problemă care trebuie rezolvată în legătură cu această metodă este alegerea pragului. O alegere frecvent folosită în prelucrarea statistică a datelor este

legată de deviaţia standard. Seriile pe care le procesăm pot fi privite ca seturi de măsurători ale unei variabile aleatoare *X*. O astfel de variabilă este descrisă de distribuția sa de probabilitate și are atașate mărimile μ (valoare medie) și σ (deviație standard).

$$\begin{aligned} \mu &= \mathrm{E}[X] \\ \sigma &= \sqrt{\mathrm{E}[(X-\mu)^2]} \end{aligned} \tag{6}$$

Pentru un set de valori discrete $\{x_i\ ,\ i=1..N\}$ rezultate în urma unor măsuratori (aşa cum sunt seriile pe care le tratăm aici) putem să definim valoarea medie a seriei şi deviaţia standard ca valori care aproximează mărimile μ și σ definite pentru variabila originală *X*.

$$\begin{aligned} \bar{x} &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ s &= \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2} \end{aligned} \tag{7}$$

Conform inegalităţii lui Cebâșev putem să delimităm probabilitatea ca variabila *X* să se afle mai departe de medie decât un anumit prag.

$$P\left(\left|X-\mu\right| \geq k\sigma\right) \leq \frac{1}{k^2} \tag{8}$$

In baza acestei inegalități, pentru k=3 probabilitatea ca variabila să fie în afara intervalului +/- 3σ este mai mică decât 1/9. Limita este dată fără vreo restricție în ceea ce privește distribuția de probabilitate a variabilei *X* și din acest motiv nu introduce decât o limita relativ slabă.

Dacă luăm în considerare însă ipoteza că datele procesate sunt reprezentante ale unei variabile aleatoare cu distribuţie normală se poate demonstra că valorile care se îndepărtează cu mai mult de 3σ de valoarea medie au o probabilitate de apariție sub 0,3%, valoare de eroare reziduală pe care o putem considera rezonabilă. Altfel spus, apariția unor valori în afara intervalului 3-sigma față de medie poate fi considerată anormală cu o probabilitate de peste 99,7%.

Se poate obţine o probabilitate arbitrar de mare ca valorile detectate să fie anormale, mărind valoarea lui *k*, dar de fapt aici este vorba atât despre minimizarea falsurilor pozitive cât și despre minimizarea falsurilor negative. O creștere a valorii lui *k* va ridica pragul pe care trebuie să-l atingă valoarea variabilei aleatoare ca să producă un eveniment, mărind rata de falsuri negative (situații în care o anomalie rămâne nedetectată). O scădere a valorii lui *k* va coborî pragul de detecție și va mări rata de falsuri pozitive (situații când nu s-a produs o anomalie dar s-a generat un eveniment).

Extragerea valorilor anormale care ar trebui să identifice anomaliile de trafic este mai eficientă dacă înainte de aplicarea pragului se face o procesare care să accentueze aceste valori anormale. Dacă introducem presupunerea că un impuls anormal se evidențiază fie prin amplitudine, fie prin durată față de traficul normal

atunci o metodă simplă este aplicarea unei filtrări de tip mediere cu fereastră mobilă. Dimensiunea ferestrei de mediere este un parametru important pentru că definește punctul de compromis între sensibilitatea detecției și eliminarea alarmelor false datorate comportamentului natural variabil al seriilor de valori.

O altă tehnică frecvent folosită datorită simplității ei este aplicarea operatorului neliniar de energie (NEO – Nonlinear Energy Operator) descris de Kaiser [58] sub numele de „Teager's Energy Algorithm" sau „Teager's Energy Operator - TEO". In continuare ne vom referi la acest operator cu acronimul generic NEO.

In forma sa discretă operatorul este descris de

$$y_n = (x_n)^2 - x_{n-1}x_{n+1} \tag{9}$$

unde x_i este seria de valori pe care trebuie să o analizăm și y_n este valoarea rezultată din aplicarea operatorului în punctul n al seriei. La origine operatorul a fost introdus pentru a estima energia necesară pentru a genera un semnal sinusoidal astfel încât să se respecte modelul mecanic al unei mase suspendată pe un resort. Datorită acestui model, energia estimată depinde de amplitudine dar și de frecvență, ceea ce face ca impulsurile (care conțin componente spectrale cu frecvențe mai ridicate) să fie accentuate.

Se poate demonstra că valoarea y_n este energia sinusoidei de fază zero care este determinată de eșantioanele x_{n-1}, x_n, x_{n+1} [59] Această observație sugerează că:

- sensibilitatea maximă a metodei de accentuare cu NEO se va regăsi în zona frecvențelor înalte (sau altfel spus pentru impulsurile înguste, ideal de lățime 1)
- detecția impulsurilor mai largi nu beneficiază de această tehnică decât într-o măsură limitată
- trebuie să folosim alte metode, din gama timp-frecvență, pentru a identifica mai bine impulsurile de lărgime mai mare.

In lumina observației de mai sus, se poate face o generalizare a NEO considerând valori aflate la o distanță mai mare de valoarea x_n. Generalizarea este oarecum echivalentă cu aplicarea NEO original pe o variantă subeșantionată a seriei de valori. Aplicarea corectă din punct de vedere a teoremei de eșantionare ar presupune într-adevăr o filtrare trece-jos suplimentară dar în cazul de față nu ne propunem decât să accentuam o caracteristică a semnalului pentru a face mai eficientă separarea cu prag. Avantajul aplicării simplificate (fără filtru) este dat de simplitate și eficiență in execuție. Așadar vom considerăm că aproximarea este acceptabilă până la o valoare rezonabilă a factorului de subeșantionare (sau altfel spus a distanței k între x_n și valorile adiacente folosite). Forma generalizată a operatorului este așadar:

$$y_{n,k} = (x_n)^2 - x_{n-k}x_{n+k} \tag{10}$$

Studiile anterioare [56] arată că în prezența zgomotului este recomandabil să se folosească o variantă a operatorului NEO și anume SNEO, obținut prin convoluția cu o fereastră de uniformizare. Pentru simplitatea implementării fereastra poate fi aleasă cât mai simplă, de exemplu o fereastră Bartlett.

4.2.2. Evaluarea tehnicilor de detecţie cu prag

Pentru a putea evalua soluțiile cu prag și filtrare respectiv soluția SNEO am derulat o serie de simulări a procesului de detecție pe cazuri de semnale sintetice [60]. Deoarece comportamentul statistic al seriilor normale extrase prin procedeul de captură descris anterior are o largă variație între anti-persistent și auto-similar (în funcție de sursa considerată) trebuie să folosim ca semnal de bază serii sintetice din fiecare categorie.

Din lucrările originale care studiază fenomenul de auto-similaritate pentru traficul de rețea rezultă că parametrul Hurst pentru o serie de valori derivată din trafic de rețea ar trebui să fie 0.8 sau chiar mai mare. Estimările parametrului Hurst pentru seriile obținute prin mecanismul de captură și numărare descris anterior conduc la valori puternic variabile în întreg intervalul 0..1 dar marea majoritate sunt plasate în zona valorilor mici. Explicația pentru aceste valori reduse constatate în capturile noastre stă în faptul că valorile de trafic din lucrarea originală a lui Leland, Taqqu, Willinger și Wilson [26] au fost globale și constatate pe o rețea cu mulți participanți iar seriile noastre sunt focalizate pe segmente de protocoale și acestea induc imediat și o populație mai redusă de surse.

Rezultatul constatat este în concordanță cu cercetări ulterioare lucrării originale. In 1997 Peha constată că natura auto-similară este posibil să fie indusă de retransmisii [28]. In 2001 Uhlig și Bonaventure analizează capturi de trafic pe rețeaua unui ISP belgian și regăsesc caracteristica auto-similară dar compusă din componente cu distribuție exponențială [61]. Concluzia lor este că distribuția auto-similară a populației de surse și a activității acestora este generatoare de comportament auto-similar. Diverse alte lucrări apărute din 2003 până în 2012 ridică rezerve asupra aplicabilității stricte a modelului LRD pentru trafic de rețea [62][27][63][64][65]. Concluzia pe care trebuie să o tragem este că modelul este aplicabil dar nu în orice condiții. Din inspecția vizuală a seriilor rezultate prin captură respectiv a seriilor sintetice generate ca mișcare Browniană fracționară rezultă că pentru majoritatea seriilor rezultate din capturile speciale cu filtrare pe scopul traficului parametrul Hurst este mic, sub 0.5. Cu toate acestea, traficul global sau segmente de trafic capturate la granița unei rețele cu multe noduri vor avea distribuții Hurst apropiate de valorile mari constatate de Leland și colaboratorii săi. Din acest motiv vom face estimările de performanță pe serii generate atât cu valori reduse cât și cu valori ridicate pentru parametrul Hurst.

In Figura 10 prezentăm un exemplu de simulare a detecției folsind metoda simplă cu filtrare și prag respectiv metoda cu aplicare SNEO urmata de aplicarea unui prag. Simularea pornește de la o serie de 8000 valori simulate ca o mișcare Browniană fracționară cu parametrul Hurst 0.1. Am ales acesti parametri pentru că sistemul de captură folosit produce 8640 valori pe zi iar estimarea parametrului Hurst pentru multe din seriile capturate experimental are tipic valori apropiate de 0.1.

Pentru că reprezentarea este doar un exemplu, am normalizat seria de valori și am suprapus un impuls unitar de lărgime 100 eșantioane, astfel încât să fie vizibil pe diagrame (diagrama b). Fereastra filtrului de mediere din exemplu are lărgimea de 64, potrivit cu lărgimea impulsului perturbator. Conform cu rezultatele prezentate mai sus, după filtrare (diagrama c) am aplicat un prag la 3-sigma care produce detecția clară a impulsului original (diagrama d).

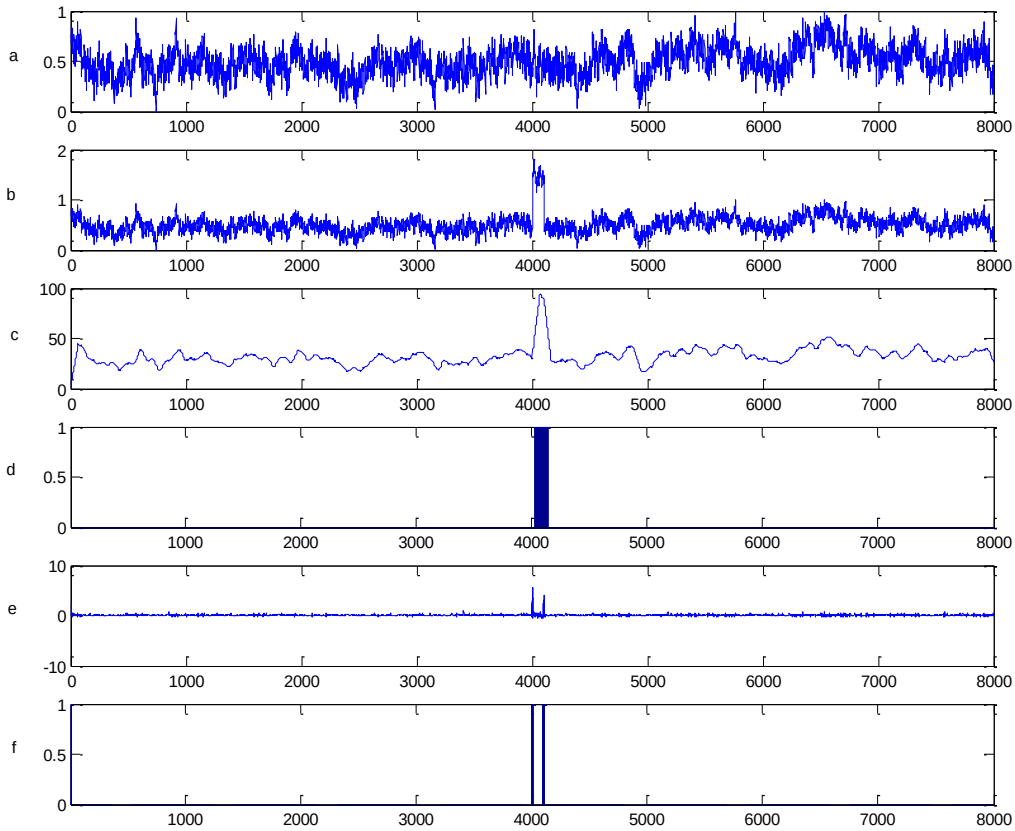


Figura 10. Detecția unui impuls perturbator. a – serie inițiala (Hurst=0.1); b – serie cu impuls unitar; c – varianta filtrata cu filtru medie rulantă; d – rezultat după detector (3-sigma); e – serie după aplicare SNEO (Bartlett 9); f – rezultat SNEO după detector (5-sigma)

Pentru ilustrarea metodei bazate pe SNEO am aplicat operatorul NEO în forma sa originală pe o extensie cu zero a semnalului original. Valorile aflate la marginea intervalului pot fi așadar afectate de erori dar aceasta nu ne deranjează pentru scop ilustrativ. Intr-o implementare operațională ar trebui aplicată o fereastră la marginea intervalului sau ar trebui extins intervalul cu valori din seriile anterioare respectiv ulterioare (în masura în care acestea sunt disponibile). Rezultatul a fost supus unei operații de convoluție cu o fereastră Bartlett de lățime 9 (diagrama e) și unei detecții de prag 5-sigma (diagrama f). Fereastra Bartlett am ales-o pentru că secvența de valori aferentă este ușor de calculat și are o comportare mai naturală decât o fereastră simplă dreptunghiulară. Am ales de asemenea un prag mai ridicat în urma câtorva experiențe care indică sensibilitatea sporită de detecție datorată operatorului NEO astfel încât să reducem rata de falsuri pozitive.

Se observă că metoda simplă cu prag produce o variantă cvasi-identică a anomaliei injectate dar cu on offset în timp datorat procesului de mediere/filtrare.

Metoda SNEO detectează de fapt pragurile crescătoare/descrescătoare ale impulsului, adică de fapt e sensibilă la componentele de frecvențe instantanee înalte ale semnalului perturbator.

Am analizat în continuare comportamentul celor două metode de detecție la diferite tipuri de semnale respectiv la lățimi diferite ale impulsului perturbator. Pentru un anumit tip de semnal (adică sintetizat cu o anumită valoare a parametrului Hurst) și pentru o anumită lățime a impulsului perturbator am crescut progresiv amplitudinea relativă a impulsului urmărind condiţiile de mai jos:

- impulsul a fost corect detectat (deci nu am fals negativ)
- au disparut falsurile pozitive datorate unui SNR prea scăzut între seria originală (echivalată cu noise) și impuls (echivalat cu semnal util).

Valoarea amplitudinii relative a impulsului la atingerea condițiilor descrise mai sus am reprezentat-o funcție de H (parametrul Hurst al seriei) și de lățimea impulsului. Deoarece rezultatele au doar semnificație statistică, fiecare punct din grafic este rezultatul medierii valorilor obținute peste un set de 20 experiențe cu aceiași parametri astfel încât să se elimine mare parte din eventualele componente accidentale rezultate din caracterul aleator al semnalelor sintetice folosite.

Am derulat două serii de simulare, una pentru filtru de lățime 32 și alta pentru filtru de lățime 16. Figura 11 este construită pentru un filtru de lățime 32 iar Figura 12 pentru un filtru de lățime 16. Am reprezentat două suprafeţe unde Z este amplitudinea relativă a implusului faţă de semnalul original normalizat:

- una pentru valorile de prag de la care nu se manifestă falsurile pozitive (în general la valori mai mici, original reprezentată în culoarea roșie)
- una pentru valorile de prag de la care nu se manifestă falsurile negative (în general la valori mai mari, original reprezentată în culoara albastră)

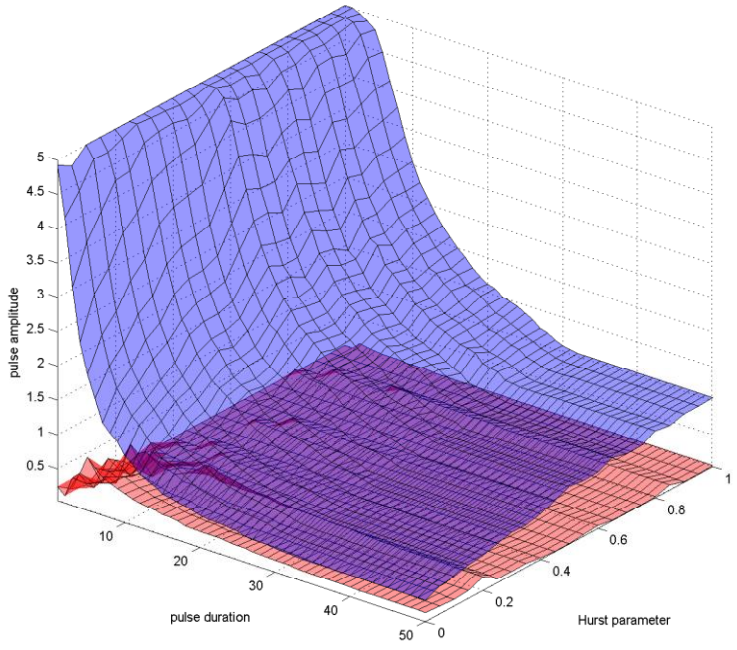


Figura 11. Amplitudinea relativă a impulsului perturbator detectabil cu prag si filtru-fereastra de 32 puncte funcție de durata și factorul Hurst.

Ca notă de implementare trebuie să spunem că limitarea amplitudinii la valoarea 5 în zona impulsurilor scurte este artificială şi se datorează algoritmului de explorare pe care am fost obligaţi să îl limităm pentru a obţine timpi de execuţie rezonabili. De asemenea se observă nişte artefacte persistente în zona valorilor parametrului Hurst 0.2 şi respectiv 0.6 – 0.8 pentru care nu am găsit explicaţie şi care probabil sunt legate de modul în care este construită intern funcţia Matlab wfbm() pe care am folosit-o la sinteza seriilor.

Se observă în Figura 11 că factorul determinant este pragul de detecţie (pragul de la care nu mai apar falsuri negative). Algoritmul de detecţie este mai eficient la cazurile cu parametru Hurst mai mic şi eficienţa se degradează spre parametru Hurst mai mare decât 0.5 unde la limita superioară (H=1) avem nevoie de un impuls perturbator egal cu semnalul original pentru a fi detectat cu succes.

Observăm de asemenea că imediat ce lăţimea impulsului depăşeşte lăţimea filtrului care mediază eficienţa detecţiei nu se mai îmbunătăţeşte, altfel spus deşi impulsul conţine o energie mai mare nu este mai bine detectat. Acest aspect e vizibil şi mai bine în Figura 12 pe suprafaţa superioară care atinge performanţa maximă de detecţie la impuls de lăţime 16.

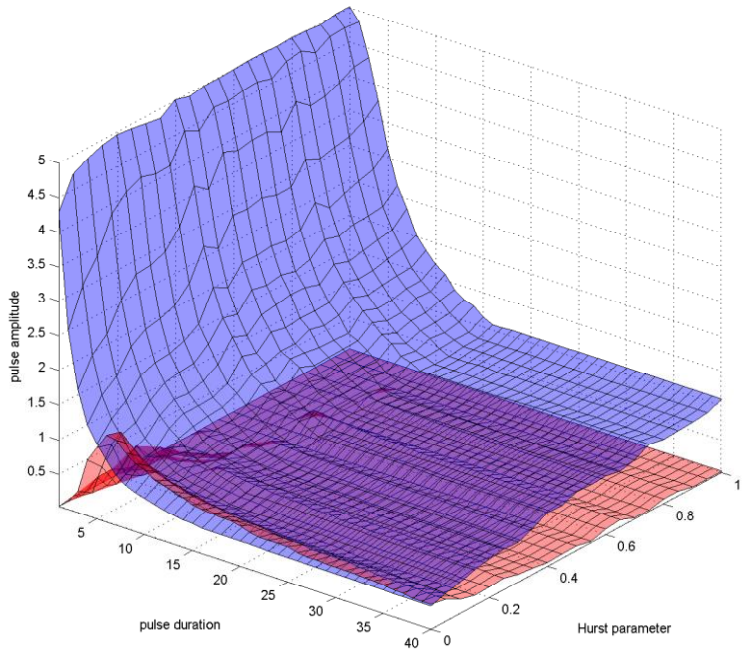


Figura 12. Amplitudinea relativă a impulsului perturbator detectabil cu prag si filtru-fereastra de 16 puncte funcţie de durata şi factorul Hurst

Aparent soluţia ar fi folosirea unui filtru foarte îngust sau eliminarea completă a filtrului dar din comparaţia celor două diagrame rezultă că un filtru mai îngust (Figura 12) conduce la o sensibilitate mai mare la falsuri pozitive în zona semnalelor cu H mic şi pentru impulsuri de scurtă durată.

O întrebare legitimă este „de ce sensibilitatea la falsuri pozitive depinde de amplitudinea şi durata impulsului perturbator?”. In modelul pe care l-am folosit se

face o raportare a pragului de detecţie la deviaţia standard a semnalului compus (care include şi impulsul perturbator) dar verificăm prezența falsurilor pozitive într-o zonă în care nu există impuls perturbator (intervalul de la începutul seriei până înainte de începutul impulsului).

Am aplicat aceeași tehnică de evaluare şi reprezentare şi pentru metoda de detecție bazată pe SNEO. Rezultatele se regăsesc în figurile Figura 13 până la Figura 18 pentru diverse combinații de parametri. Am investigat impactul dimensiunii ferestrei cu care se execută convoluția (9, 17, 33, 65) și efectul aplicării variantei extinse a operatorului (*k* ia valorile 1, 4, 16). Ca observație de implementare trebuie să ținem seama la evaluarea comparativă a rezultatelor că reprezentările sunt făcute cu alt punct de vizualizare față de cele de la varianta filtru și prag deoarece comportamentul dependenței de lățimea impulsului și de parametrul Hurst se poate observa mai bine în acest caz.

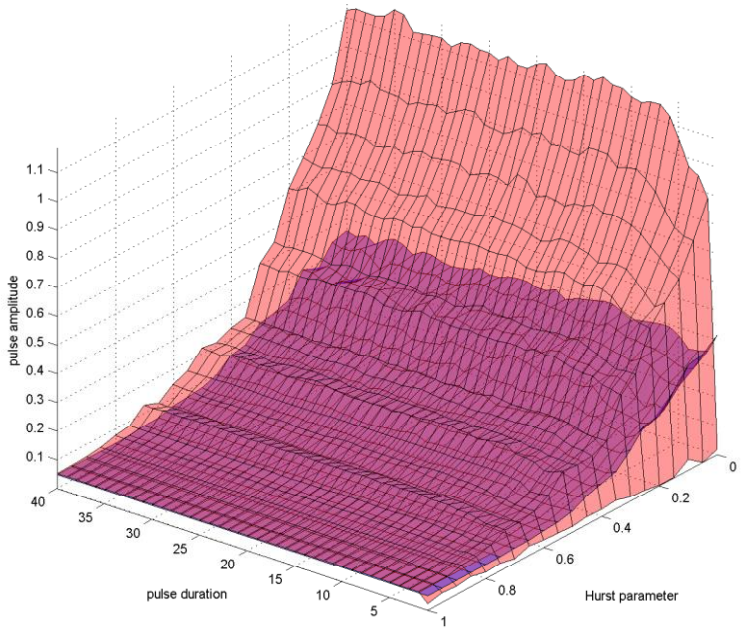


Figura 13. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 9. Pragul superior (roşu) este cel pentru falsuri pozitive.

Prima observație pe care o putem face este că sensibilitatea metodei SNEO este mai bună. Valorile maxime pentru cazul filtru și prag sunt mult mai mari și cea mai proastă sensibilitate la SNEO este sub raportul 1 constatat la impulsuri largi în cazul filtru și prag. In fapt, sensibilitatea de detecție este în cel mai rău caz în jur de 0.4-0.5, cu comportare mai proastă la filtrare mai agresivă (fereastra Bartlett mai lată). Sensibilitatea este maximă pentru parametrul Hurst cu valori mari. In fapt, sensibilitatea de detecție (lipsă falsuri negative) este probabil chiar mai bună decât se vede în diagramă pentru că primul prag testat este chiar la proporția 0.05 impuls/semnal. Se observă din diagramele consecutive că sensibilitatea de detecție este foarte asemănătoare un *k* dat, la o gamă largă a dimensiunii ferestrei folosite la convoluție.

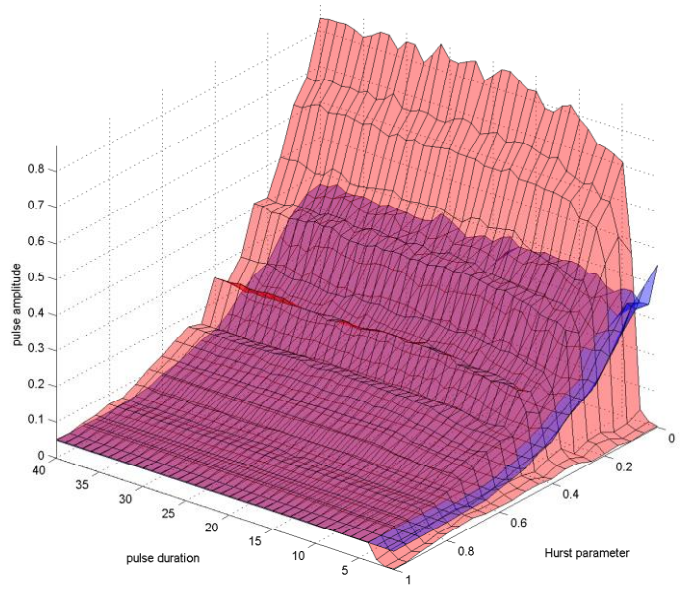


Figura 14. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 17. Pragul superior (roşu) este cel pentru falsuri pozitive.

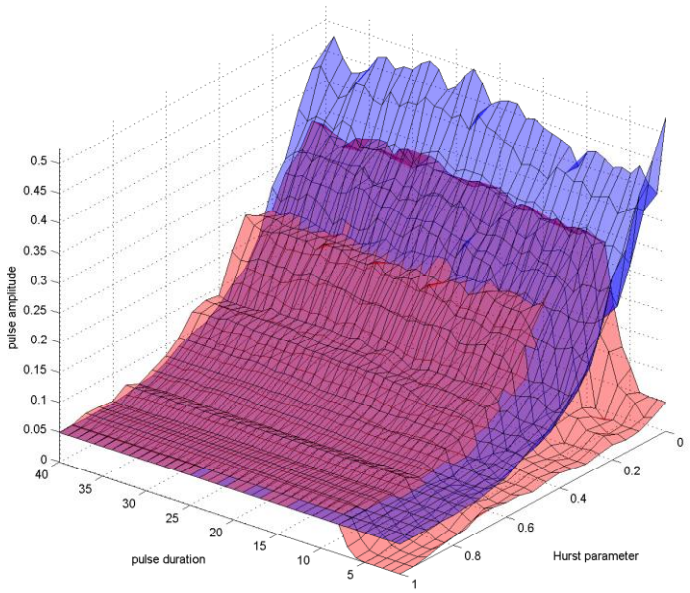


Figura 15. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 33. Pragul superior (albastru) în zona H=0..0.1 este cel pentru falsuri negative.

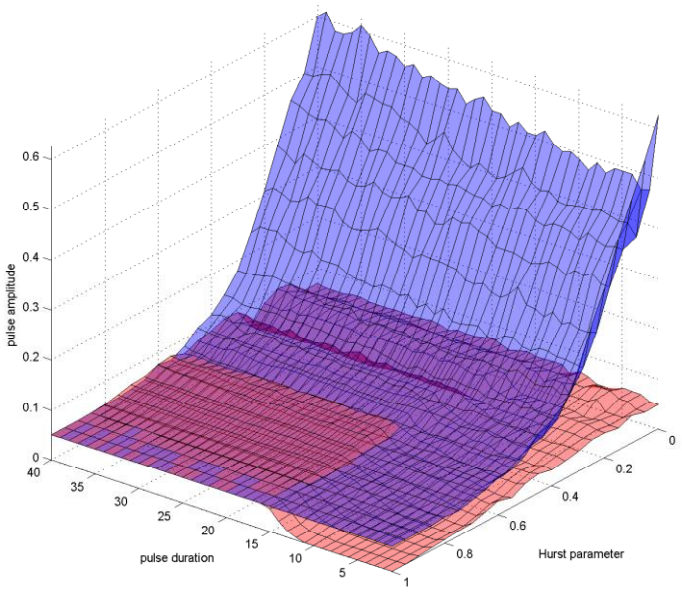


Figura 16. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=1 şi fereastră Bartlett 65. Pragul superior (albastru) este cel pentru falsuri negative.

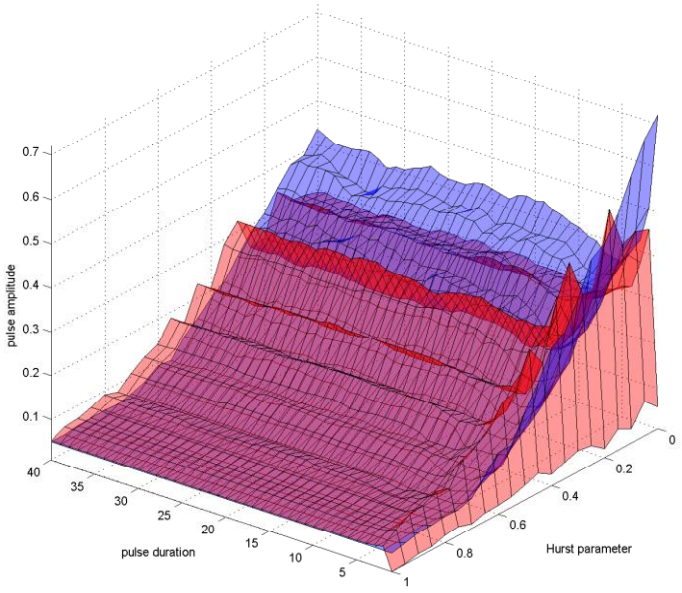


Figura 17. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO *k*=4 şi fereastră Bartlett 9. Pragul superior (albastru) în zona H=0..0.1 este cel pentru falsuri negative.

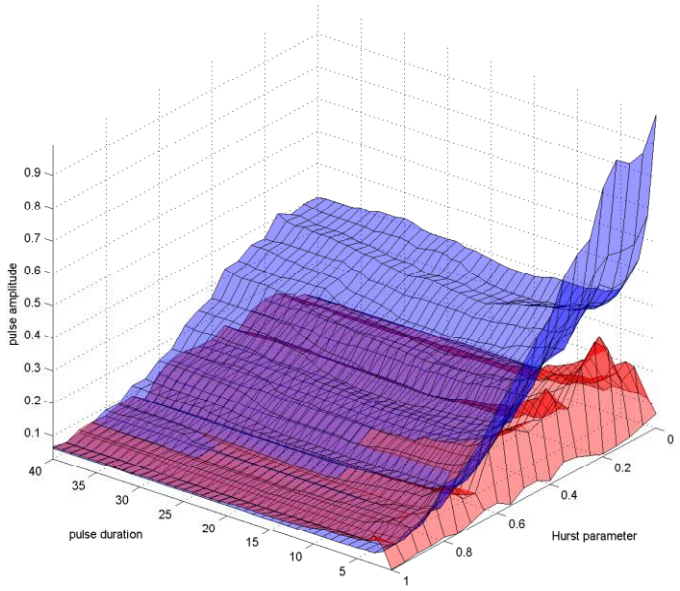


Figura 18. Praguri pentru amplitudinea impulsului perturbator la detecţia cu SNEO k=16 şi fereastră Bartlett 9. Pragul superior (albastru) este cel pentru falsuri negative.

O altă observaţie este că nu mai există o relaţie clară între pragurile de dispariţie a falsurilor pozitive şi a falsurilor negative. Aici valoarea ferestrei Bartlett devine importantă. Pentru o fereastră de lăţime redusă pragul de la care dispar falsurile pozitive este semnificativ deasupra pragului de detecţie (dispariţia falsurilor negative). Pe măsură ce mărim dimensiunea ferestrei efectul de filtrare creşte şi zgomotul aferent semnalelor cu valori mici pentru H este mai bine eliminat. La lăţimea ferestrei egală cu 33 pragul falsurilor pozitive devine parţial nerelevant şi la lăţimea ferestrei egală cu 65 pragul falsurilor pozitive este practic complet nerelevant (deci singurul factor limitator este sensibilitatea de detecţie, ca şi în cazul filtru + prag). Se observă foarte bine pe diagramele aferente ferestrelor mai mari cum rata de falsuri pozitive scade drastic în zona impulsurilor de lăţime scăzută (cu prag la un sfert din lăţimea ferestrei).

Modificarea spaţiului pe care aplică operatorul SNEO în forma sa generalizată din ecuaţia (10) are după cum ne aşteptam efect asupra sensibilităţii detecţiei la impulsuri de durată scăzută, mai exact de durată mai scurtă decât frecvenţa corespunzătoare operatorului SNEO(k). Se manifestă în acelaşi timp şi o scădere semnificativă a pragului aferent falsurilor pozitive pe întreaga gamă studiată. Nu am găsit o explicaţie directă pentru acest fenomen. Simpla comportare corelată cu frecvenţa corespunzătoare operatorului SNEO(k) nu ar trebui să fie suficientă pentru că seriile de date sunt anti-persistente chiar în zona în care se manifestă scăderea pronunţată a pragului (pentru valori ale lui H scăzute).

Mai observăm că există o îmbunătăţire locală a performanţelor în zona impulsurilor de lăţime corespunzătoare valorii lui k. Modul simplificat de aplicare a operatorului SNEO(k) nu este foarte potrivit pentru a evidenţia acest aspect dar comportamentul sugerează că o tehnică similară dar care să exploateze scalarea seriilor în timp ar putea să dea rezultate mai bune.

Concluzia studiului asupra celor două tipuri de mecanisme de detecție este că aplicarea SNEO este benefică atât pentru sensibilitate (lipsa falsuri negative) cât și pentru corectitudine (lipsa falsuri pozitive). Aplicarea unei ferestre de lățime rezonabilă (de exemplu 17, 33) are un efect bun asupra performanțelor.

Din păcate metoda SNEO are însă o comportare mai proastă la serii cu parametru Hurst scăzut, adică tocmai la tipul de semnal frecvent întâlnit ca rezultat al procesului nostru tipic de captură. Vom avea nevoie şi de alte tehnici de detecţie şi eventual de o combinaţie între acestea pentru a obţine rezultate aplicabile cu succes în practică.

4.2.3. Detecţie cu procesare timp-frecvenţă

Detecţia SNEO descrisă mai sus oferă rezultate bune, dar experimentele sugerează un rezultat care se potriveşte cu aşteptările teoretice şi anume că spaţiul pe care se aplică operatorul trebuie corelat cu lăţimea impulsului perturbator. Această constatare ne conduce spre necesitatea aplicării unor tehnici timp-frecvenţă care ar putea să ofere rezultate mai bune pentru o gamă mai largă a formei impulsului perturbator şi eventual pentru o gamă mai largă de valori ale parametrului Hurst. Am căutat aşadar metode care să folosească tehnici de analiză timp-frecvenţă la detecţia anomaliilor din seriile de date produse cu mecanismele de captură descrise anterior [66]

Tehnicile de analiză timp-frecvență au fost dezvoltate pentru cazul semnalelor nestaționare. Mecanismele de analiză din categoria transformatei Fourier pot să reprezinte un semnal în spațiul frecvență, evidențiind modul în care energia semnalului original este distribuită în frecvență. Dezavantajul acestui tip de analiză pentru cazul semnalelor nestaționare este că prin transformare se elimină dimensiunea timp, care în acest caz ne interesează. Valoarea constatată la o anumită frecvență este calculată prin integrarea peste tot spațiul timp deci semnalele care se pot analiza confortabil în acest mod sunt doar cele staționare.

Dacă dorim să analizăm semnale nestaționare avem nevoie de o reprezentare timp-frecvență care se desfășoară simultan atât pe dimensiunea timp cât și pe dimensiunea frecvență și dau o imagine a evoluției densității energiei pe ambele dimensiuni. O astfel de reprezentare este o distribuție timp-frecvență (TFD – Time-Frequency-Distribution) [67]. Putem să privim un semnal nestaționar ca un semnal cu frecvența instantanee variabilă în timp sau ca o sumă (eventual infinită) de astfel de semnale.

De-a lungul timpului au fost dezvoltate multiple metode de construcție a unor astfel de reprezentări: distribuția Wigner-Ville, transformări Fourier localizate în timp (STFT, transformarea Gabor), pachete de filtre, alte distribuții (Page, Rihaczek, Levin, Choi-Williams) [67]. Aceste metode permit obținerea unor funcții $\rho_s(t, f)$ care descriu densitatea de energie a semnalului în planul timp-frecvență dar fiecare are proprietăți ușor diferite și în particular fiecare introduce artifacte datorate aproximării pe care o fac asupra semnalului original. Pentru un semnal dat (sau mai degrabă pentru o clasă de semnale date) unele distribuții pot să aibă performanțe mai bune decât altele în ceea ce privește acuratețea reprezentării care în ultimă instanța se referă la rezoluția pe care o permit în reprezentarea conformă a semnalului.

Încă de la lucrarea originală a lui Gabor prin care introduce transformarea care îi poartă numele s-a evidențiat faptul că există o limitare în ceea ce privește posibilitatea de a obține simultan rezoluții bune în timp și în frecvență. Această

limitare este similară cu principiul de incertitudine al lui Heisenberg și introducerea ei în contextul analizei timp-frecvență este de așteptat pentru că Gabor și-a desfășurat activitatea în contextul fizicii teoretice.

In lucrarea sa din 1946 Gabor împarte planul timp-frecvență în domenii rectangulare numite de el *logon* cu dimensiunea în timp Δt și dimensiunea în frecvență Δf , care trebuie să satisfacă relația de incertitudine:

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{11}$$

Elementele în care se descompune planul timp-frecvență pot fi uniforme, așa cum au fost ele introduse original sau pot fi de forme diferite, ajustate la scopul analizei dorite. În particular, ele pot fi aranjate de așa natură încât la capătul spre frecvențe scăzute să avem o întindere mare pe direcția timp și la capătul spre frecvențe înalte să avem o rezoluție mai bună în timp (cu prețul unei reduceri de rezoluție în frecvență.

Această direcție de analiză permite o analiză multirezoluție. Anumite componente ale semnalului sunt analizate cu o rezoluție scăzută în timp, altele cu o rezoluție mai ridicată, astfel încât să putem captura atât componentele semnalului care au o variație lentă cât și cele care au o variație rapidă, păstrând nivelul optim de rezoluție pe axa timp.

Conform prezentării făcute de Mallat [68] se poate privi transformarea Fourier continuă a unui semnal definit pe un interval finit ca o descompunere a semnalului original peste o bază ortonormală $\left\{e^{i2\pi mt}\right\}_{m\in Z}$. Și pentru cazul discret se poate face descompunerea, pe baza ortogonală $\left\{e^{i2\pi kn/N}\right\}_{0\leq k<N}$, descompunere care conduce la transformarea Fourier rapidă.

Important din punctul nostru de vedere este continuarea acestei idei și anume faptul că se poate face o descompunere după alte baze, în particular baze care au o comportare multirezoluție. Original Haar introduce în 1910 un set de funcții care au comportament de bază ortonormală, după care contribuțiile succesive ale lui Morlet și Grossman, Strömberg, Meyer, Daubechies și Mallat au condus la teoria construcției bazelor ortonormale wavelet și algoritmii de descompunere rapidă pentru semnale discrete.

In cazul pe care dorim să îl analizăm impulsul perturbator poate fi descompus ca o sumă de componente spectrale peste o gamă largă de frecvențe.Banda peste care sunt împrăștiate componentele semnificative rezultate din descompunerea impulsului depinde de lungimea acestuia dar și de aspectul fronturilor sale, conform relaţiei binecunoscute bazată pe modelul unui circuit simplu RC:

$$BW = \frac{0.35}{T_r} \tag{12}$$

unde BW este banda şi T_r este durata frontului de durată nenulă al impulsului. Rezultă că un impuls de durată scurtă cu fronturi rapide va fi mai uşor de detectat pentru că se manifestă pe o gamă mai largă de componente spectrale.

In același timp însă ne interesează și poziționarea pe dimensiunea timp a componentelor de energie aferente impulsului perturbator pentru că dorim să

determinăm și când a apărut anomalia. Așadar pentru a detecta impulsul ar trebui să descompunem semnalul dat de seria de date într-o distribuție timp-frecvență și apoi să urmărim în acest spațiu concentrarea de energie pe o bandă largă cauzată de impulsul perturbator.

In fapt, trebuie să observăm că problema pe care dorim să o rezolvăm pare a fi ceva mai simplă. Analiza pe care o facem este una în timp real. Seriile de date sunt continuu generate și trebuie să detectăm apariția anomaliei cât mai repede posibil. In acest sens, deși am putea avea un plan timp-frecvență care să se întindă teoretic la infinit în toate direcțiile, de fapt anomalia pe care o căutăm se va afla întotdeauna in partea dreaptă a axei timp. Din această cauză aparent am putea să ne mulțumim și cu o descompunere în frecvență doar pentru o fereastră limitată de timp, fără să facem o analiză timp-frecvență.

Observația nu este întru totul adevărată. Așa cum arătam în analiza inițială a tipurilor de perturbații acestea pot să aibă atât prezentare de impuls de amplitudine mare concentrat pe o durată redusă de timp cât și prezentare de impuls de amplitudine mai scăzută, întins pe o durată mai mare de timp, dar ambele având o energie semnificativă față de traficul normal, așa încât să avem o șansă să le detectăm.

4.2.4. Detecţie cu descompunere wavelet şi cumulare de praguri

Am ales să analizăm seria de valori care conţine trafic normal şi impuls perturbator folosind descompunere wavelet discretă (DWT). Datorită spectrului larg al impulsului perturbator ne aşteptăm să regăsim în urma descompunerii wavelet componente semnificativ mărite pe multiple scări de expandare (adică pe multiple benzi de frecvenţă) în zona în care se află perturbaţia [68]. Aşadar schema de analiză este cea din Figura 19.

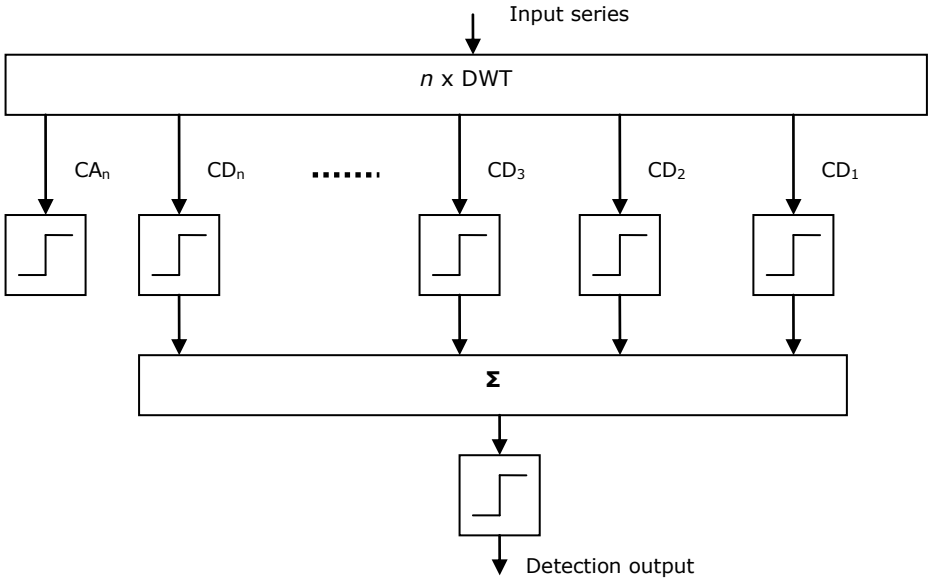


Figura 19. Schema de analiză folosind descompunere wavelet şi cumulare de praguri pe coeficienţi detaliu.

Se descompune semnalul dat de seria de valori folosind *n* aplicări de DWT într-o serie de valori care conţine coeficienţi de aproximare la nivel *n* CA_n şi *n* serii de valori care conţin coeficienţi de detaliu la nivele 1 până la *n* (CD_1 ... CD_n). Datorită decimării, lungimea seriilor CD_n scade odată cu creşterea lui *n*. Procesul de sumare ține seama de aceasta și scalează înapoi seriile astfel încat lungimea seriei sumate să fie din nou egală cu lungimea lui CD_1 (aproximativ jumătate din lungimea seriei inițiale, în funcție de modul de tratare a condițiilor de margine).

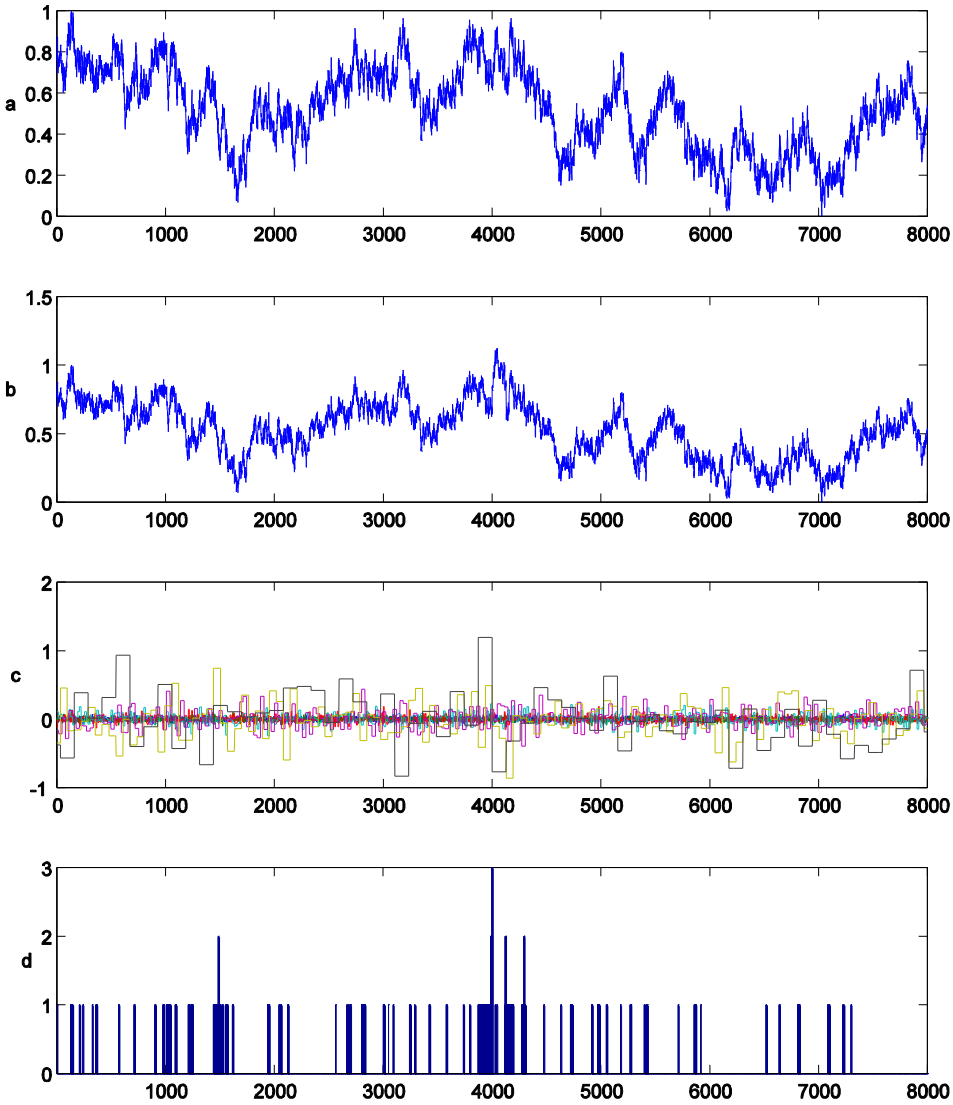


Figura 20. Detecţie cu descompunere wavelet, praguri şi cumulare de decizie. a – serie inițiala (Hurst=0.3); b – serie perturbată cu impuls unitar de amplitudine relativă 0.2 și durata 128; c – descompunere wavelet (Meyer, n=7); d – rezultat la ieșirea sumatorului, înainte de prag final.

Fiecare serie de coeficienți detaliu care reprezintă echivalent importanța componentei semnal în banda corespunzătoare de frecvență este supusă unei detecții simple de prag care generează valoarea 0 în zonele cu amplitudine scăzută (considerată nesemnificativă) și 1 în zonele cu amplitudine semnnificativă. Setul de serii de detecție parțială se scalează înapoi pentru a păstra coerența în domeniul timp și se sumează, producând o serie Σ cu valori între 0 și *n*. Pe această serie ne așteptăm ca valorile maxime să fie produse în zonele în care există prezență semnificativă de semnal pe toate benzile de frecvență (sau multe din ele), așadar aplicăm încă o detecție cu prag pentru izolarea momentelor de timp unde se află anomaliile.

Procesul se poate implementa și folosind SWT (descompunere fără decimare), caz în care nu mai este necesară rescalarea componentelor detaliu înainte de sumare. Un exemplu de astfel de procesare este prezentat în Figura 20. Detecția cu prag aplicată la fiecare serie de coeficienți de detaliu este 3σ.

Se observă prin comparație cu Figura 10 că amplitudinea impulsului perturbator este mult mai mică, astfel încât prezența impulsului este insesizabilă vizual pentru un observator neavizat. Cu toate acestea, 3 din cele 7 serii de coeficienți de detaliu au detectat valori anormale în zona impulsului astfel încât putem să aplicăm o detecție cu prag pe ieșirea sumatorului și să detectăm cu succes anomalia de tip impuls unitar aplicată.

Cazurile explorate până acum au fost analizate pentru impulsuri dreptunghiulare ideale (perturbația atinge valoarea maximă de la un eșantion la următorul). Această structură a perturbației are componente spectrale senmificative în zona frecvențelor înalte și explică plasarea punctelor în care se face detecția la începutul și/sau sfârșitul perturbației. Pentru a verifica această ipoteză am testat metoda de mai sus (descompunere wavelet și cumulare de decizie) pentru un semnal impuls perturbator care are nevoie de 4 eșantioane să atingă valoarea maximă și respectiv să revină la valoarea zero la finalul perioadei perturbate. Numărul maxim de serii care detectează s-a redus la 2 și deși există o densitate mai mare de decizii pozitive în zona perturbației, există semnificativ de multe falsuri pozitive în afara zonei perturbate. Pentru a reveni la o calitate comparabilă a deciziei cu situația din Figura 20 a fost necesară creșterea amplitudinii relative a perturbației la 0.35 (față de 0.2).

4.2.5. Detecţie cu denoising incomplet pe bază de wavelet

Din analiza calitativă a metodei descrise mai sus am ajuns la presupunerea că aplicarea de praguri discrete (cu rezultat 0/1) poate să piardă din sensibilitate în defavoarea capabilității de detecție. Pentru a rezolva această problema, am testat o metodă ușor diferită, care este înrudită cu tehnicile de eliminare a zgomotului folosind descompunerea wavelet.

Pentru eliminarea zgomotului (denoising) folosind descompunerea wavelet se aplică următorul algoritm:

- se descompune semnalul original prin aplicarea DWT de *n* ori în seria de coeficienți de aproximare și un set de serii de coeficienți de detaliu
- se anulează selectiv o parte din coeficienții de detaliu din fiecare serie. Pragul peste care se conservă coeficienții poate fi prefixat sau poate fi calculat din valorile coeficienţilor.
- se reconstruieşte semnalul din coeficienţii de aproximare şi coeficienţii de detaliu

Detecţia impulsului perturbator trebuie să izoleze componente de amplitudine semnificativă în zona frecvenţelor relativ ridicate (deci la scări de expansiune relativ reduse) şi să ignore semnalul original. Pentru a obţine acest efect folosim o variantă adaptată a algoritmului de denoising. Diferenţa este că la reconstrucţie nu mai folosim şi coeficienţii de aproximare, eliminând astfel eficient semnalul inițial. Metoda este prezentată în Figura 21.

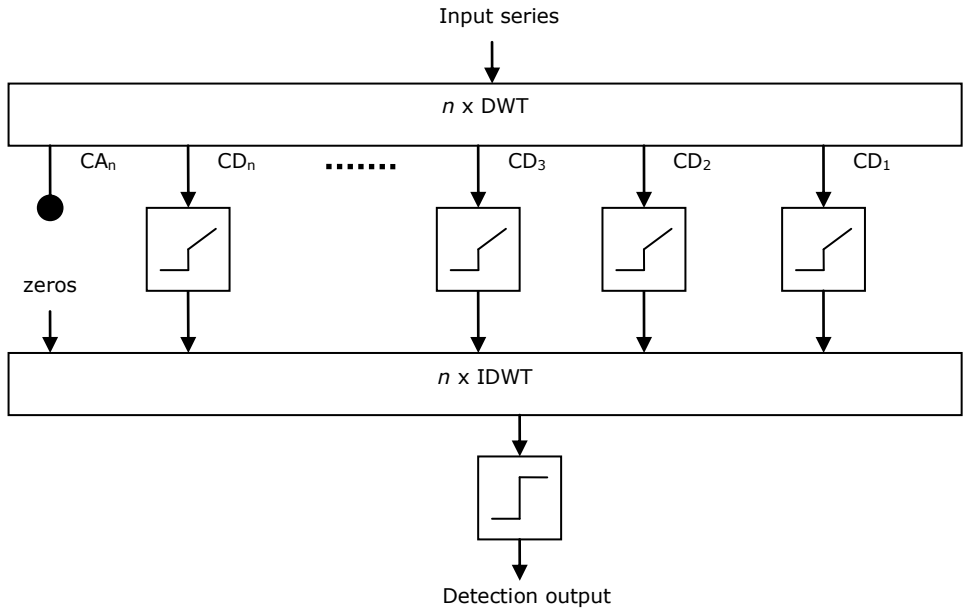


Figura 21. Schema de analiză folosind o variantă de denoising incomplet pe bază de wavelet şi praguri.

In figură se observă că după descompunere de nivel *n* se elimină complet coeficienții de aproximare și se aplică praguri pe fiecare serie de coeficienți de detaliu astfel încât să păstrăm doar valorile de amplitudine mai mare decât un prag. In implementarea testată am folosit pentru decizia de conservare a coeficienților praguri simple la 3-sigma și praguri tip SNEO(1) fără fereastră de filtrare.

Setul de serii de coeficienți de detaliu (si seria de valori zero pentru coeficienții de aproximare, necesară pentru consistența algoritmului) sunt apoi folosite la reconstrucția unui semnal (seria de valori) care ar trebui să conțină valori semnificative în zona impulsului perturbator. Acest semnal este supus unei detecții simple de prag 6-sigma pentru a produce ieșirea care marchează detecția anomaliei. Procesul este exemplificat în Figura 22.

Se observă vizual că eficienţa metodei este mai bună decât a metodei cu praguri și sumator de decizie. Semnalul original este același în ambele cazuri și nivelul de perturbație este același. Prezenţa impulsului perturbator induce componente semnificative pe seriile de coeficienţi şi semnalul reconstruit conţine valori detectabile în zona acestuia dar are şi componente (mai puţin semnificative) în alte porţiuni, corespunzătoare unor potenţiale falsuri pozitive. Din acest motiv am constatat că e benefic să ridicăm pragul de detecţie final la minim 6-sigma.

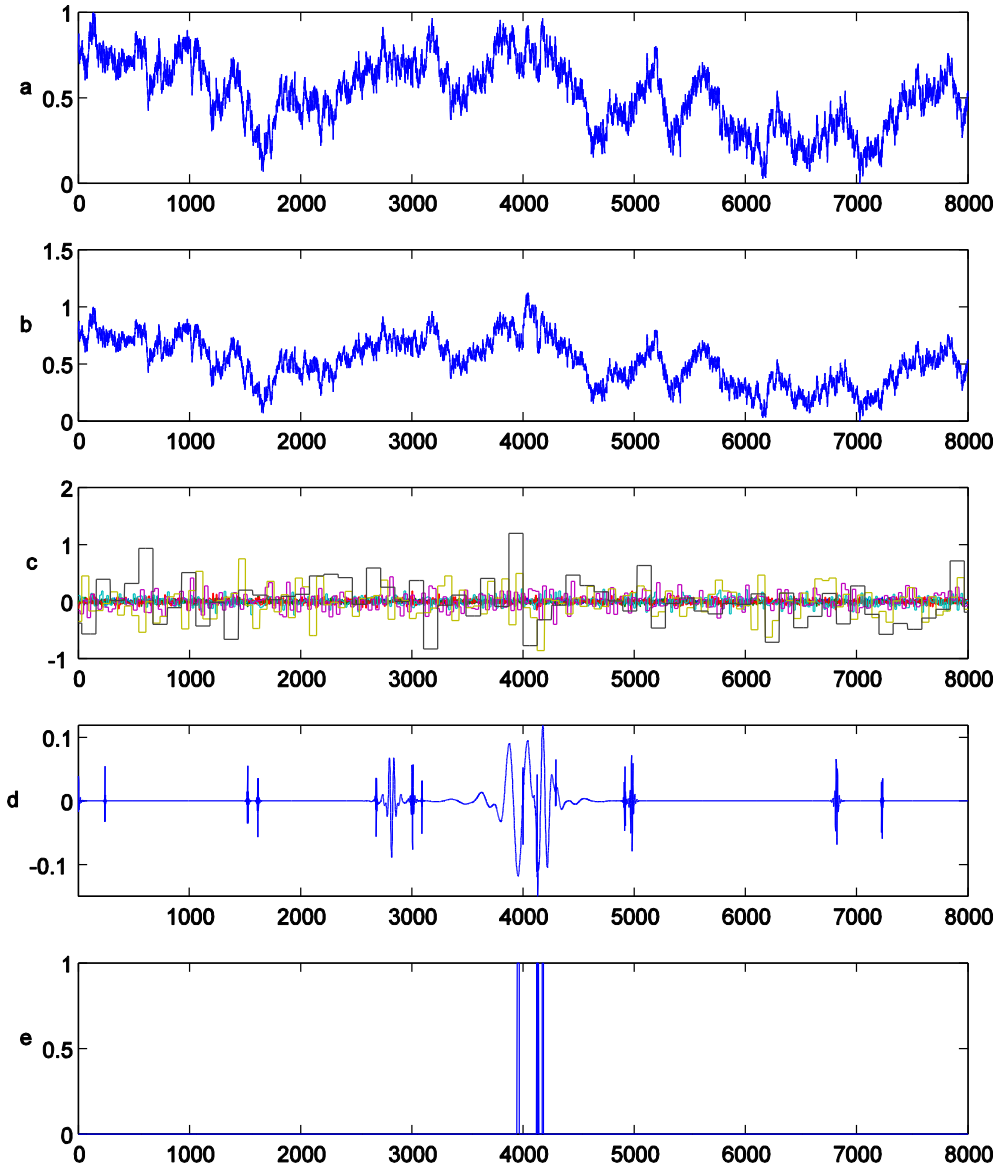


Figura 22. Detecţie cu denoising incomplet tip wavelet şi praguri. a – serie iniţiala (Hurst=0.3); b – serie perturbată cu impuls unitar de amplitudine relativă 0.2 și durata 128; c – descompunere wavelet (Meyer, n=7); d – semnal reconstruit; e – ieşire detector la 6-sigma.

Din păcate se observă şi un efect de incertitudine asupra momentului perturbaţiei, introdus de caracterul simetric al funcţiei wavelet folosite. Am ales

funcţia Meyer pentru experimentele iniţiale pentru că are un caracter simetric şi este intuitiv potrivită cu forma perturbaţiei deci ar trebui să ofere performanţe mai bune. Caracterul ei simetric și zona largă de suport însă induce detecţia pe poziţii temporale anterioare impulsului perturbator, ceea ce e un efect mai puţin potrivit. Din pacate nu e vorba de o detecţie antecauzală, oricât de interesant ar fi un astfel de fenomen, pentru că noi analizăm postfactum o întreagă zonă de semnal inclusiv după apariția perturbației, în acest caz experimental chiar mult după apariția perturbației. Impactul acestui efect asupra experiențelor următoare este că trebuie să lărgim zona de gardă în jurul impulsului perturbator atunci când evaluăm răspunsul la falsuri pozitive. Prin lărgirea zonei de gardă nu afectăm acuratețea evaluării, pentru că oricum falsuri pozitive înseamnă „detecții în lipsa elementului perturbator".

Din impactul simetriei funcției wavelet asupra momentului detecției rezultă că o altă dimensiune care trebuie evident explorată în experiențele care urmează este gradul de adecvare a diverselor funcții wavelet pentru procesele de detecție descrise.

Am evaluat experimental și impactul apariției unui impuls cu fronturi non-ideale asupra eficienței detectorului cu denoising parțial. Pentru cazul particular al semnalului din exemplu introducerea unor fronturi de 4 eșantioane a fost fără impact asupra calității detecției. De fapt nici transformarea impulsului dreptunghiular în impuls triunghiular (64 eșantioane creștere și 64 descreștere) nu au afectat eficiența detecției în acest caz.

Pentru a confirma performanțele metodelor de detecție pe bază de wavelet am derulat teste pentru diferite valori ale lățimii impulsului și ale parametrului Hurst, în aceleași condiții ca și pentru metodele cu prag. Configurația similară a testelor ne permite să facem comparații calitative între metode. Din păcate performanța ca timp de rulare este mult mai scăzută pentru metodele care folosesc wavelet din cauza complexității mult sporite a calculelor așa că a fost nevoie să facem unele compromisuri pentru a păstra timpii de execuție în limite rezonabile. Am testat doar lățimi ale impulsului din 2 in 2 unități pe aceeași gamă de valori și am folosit un cache de semnale pregenerate pentru diverse valori Hurst. Nici unul din aceste compromisuri nu afectează acuratețea rezultatelor.

Prima serie de rezultate este pentru metoda cu praguri la coeficienții de detaliu și cumulare de decizie. Am rulat simularea pentru mai multe tipuri de funcții wavelet (Meyer, Daubechies, Symlet, Coiflet, Biortogonal). In Figura 23 pana la Figura 36 se regăsesc rezultatele câtorva din aceste simulări, limitate pe axa Z la gama 0..2 pentru a putea face mai ușor comparație între ele. Din analiza vizuală calitativă a diagramelor se constată că rezultatele sunt comparabile, cu un avantaj sesizabil pentru funcțiile biortogonale. Se constată o uniformitate pronunțată a pragului de detecție (falsuri negative), cu o sensibilitate bună (sub 0.4 relativ în cel mai rău caz, mult mai bună decât la metodele simple, bazate pe praguri). Se constată de asemenea o sensibilitate la falsuri pozitive pentru parametru Hurst mic și impulsuri înguste. Există posibilitatea ca această sensibilitate să fie datorată indirect modului de evaluare folosit la simulare, dependent de deviația standard a semnalului global. In orice caz, pentru aceeași metodă de evaluare, calitatea detecției cu wavelet cu cumulare de decizie este clar superioară metodelor simple cu prag.

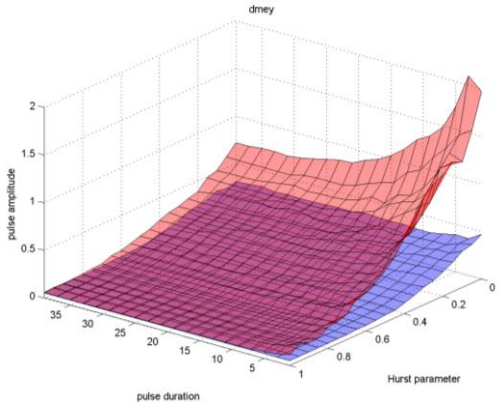


Figura 23. Performanţe la metoda cu praguri şi cumulare, wavelet Meyer discret, 7 nivele

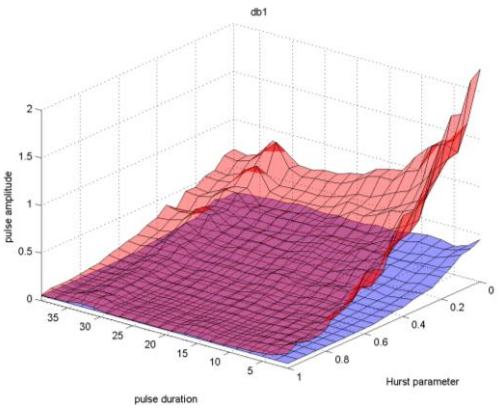


Figura 24. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-1, 7 nivele

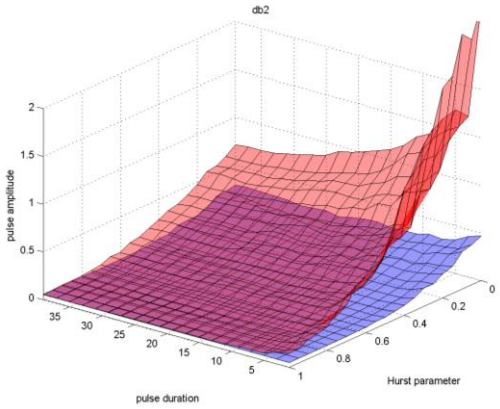


Figura 25. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-2, 7 nivele

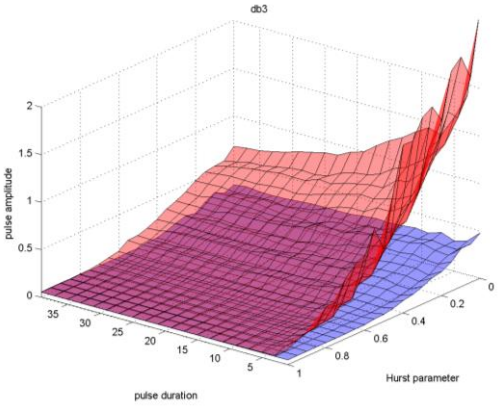


Figura 26. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-3, 7 nivele

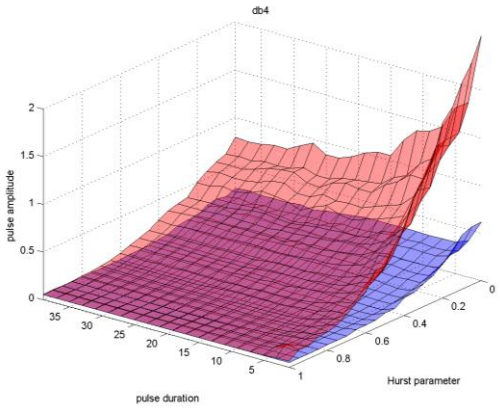


Figura 27. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-4, 7 nivele

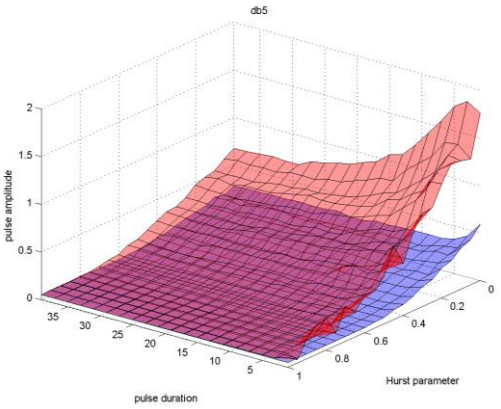


Figura 28. Performanţe la metoda cu praguri şi cumulare, wavelet Daubechies-5, 7 nivele

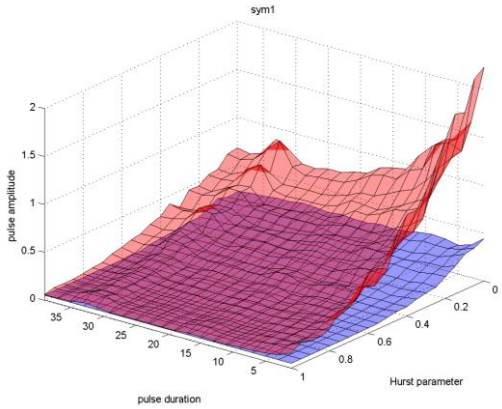


Figura 29. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-1, 7 nivele

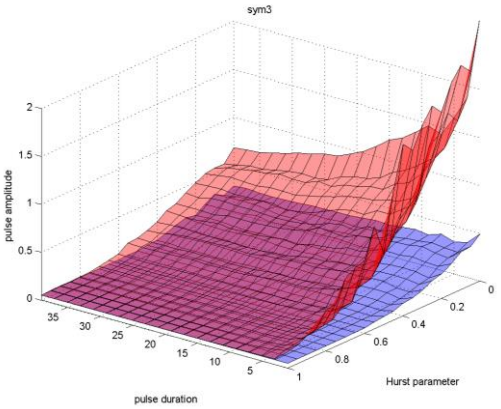


Figura 30. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-3, 7 nivele

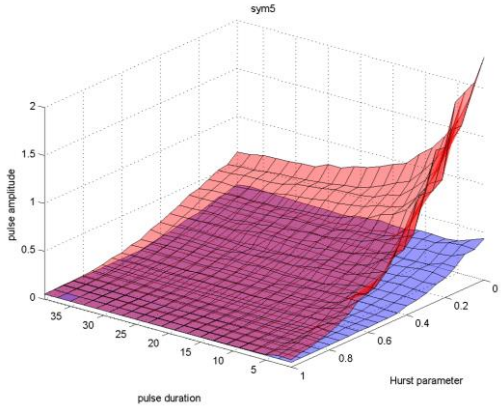


Figura 31. Performanţe la metoda cu praguri şi cumulare, wavelet Symlet-5, 7 nivele

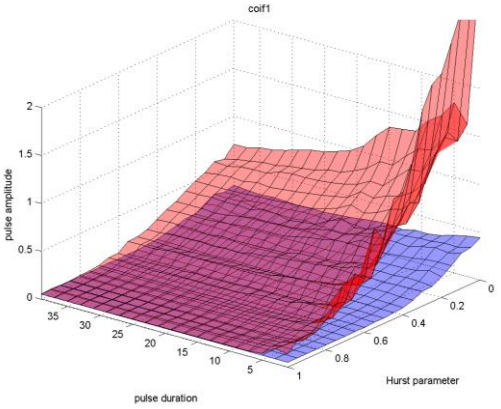


Figura 32. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-1, 7 nivele

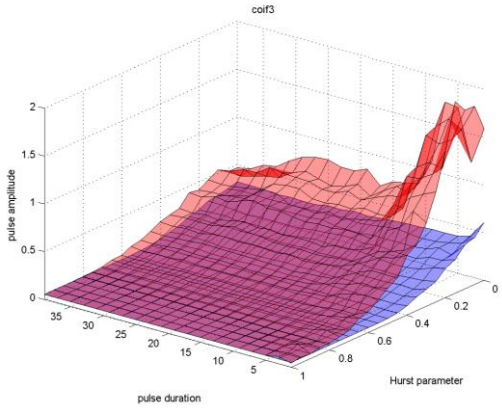


Figura 33. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-3, 7 nivele

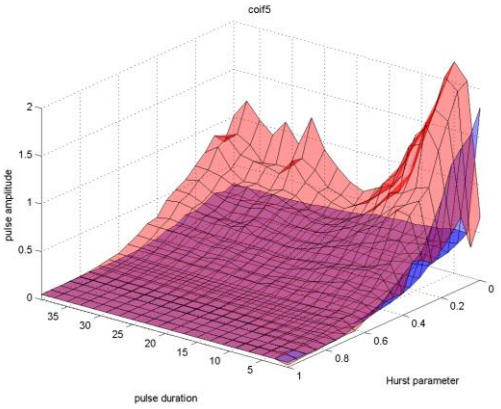


Figura 34. Performanţe la metoda cu praguri şi cumulare, wavelet Coiflet-5, 7 nivele

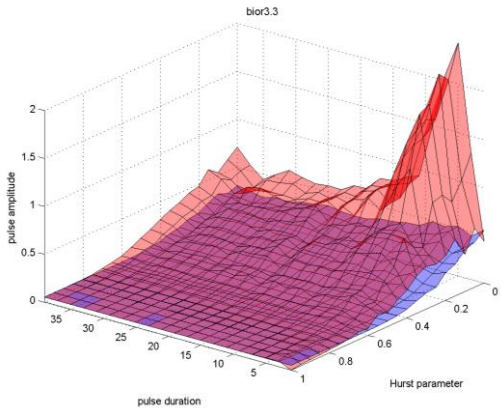


Figura 35. Performanţe la metoda cu praguri şi cumulare, wavelet Biortogonal-3.3, 7 nivele

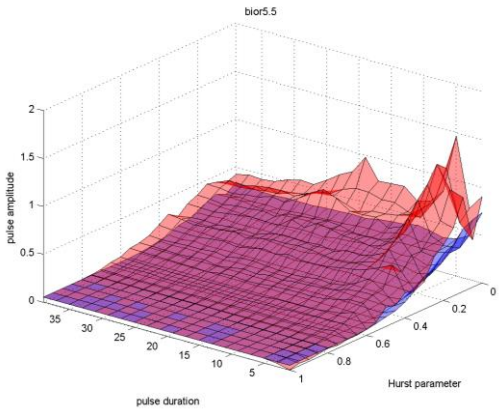


Figura 36. Performanţe la metoda cu praguri şi cumulare, wavelet Biortogonal-5.5, 7 nivele

Am reluat aceeaşi serie de simulări şi pentru metoda cu denoising parţial şi aplicare de prag. Pentru această metodă, din evaluările vizuale experimentale am constatat că există o puternică accentuare a impulsurilor în semnalul produs după reconstrucţie, aşa că am efectuat o serie de simulări pentru pragul de detecţie final 6-sigma, ca termen de comparaţie cu metoda anterioară şi respectiv pentru pragul de detecţie final 10-sigma, valoare care am ales-o în urma unor experimente iniţiale cu diferite valori. Trebuie să observăm că în contextul acestor metode, referinţa la deviaţia standard nu mai are aceeaşi semnificaţie statistică ci mai degrabă este un termen de calibrare a detecţiei cu prag. Din acest motiv deşi 10-sigma probabil că pare a fi o valoare exagerată, ea este eficientă din punct de vedere al rezultatelor obţinute.

In Figura 37 până la Figura 50 se regăsesc rezultatele pentru metoda cu denoising parțial și aplicare de prag, cu nivel de prag 6-sigma. Pragul falsurilor pozitive sunt reprezentate de suprafaţa roşie (tipic suprafaţa superioară, pentru cazul în care diagramele sunt alb-negru) iar pragul falsurilor negative este reprezentat de suprafaţa albastră (tipic suprafaţa inferioară, pentru cazul alb-negru). Se observă că performanțele de falsuri negative sunt în continuare bune și relativ uniforme dar performanțele de falsuri pozitive nu prezintă o îmbunătățire față de metoda anterioară, ceea ce justifică nevoia de a mări pragul de detecție la 10-sigma.

Se observă o creştere pronunţată a pragului de falsuri pozitive în zona valorilor Hurst scăzute şi a impulsurilor de durată scăzută (înguste). Fenomenul este legat pe de o parte de caracterul mai pronunţat de variabilitate al semnalelor anti-similare şi de degradarea raportului între energia impulsului pe care dorim să îl detectăm şi energia componentelor rapid variabile ale fondului.

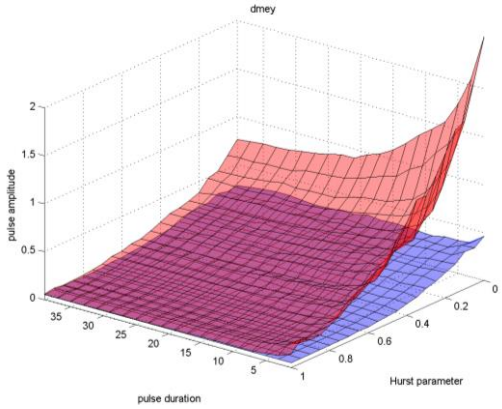


Figura 37. Performanţe la metoda cu denoising parţial, wavelet Meyer discret, 7 nivele, prag 6-sigma

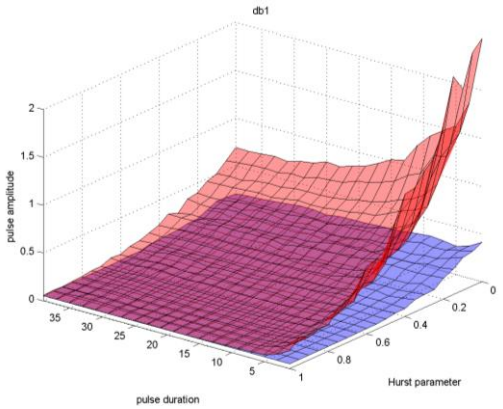


Figura 38. Performanţe la metoda cu denoising parţial, wavelet Daubechies-1, 7 nivele, prag 6-sigma

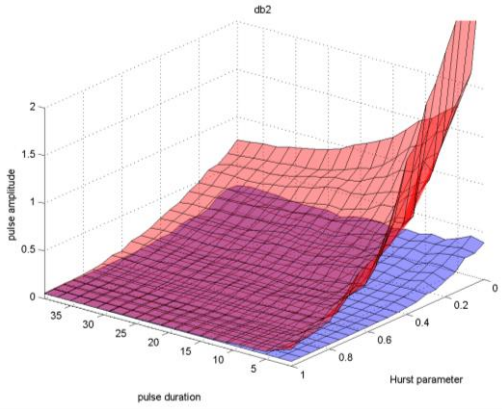


Figura 39. Performanţe la metoda cu denoising parţial, wavelet Daubechies-2, 7 nivele, prag 6-sigma

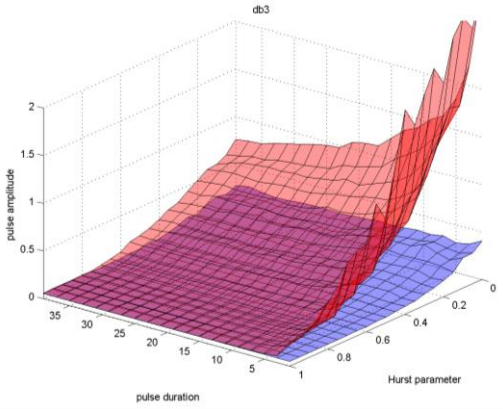


Figura 40. Performanţe la metoda cu denoising parţial, wavelet Daubechies-3, 7 nivele, prag 6-sigma

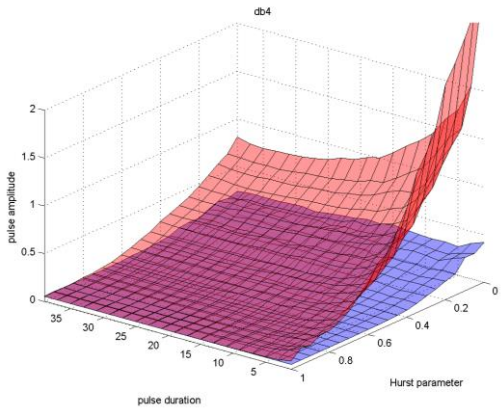


Figura 41. Performanţe la metoda cu denoising parţial, wavelet Daubechies-4, 7 nivele, prag 6-sigma

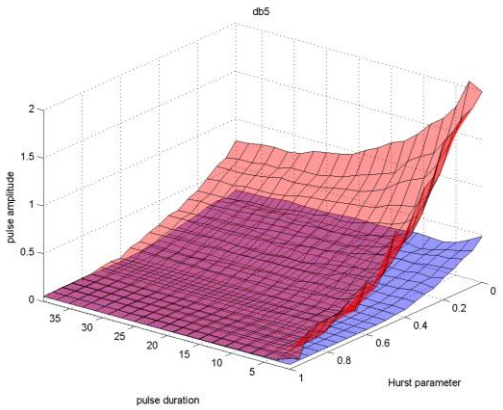


Figura 42. Performanţe la metoda cu denoising parţial, wavelet Daubechies-5, 7 nivele, prag 6-sigma

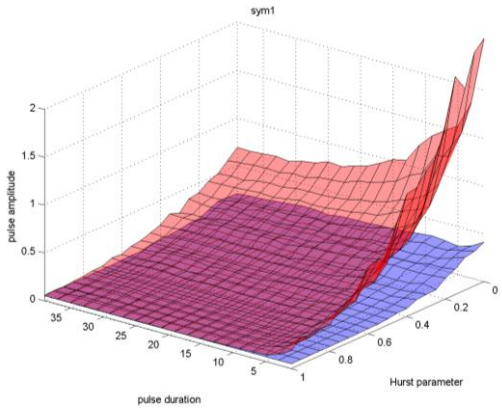


Figura 43. Performanţe la metoda cu denoising parţial, wavelet Symlet-1, 7 nivele, prag 6-sigma

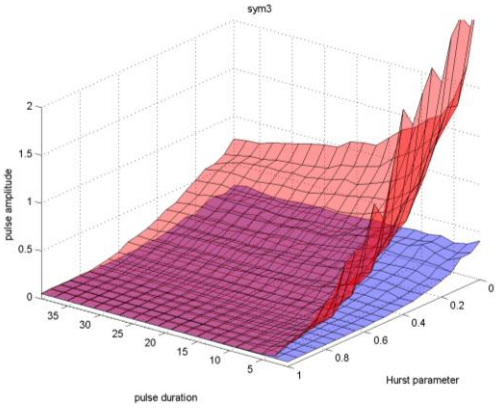


Figura 44. Performanţe la metoda cu denoising parţial, wavelet Symlet-3, 7 nivele, prag 6-sigma

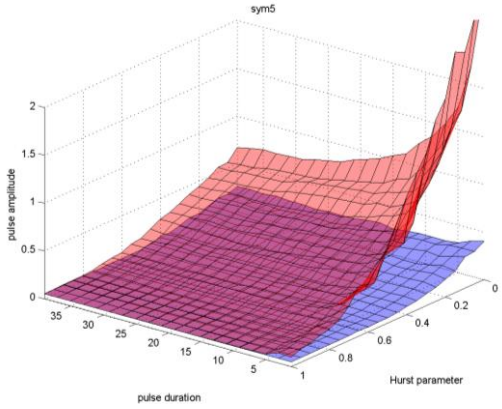


Figura 45. Performanţe la metoda cu denoising parţial, wavelet Symlet-5, 7 nivele, prag 6-sigma

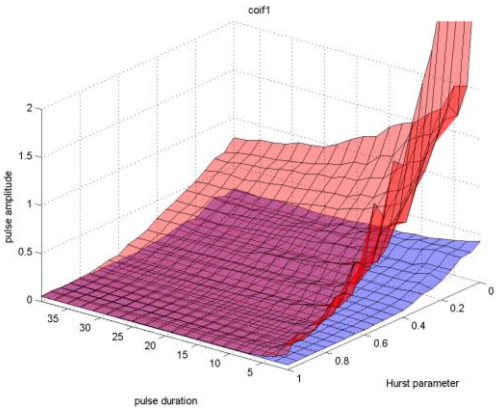


Figura 46. Performanţe la metoda cu denoising parţial, wavelet Coiflet-1, 7 nivele, prag 6-sigma

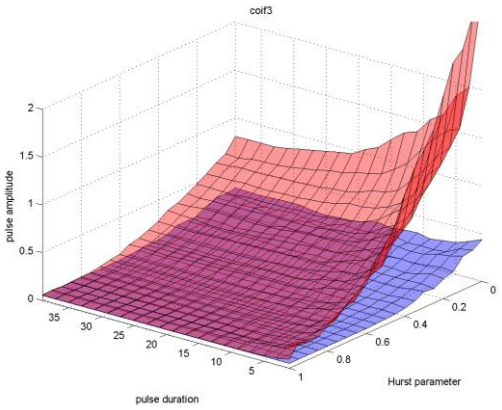


Figura 47. Performanţe la metoda cu denoising parţial, wavelet Coiflet-3, 7 nivele, prag 6-sigma

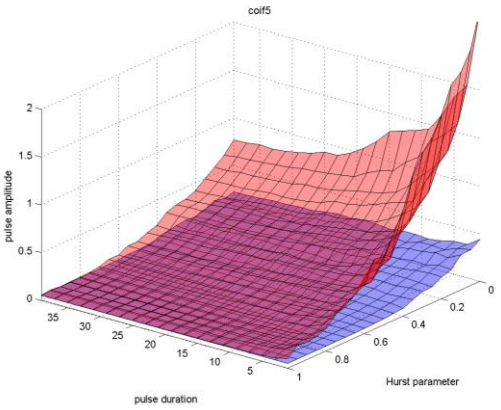


Figura 48. Performanţe la metoda cu denoising parţial, wavelet Coiflet-5, 7 nivele, prag 6-sigma

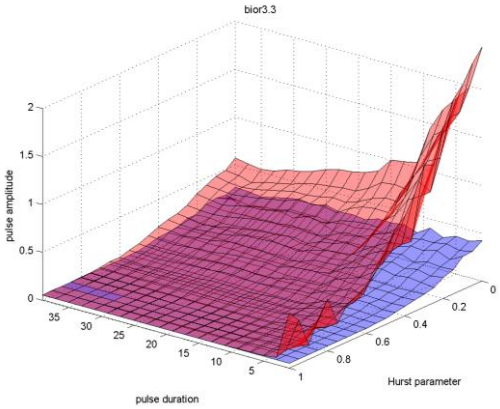


Figura 49. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-3.3, 7 nivele, prag 6-sigma

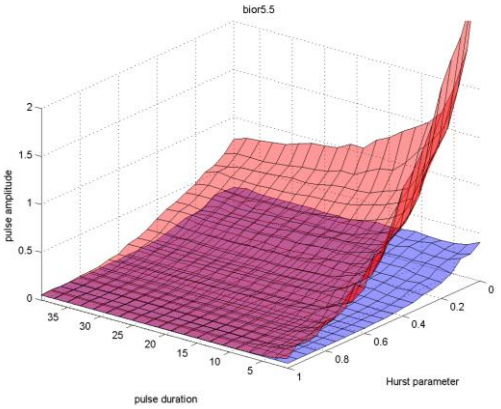


Figura 50. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-5.5, 7 nivele, prag 6-sigma

In Figura 51 până la Figura 64 se regăsesc rezultatele simulărilor pentru nivel de prag 10-sigma. Nivelul de performanță pentru falsuri negative este în continuare bun, iar pentru performanța de falsuri pozitive constatăm o îmbunătățire semnificativă. In fapt, pentru anumite cazuri pragul falsurilor pozitive scade pe alocuri sub pragul falsurilor negative. Aşa cum era de aşteptat, aspectul suprafeţelor nu este fundamental modificat, adică în continuare avem un impact al valorilor reduse pentru parametrul Hurst şi al impulsurilor de scurtă durată. Și în acest caz se constată o comportare bună a funcțiilor wavelet biortogonale.

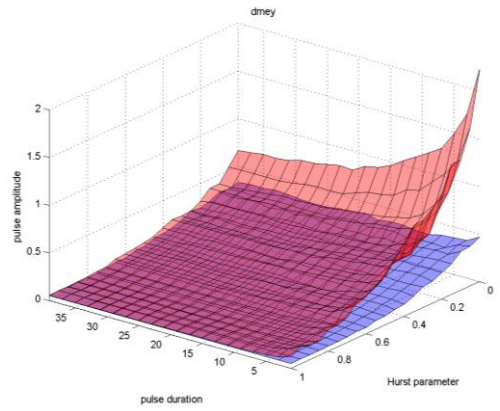


Figura 51. Performanţe la metoda cu denoising parţial, wavelet Meyer discret, 7 nivele, prag 10-sigma

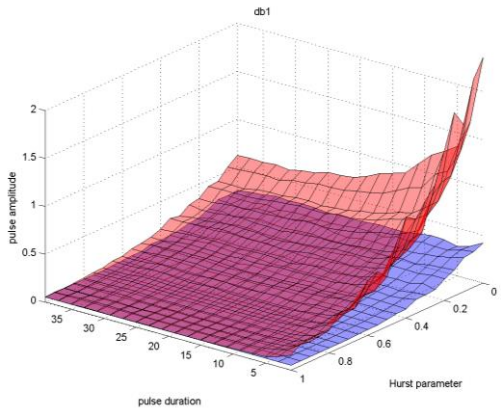


Figura 52. Performanţe la metoda cu denoising parţial, wavelet Daubechies-1, 7 nivele, prag 10-sigma

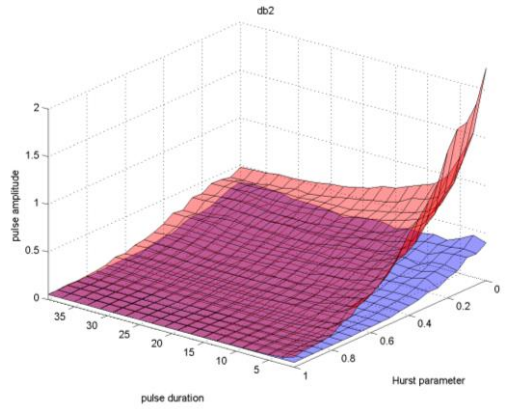


Figura 53. Performanţe la metoda cu denoising parţial, wavelet Daubechies-2, 7 nivele, prag 10-sigma

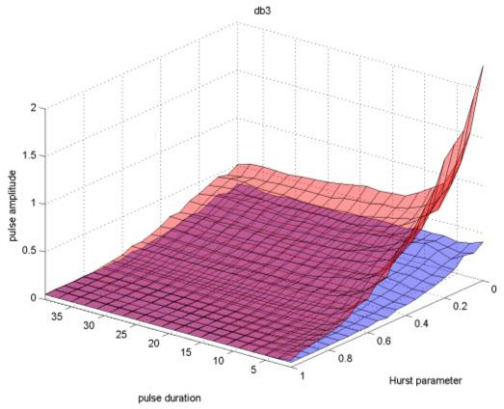


Figura 54. Performanţe la metoda cu denoising parţial, wavelet Daubechies-3, 7 nivele, prag 10-sigma

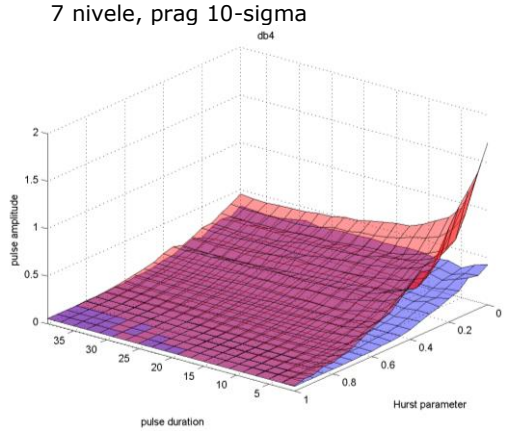


Figura 55. Performanţe la metoda cu denoising parţial, wavelet Daubechies-4, 7 nivele, prag 10-sigma

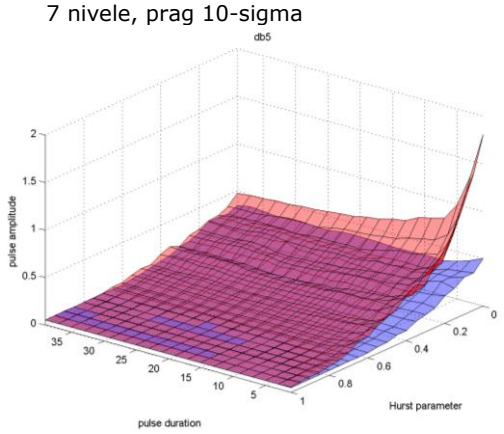


Figura 56. Performanţe la metoda cu denoising parţial, wavelet Daubechies-5, 7 nivele, prag 10-sigma

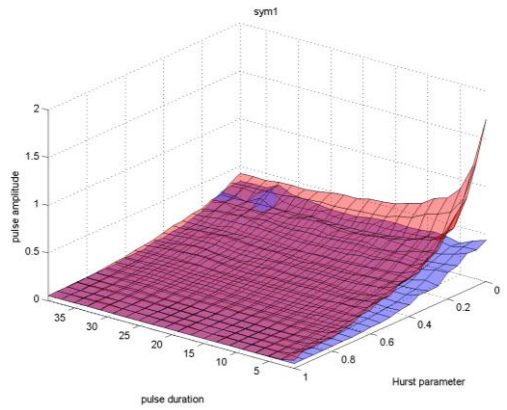


Figura 57. Performanţe la metoda cu denoising parţial, wavelet Symlet-1, 7 nivele, prag 10-sigma

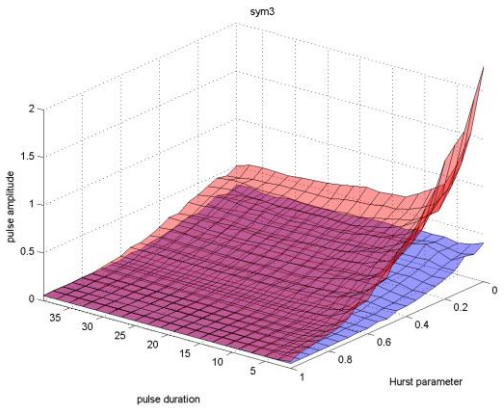


Figura 58. Performanţe la metoda cu denoising parţial, wavelet Symlet-3, 7 nivele, prag 10-sigma

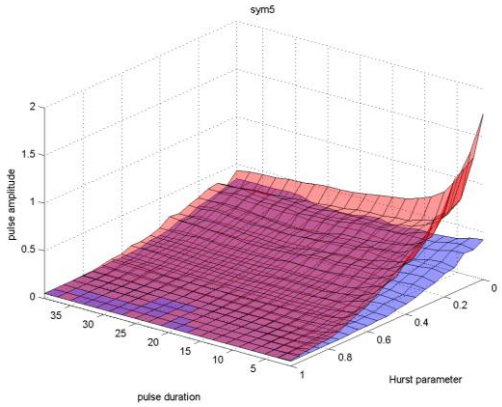


Figura 59. Performanţe la metoda cu denoising parţial, wavelet Symlet-5, 7 nivele, prag 10-sigma

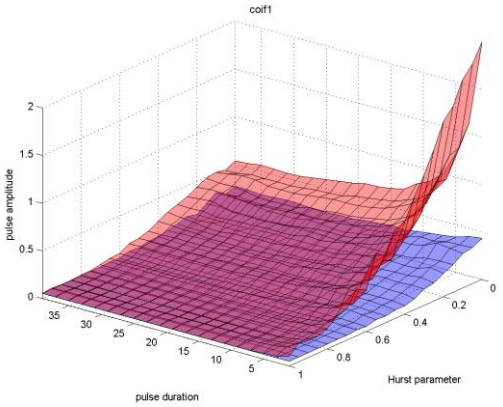


Figura 60. Performanţe la metoda cu denoising parţial, wavelet Coiflet-1, 7 nivele, prag 10-sigma

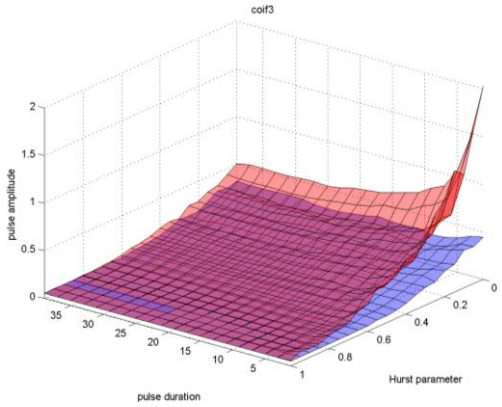


Figura 61. Performanţe la metoda cu denoising parţial, wavelet Coiflet-3, 7 nivele, prag 10-sigma

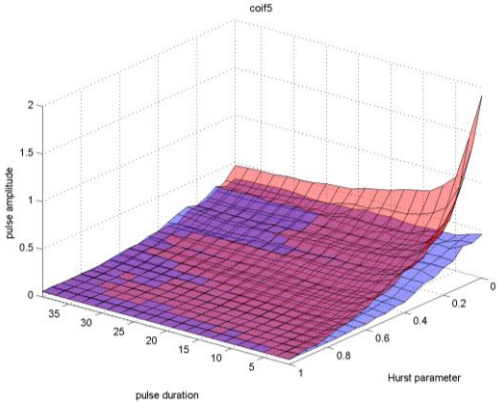


Figura 62. Performanţe la metoda cu denoising parţial, wavelet Coiflet-5, 7 nivele, prag 10-sigma

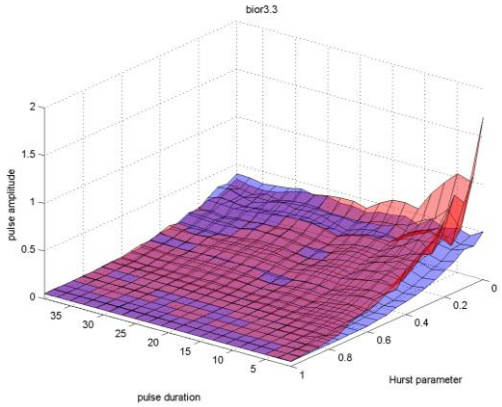


Figura 63. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-3.3, 7 nivele, prag 10-sigma

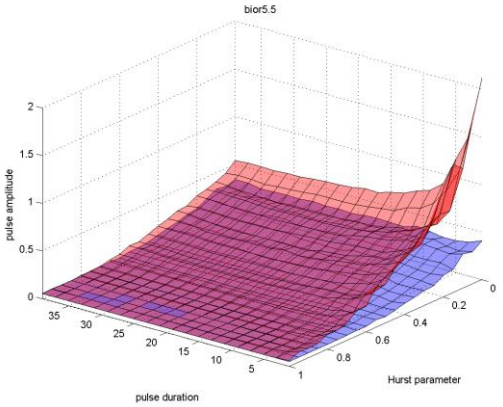
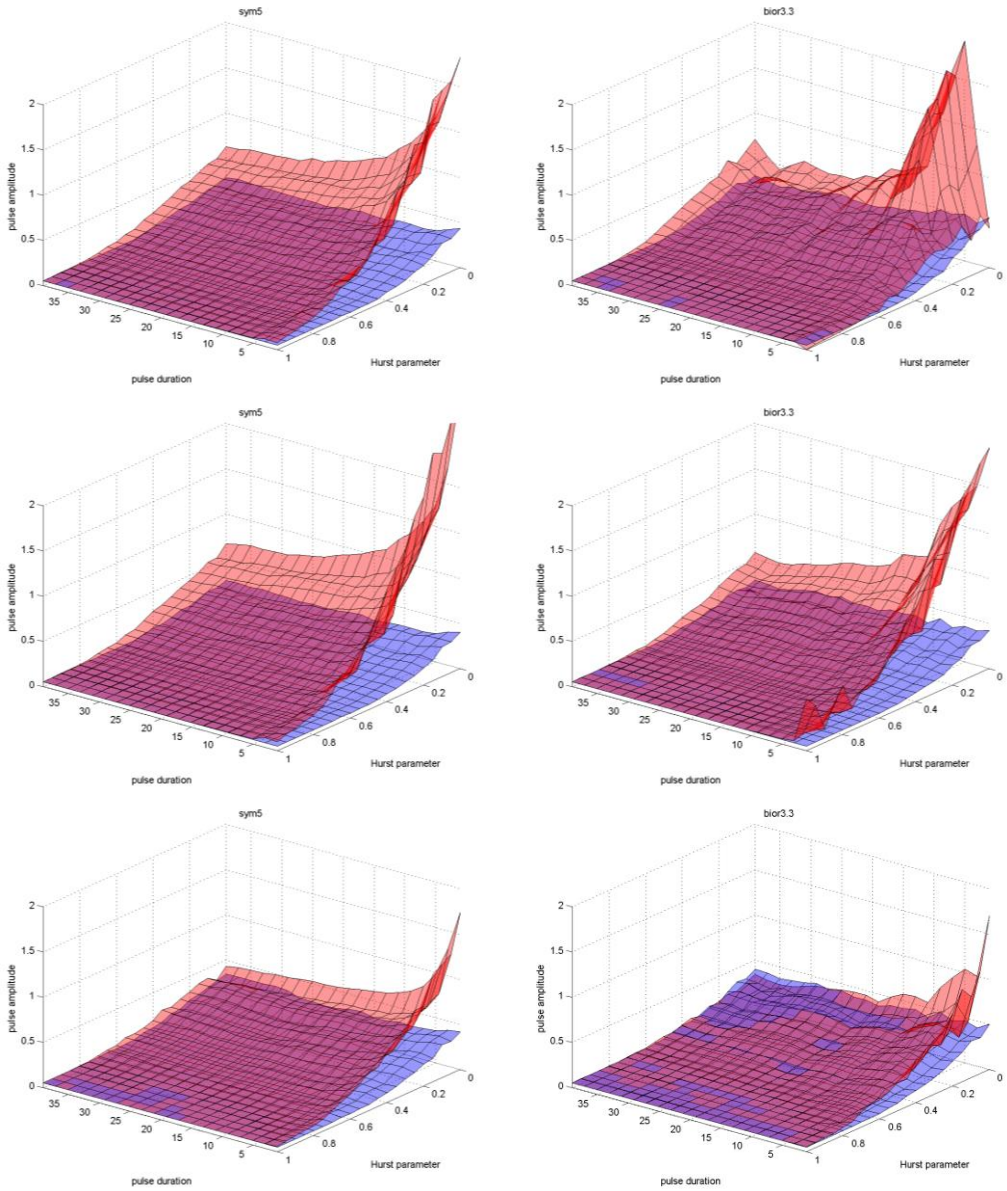


Figura 64. Performanţe la metoda cu denoising parţial, wavelet Biortogonal-5.5, 7 nivele, prag 10-sigma



Symlet-5 Biortogonal-3.3

Figura 65. Comparaţie între simulări pentru Symlet-5 şi Biortogonal-3.3. Sus – prag şi cumulare; mijloc – denoising parţial şi prag 6-sigma; jos – denoising parţial şi prag 10-sigma

Pentru o comparație mai ușoară între cele trei variante pentru care am rulat simulari, în Figura 65 prezentăm din nou diagramele pentru funcțiile Symlet-5 și Biortogonal-3.3 în toate cele trei serii de simulare. Se observă performanţa

semnificativ mai bună la algoritmul cu denoising parţial, dacă aplicăm un prag sever de detecţie.

Am colectat de asemenea informație despre timpii de execuție. Deoarece execuția experimentală s-a derulat în paralel pe un set de procesoare neomogene (4 cores i5-750 și 2 cores CentrinoDuo) am ales să comparăm relativ timpii de execuție pentru un ciclu Monte-Carlo pe un singur core din setul mai rapid. Chiar dacă nu constituie o măsură absolută (timpul include și durata de transfer prin rețea a setului de date care in cazul de față poate atinge valori de ordinul 1-2 secunde) putem să folosim acești timpi ca o măsură a eficienței relative a diferitelor funcții wavelet cu privire la consumul de resurse CPU.

Wavelet	Timp execuție metoda cumulat (sec/iter)	Timp execuție metoda denoise (sec/iter)
Meyer (discrete)	21	21
Daubechies 1	10	10
Daubechies 2	10	10
Daubechies 3	12	10.5
Daubechies 4	10.5	9.1
Daubechies 5	13	10
Symlet 1	10	8
Symlet 3	12	10
Symlet 5	12	10
Coiflet 1	12.5	10.5
Coiflet 3	11	11
Coiflet 5	12	12
Biortogonal 3.3	10	9.1
Biortogonal 5.5	10.5	11

Tabel 3. Eficienţa relativă a unor funcţii wavelet, ca timpi de execuţie

Se observă că toate funcțiile wavelet au performanțe sensibil egale (în limita de aproximativ 20%, comparabilă cu eroarea indusă de metoda de măsurare), cu excepția funcției Meyer discret. Explicația este că Meyer discret are un domeniu suport semnificativ de mare comparativ cu celelalte, ceea ce conduce la un număr mai mare de calcule.

Cele două metode sunt de asemenea comparabile ca performanță pentru că efortul depus la procesul de descompunere este același la ambele metode și reconstrucția de la metoda denoise este comparabilă cu mecanismul de cumulare a deciziei. Metoda cu denoising parțial prezintă totuși aparent un oarecare avantaj. Presupunem că acesta este dat de faptul că la reconstrucție nu mai participă decât un numar mult limitat de valori nenule ceea ce permite unele optimizări în calcul.

In condițiile descrise, concluzia este că metoda recomandabilă este denoising parțial cu prag sever (10-sigma) și o funcție Biortogonal sau chiar o funcție Symlet de ordin superior.

5. CONSIDERAȚII PRACTICE. SISTEMUL NEAR

Metodele şi tehnicile de detecţie a anomaliilor de trafic prezentate în această teză au fost dezvoltate parţial din considerente practice, în urma experienţei de administrare a unor reţele de dimensiuni moderate. Din acest motiv, metodele şi tehnicile menţionate au fost aplicate pe cazul particular al acestor reţele pe care va fi necesar să le descriem pentru referinţă.

Rezultatele obţinute au valoare şi în cazul general, nu doar pentru rețelele pe care le-am avut la dispoziție. Din această cauză trebuie să ne raportăm şi la experienţa anterioară din domeniu şi pentru acest scop este nevoie de seturi de date publice pe care în mod ideal s-au derulat anterior eforturi de analiză în scop de detecție a intruziunilor.

Majoritatea seturilor de capturi disponibile public sunt capturi de scurtă durată, care evidențiază un anumit tip de atac[69][70]. Astfel de capturi sunt potrivite pentru sisteme IDS care folosesc inspecția pachetelor dar nu sunt utile pentru testarea unor sisteme care folosesc analiza traficului. Alte seturi de date publice sunt rezultate gata agregate în forma unor colecții de fluxuri, cu relevanță mai redusă pentru metodele prezentate aici deoarece o parte a metodei descrise face exact o discriminare de direcție multi-nivel care nu mai poate fi reconstituită din capturi de fluxuri în formatele curente (de exemplu seturile de date bazate pe fluxuri nu au informaţie despre ARP).

Ca set de date de referință am ales seturile de date produse de MIT Lincoln Laboratory cu sprijinul DARPA (Defense Advanced Research Projects Agency) şi al AFRL (Air Force Research Laboratory). Aceste seturi de date sunt parte a unui efort de evaluare sistematică şi coerentă a sistemelor de detecţie a intruziunilor (IDS – Intrusion Detection System) care s-a derulat în 1998 [71] şi 1999 [72]. Aceste seturi de date sunt cunoscute informal în domeniu drept DARPA'98 respectiv DARPA'99 şi în continuare ne vom referi la ele în acest mod.

Seturile de date DARPA au câteva dezavantaje evidente, între care primul este vârsta (reflectată în primul rând în atacurile prezentate și în tehnologia de captură). Am folosit totuși aceste seturi de date în analiză pentru că este dificil de obținut alte seturi de date de referință suficient de realiste și de dimensiuni rezonabile. Dificultatea de a obține capturi publice de volum și de calitate necesară unei analize de trafic este dată pe de o parte de dimensiunea datelor implicate (mai ales dacă luăm în considerare viteza la care funcționează interfețele de rețea uzuale și volumele de trafic existente) și pe de altă parte de aspectele de securitate și confidențialitate care apar la trecerea unor capturi de pachete din domeniul privat în spațiul public. Dacă dificultățile legate de volum ar putea fi depășite prin tehnici de decimare sau prin analiza unor segmente de rețea cu trafic rezonabil de variat dar relativ redus, anonimizarea capturilor reale este dificil de făcut corect și poate să mascheze exact caracteristicile de trafic pe care le căutăm.

Trebuie să observăm că există și alte seturi de date publice în format brut PCAP care au volumul necesar pentru a putea aplica metodele descrise de analiză a traficului dar acestea sunt rezultate strict din sinteză şi nu din captură de trafic real. Metodele folosite la sinteza acestor seturi de date pot să afecteze statistica distribuției traficului așa încât să nu reflecte o situație realistă în fața metodelor de analiză folosite aici. Nici seturile de date DARPA nu sunt complet obținute din trafic real ci sunt rezultatul unor simulări dar au reușit să câștige o poziție de referință pe care nu putem să o ignorăm. Părerile cu privire la relevanța seturilor de date DARPA

în testarea sistemelor IDS curente sunt împărțite dar există o înclinare spre poziția „setul DARPA este un prag minim pe care orice sistem IDS modern trebuie să îl treacă" [73][74].

In aceste condiții, mecanismele de detecție descrise în această lucrare au fost dezvoltate și testate pe capturi reale din rețele reale dar care din motive de securitate nu pot fi făcute publice și respectiv verificate pe seturile de date DARPA, pentru care am evidențiat posibilele deviații de la situațiile reale acolo unde este cazul. Această abordare este aparent uzuală în domeniu [75], chiar dacă există și propuneri de metodologii pentru construcția a unor seturi noi de date care să înlocuiască seturile DARPA [76].

5.1. Evaluarea IDS DARPA'98

Primul exerciţiu de evaluare a IDS organizat de MIT Lincoln Laboratory s-a derulat în 1998 şi a fost gândit pentru a aprecia sistemele IDS existente la acel moment, cu intenția adițională de a obține experiență suplimentară care să permită realizarea unor sisteme de calitate superioară [71][77]. Evaluarea s-a derulat în două sesiuni. Prima a fost deschisă tuturor entităților interesate și a fost o evaluare offline bazată pe seturi de date generate special în cadrul proiectului DARPA'98. A doua sesiune s-a derulat pe o rețea reală într-un mediu controlat organizată de AFRL. A doua sesiune a fost cu acces limitat, destinată sistemelor care au reușit să demonstreze în prima sesiune că sunt pregătite pentru un test realist. Din cauză că datele primei sesiuni au fost făcute publice și pot fi folosite offline, multe lucrări științifice din domeniu s-au raportat la aceste seturi de date. Derularea directă pe rețea reală a celei de a doua sesiuni a făcut ca aceasta să fie de circulație restrânsă. Pentru scopurile noastre, doar setul de date public al primei sesiuni DARPA'98 prezintă interes.

Derularea primei sesiuni DARPA'98 a presupus generarea a trei seturi de date:

- date de antrenament conținând secțiuni fără atacuri și secțiuni cu atacuri
- date de test nominal desemnate pentru validarea funcționării IDS
- date de test pentru evaluarea eficienței IDS

In fapt, datele de antrenament și datele de test nominal fac parte din aceeași categorie. Diferența este că datele de test nominal sunt desemnate special (pentru DARPA'98 a fost vorba de ultima săptămână de date de antrenament) pentru ca să existe un termen comun de validare inițială pentru toate sistemele implicate.

Toate seturile de date conțin aceleași categorii de informații:

- capturi complete tcpdump pentru traficul de rețea
- sumar de sesiuni de rețea coerent cu traficul tcpdump care conține inclusiv informație despre prezența sau absența unui atac in respectiva sesiune (pentru primele doua seturi de date)
- date de audit din sistemul de operare Solaris al mașinilor implicate (BSM – Basic Security Module)
- sumar de atacuri pentru datele de audit BSM, similar cu sumarul pentru tcpdump
- date despre procesele aflate in execuție in fiecare minut pe mașinile implicate

- arhive de salvare săptămânale și zilnice a fișierelor de pe mașinile implicate
- diagrama rețelei în care s-a derulat simularea

Pentru scopurile acestei lucrări, datele relevante sunt cele de trafic colectate cu tcpdump. Fișierele sumar sunt declarate chiar de organizatorii evaluării ca posibil incomplete. Informația referitoare la BSM nu este ușor de generalizat pentru alte sisteme iar informația de procese este specifică unor atacuri care nu constituie obiectul tehnicilor pe care le descriem în continuare.

Pentru a putea trata corect datele din capturile tcpdump trebuie să luam însă în considerare diagrama rețelei pe care s-a facut generarea seturilor de date, aşa cum apare ea în setul public de date DARPA'98 [Figura 66].

Se observă din diagramă ca reţeaua folosita la generarea setului de date are două segmente Ethernet, marcate în figură cu „Inside" şi „Outside". Segmentele sunt implementate folosind hub-uri (tehnologie uzuală pentru perioada respectivă), ceea ce implică faptul că un nod oarecare are acces la tot traficul de reţea al segmentului. Cele două segmente sunt separate de un router care din documentaţia disponibilă rezultă că nu are funcţionalitate de filtrare (permite tot traficul fără restricţii). Trebuie să observăm că denumirile „Inside" şi „Outside" sunt cele originale introduse de autorii setului de date și nu au legătură directă cu domeniile de autoritate pe care le-am descris anterior.

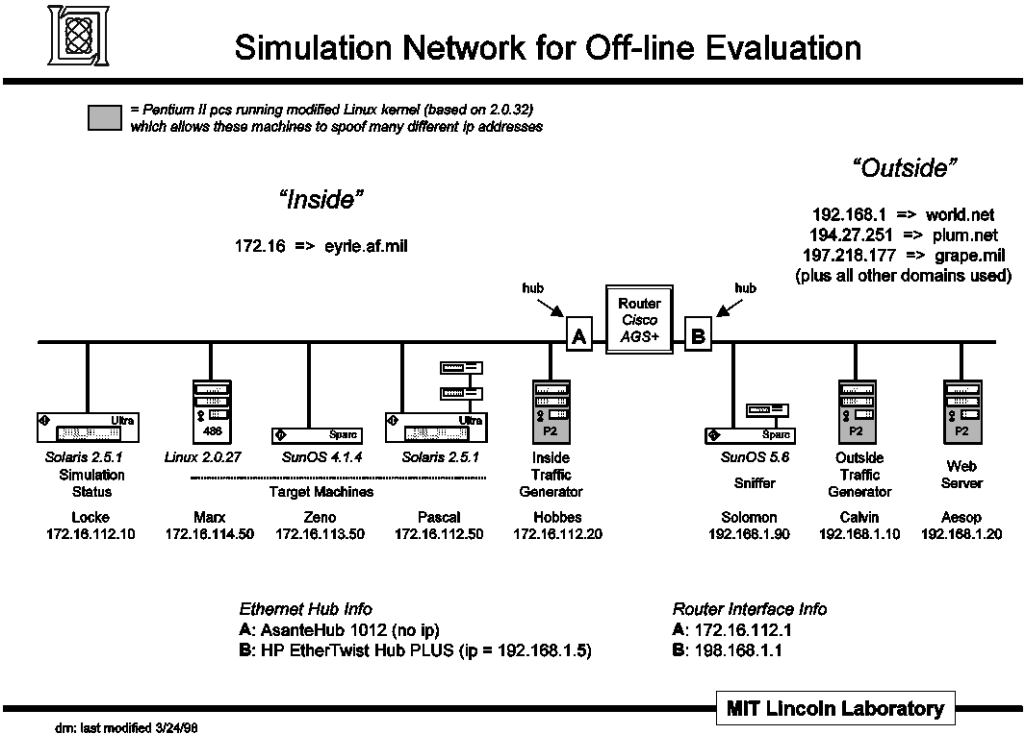


Figura 66 Diagrama reţelei DARPA'98 (conform setului public de date)

Singurul punct în care se face captură de trafic este pe segmentul implementat cu hub-ul B. Captura este făcută pe o singură interfață a unei stații dedicate în acest scop (Solomon, IP 192.168.1.90) dar datorită faptului că segmentul Ethernet este implementat cu un hub captura este echivalentă cu o captură făcută direct pe interfaţa router-ului care are adresa IP 192.168.1.1. Din punct de vedere al topologiei, rețeaua internă are o structură simplă, cu un singur segment, dar acest lucru nu este vizibil pentru noi deoarece folosim captura care este oricum executată pe interfața externă a router-ului (deci nu putem vedea de fapt detaliile de structură a rețelei interne). Detaliile de adresă menționate mai sus nu sunt de fapt importante pentru procesul de extracție și analiză pe care l-am aplicat dar au semnificație pentru configurare deoarece folosim ca sursă de date fișiere și nu capturi directe în timp real. Așa cum am explicat în capitolul referitor la principiile de captură, este nevoie să putem identifica interfața, segmentul de rețea și direcția pachetelor capturate.

Pentru a putea procesa corect capturile offline avem așadar nevoie de adresa MAC a interfeței router-ului. O analiză sumară a capturilor ne arată că adresa MAC a interfeței router-ului cu adresa 192.168.1.1 este 00:00:0C:04:41:BC. Mecanismele de analiză au nevoie în principiu și de masca de rutare pentru interfața pe care se face captura dar în acest caz rețeaua unde se face captura este oricum o rețea considerată „externă” așa încât masca nu are o importanță prea mare. Aparent din capturi rezultă că masca folosită a fost 255.255.0.0 deci această mască am folosit-o și la extracție.

Traficul care face parte din seturile de date DARPA’98 nu este în întregime trafic real. In fapt, în lucrarea originală care descrie modul de organizare a experimentului [71] autorii declară că la momentul pregătirii experimentului au luat în considerare trei posibilități:

- captură complet realistă a unei situații reale de trafic în timp ce se execută atacuri reale
- captură de trafic complet real urmată de anonimizarea informațiilor transportate și urmată de injecția de atacuri artificiale
- captură pe o rețea privată în care se execută sinteză de trafic bazată pe parametri statistici constatați din trafic real combinată cu atacuri sintetizate și componente de trafic real non-confidențial.

Primele două variante au fost eliminate ca fiind nerezonabile și setul final de date a fost generat folosind ultima metodă. Nu există detalii despre metodologia exactă folosită pentru a genera trafic sintetic pe baza caracteristicilor constatate din trafic real [73] dar nu putem decât să ne aliniem la consensul aproximativ că această simulare este rezonabil de realistă.

5.2. Evaluarea IDS DARPA’99

In urma interesului pe care l-a ridicat evaluarea DARPA’98 autorii au luat măsuri de ajustare, îmbunătățire și extindere a experimentului [72]. Pentru această a doua ediție s-au adăugat stații Windows la rețeaua internă și externă și s-a extins setul de date cu capturi și pentru segmentul intern. La modelul atacurilor s-au adăugat atacuri originate în spațiul intern, orientate către spațiul intern. In plus, de această data nu toate atacurile din segmentul de test au fost prezente și în segmentul de training. In acest fel s-a încercat evaluarea abilității sistemelor de a detecta atacuri noi. Aspectul statistic al atacurilor a fost de asemenea ajustat astfel

încât acest set conține și atacuri care încearcă să simuleze trafic normal pentru a evita detecția.

Structura rețelei folosite pentru DARPA'99 este reprezentată în Figura 67. Diagrama este diagrama originală oferită de autori [78] (în această diagramă rețeaua internă este plasată în partea dreaptă, spre deosebire de diagrama DARPA'98). Se observă noua stație de monitorizare (locke, IP 172.16.112.10) și mașinile Windows adăugate. Și în acest caz segmentele Ethernet au fost implementate cu hub-uri deci captura poate fi considerată a fi făcută pe interfețele router-ului.

Aceleași observații pe care le-am făcut la setul DARPA'99 cu privire la importanța adreselor se aplică și aici. In acest caz avem două puncte de captură. Pentru punctul de captură extern echivalent (pe interfața router-ului spre hub B) adresa IP este în continuare 192.168.1.1 cu masca 255.255.0.0 iar adresa MAC a interfeței este de asemenea neschimbată și anume 00:00:0C:04:41:BC.

Pentru punctul de captură intern echivalent (interfața router-ului spre hub A) adresa IP folosită este 172.16.0.1 cu masca 255.255.0.0 iar adresa MAC este 00:10:7B:38:46:33. La prima vedere poate să pară că determinarea adresei MAC pentru cele doua interfețe conține o eroare, deoarece nu aparțin din aceeași familie de adrese, deși sunt interfețe ale aceluiași router. In fapt, atât prefixul 00:00:0C cât și prefixul 00:10:7B sunt alocate pentru Cisco Systems. Singurul punct interesant dar strict de domeniul curiozitate rămâne întrebarea „de ce interfețele sunt din prefixe diferite, daca router-ul este un Cisco 2514 care are exact doua interfete AUI built-in?".

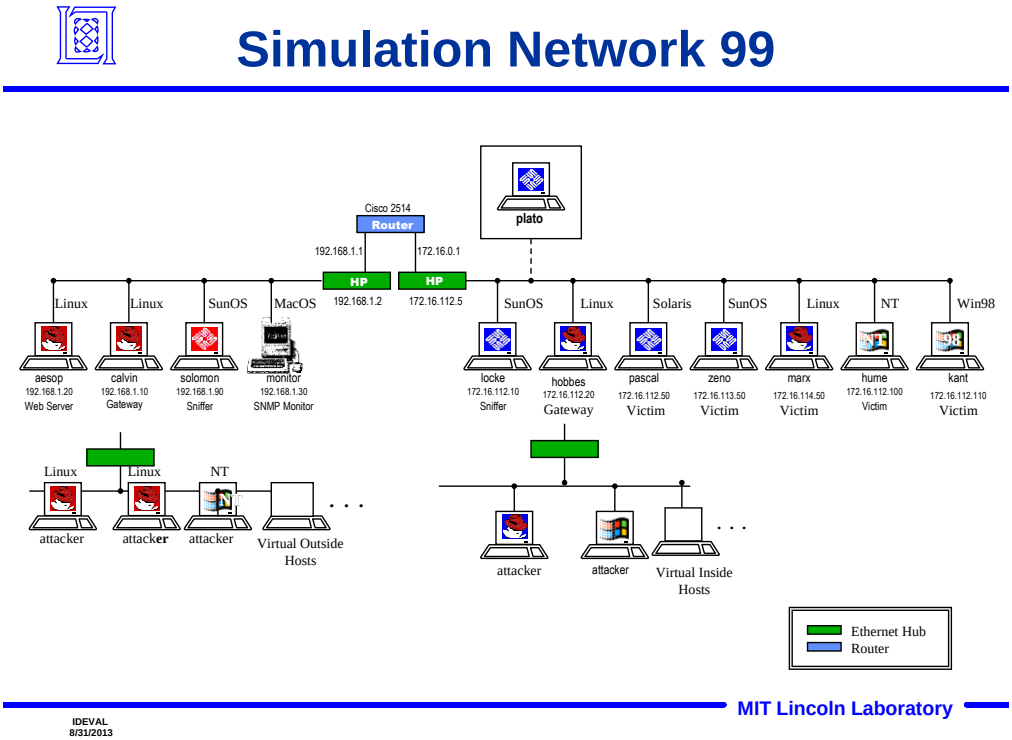


Figura 67. Diagrama rețelei DARPA'99 (conform setului public de date)

Detaliile referitoare la construcția DARPA'99 au fost mult mai bine documentate decât pentru DARPA'98 atât din punct de vedere al sintezei de trafic cât și al atacurilor incluse, într-un raport final [79]. De această dată este evident și documentat că s-au depus eforturi semnificative pentru a simula trafic realist. Chiar dacă nu există substitut perfect pentru captura de trafic real, setul DARPA'99 poate fi considerat o aproximare rezonabilă, cel puțin pentru specificul de trafic al perioadei respective.

Experiența obținută de echipa de la MIT Lincoln Laboratory cu ocazia implementării celor două evaluări DARPA a fost dusă mai departe în sistemul de sinteză LARIAT [81] și în seturi suplimentare de date de sinteză [80] dar acestea nu au mai avut un impact la fel de mare ca și cele două evaluări inițiale.

5.3. Rețele proprii ca sursă de date pentru analiză

Pentru a compensa deficiențele menționate ale seturilor de date DARPA'98 și DARPA'99 am folosit și capturi efectuate pe rețele reale la care am avut acces. Din motive evidente pe care l-am descris deja mai sus, traficul capturat pe aceste rețele nu poate fi făcut public. In aceste condiții, nici măcar nu am mai trecut prin faza de fișiere care conțin trafic ci am executat captura și extracția în timp real, folosind unealta near-agent care va fi descrisă într-o secțiune următoare. Deoarece near-agent a funcționat în acest caz în regim captură-extracție, adresele au fost determinate direct prin interogarea sistemului de operare așadar nu sunt relevante pentru această prezentare și deoarece pot fi considerate informații confidențiale nu vor fi menționate aici.

Pentru a înțelege semnificația seriilor de date obținute este însă necesar să descriem contextul general și punctele în care s-a făcut captura și extracția respectiv funcționalitatea nodurilor de rețea participante.

Prima rețea pe care s-a dezvoltat și testat sistemul este o rețea privată de cercetare și dezvoltare software pe care o vom numi în continuare rețeaua RD, care are structura de principiu simplificată prezentată în Figura 68. Rețeaua principală este un domeniu Ethernet realizat cu mai multe switch-uri Fast Ethernet conectate într-o structură ierarhizată pe două nivele. Unul din switch-uri este un Cisco 2960 folosit la diverse experiențe legate de rețea și care este producător de trafic STP şi CDP (deci din punctul nostru de vedere cadre periodice 802.3-v1, cu payload non-IP). Chiar dacă topologia rețelei nu necesită protocolul STP, prezența lui este conformă cu cazul unor rețele mai mari și este relevant pentru mecanismele de analiză dezvoltate. Trebuie menționat că rețeaua reală conține și trafic VLAN (802.1q) și alte sisteme care nu sunt descrise aici explicit (W-DVn, S-n). In măsura în care anumite caracteristici colectate sunt explicate de aceste elemente omise aici, ne vom referi la ele dar numai acolo unde este nevoie.

Rețeaua principală este conectată cu rețeaua furnizorului de acces Internet printr-un gateway Linux (GW-I) care funcționează ca router direct și cu translație de adrese (NAT), proxy http, server de mail, server DNS. Pe aceeași mașină funcționează sistemul principal de monitorizare al unei surse de alimentare neinteruptibilă (UPS) bazat pe pachetul open-source apcupsd [82]. Deoarece mai multe servere din configurație sunt alimentate de același UPS, acestea interoghează periodic GW-I pentru a monitoriza starea sursei de alimentare astfel încât să poată reacționa (shutdown corect) înainte de epuizarea bateriei. Server-ul mai produce și alte tipuri de trafic (de exemplu trafic de tip syslog spre un colector intern sau trafic de sincronizare DNS cu alti parteneri interni).

Rețeaua principală este de asemenea conectată cu alte rețele private prin intermediul unui al doilea gateway Linux (GW-P). Acesta are rol de router NAT, server DNS. In mod normal traficul inter-rețea se desfășoară dinspre rețeaua principală spre alte rețele private dar nu invers.

In rețeaua principală sunt conectate mai multe servere și stații de lucru care rulează sisteme de operare Linux și diferite variante de Windows. In particular, interesant pentru scopurile analizei noastre este vorba de un server pentru bază de date (S-DB), un server de integrare continuă (S-CB) și o stație de dezvoltare (W-DV).

Server-ul bază de date (S-DB) rulează o instanță de bază de date Oracle folosită pentru teste și funcționează de asemenea ca server de fișiere, server CVS și server http pentru module software finite. Accesul la el pentru administrare se face prin SSH.

Server-ul de integrare continuă urmărește permanent modificările apărute în fișierele disponibile în server-ul CVS respectiv în module software finite. Dacă detectează modificări execută automat procesul de build care construiește și testează modulele software aflate în dezvoltare. Procesul este automat și are ca rezultat finit modulele software finite care se depun pe S-DB. De-a lungul ciclului de funcționare S-CB generează diverse tipuri de trafic, în principal către S-DB dar și către alte servere din domeniu.

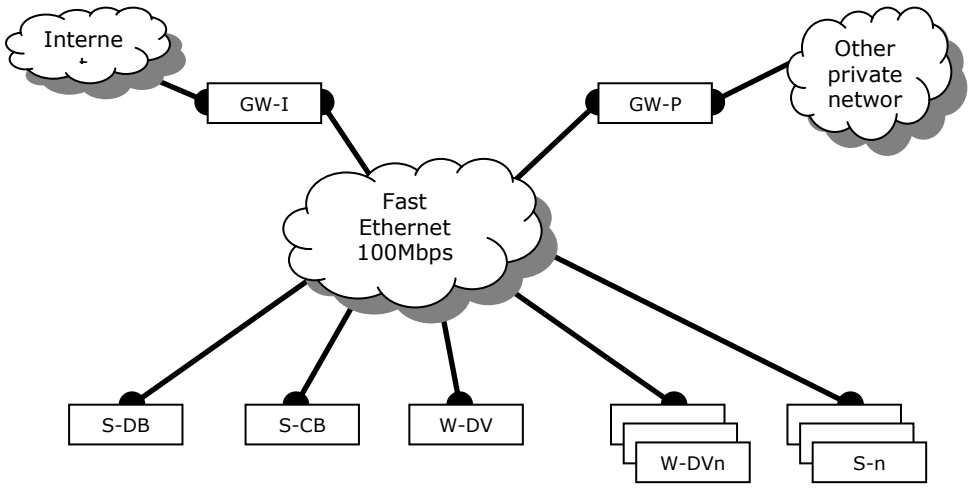


Figura 68. Diagrama reţelei private RD

In faza de monitorizare, S-CB accesează S-DB folosind FTP, HTTP și protocolul pserver specific CVS pentru a detecta apariția unor modificări. Aceste accese au o puternică prezentare periodică, dar compusă din mai multe „semnale” cu perioade diferite pentru că module diferite sunt interogate cu perioade diferite. Odata inițiată faza de compilare/asamblare/testare modelul accceselor se schimbă și predomină accese HTTP de volum relativ mai mare către S-DB și alte servere, urmate de accese la baza de date S-DB. Această fază este însoțită și de o creștere semnificativă a gradului de utilizare a procesorului. La final se fac accese de tip FTP pentru depunerea modulelor finite pe server-ul S-DB.

Mașina S-CB are pe lângă rolul principal de integrare continuă și alte funcții secundare. Pe aceeași mașină rulează un colector periodic de statistici de trafic pentru întreaga rețea și alte informații asociate (temperatură, încărcare CPU) bazat pe MRTG [83]. Acest colector produce trafic SNMP la intervale de 5 minute, către diferite noduri din rețea. Datele produse de MRTG și datele rezultate din bucla de build continuu sunt consultate ocazional de clienți din rețea via HTTP.

Stația de dezvoltare W-DV este de fapt o mașină virtuală Linux pe care am derulat dezvoltarea modulului colector si extractor de trafic near-agent. Interacțiunea cu această mașină este bazată pe SSH (care funcționează și ca tunel pentru trafic de X-window server). Nu există alte servicii care rulează pe această mașină și din acest motiv capturile efectuate aici sunt relevante pentru traficul de fond al rețelei și pentru specificul unei mașini client.

Pe rețeaua RD nu am derulat atacuri reale deoarece este o rețea a cărei integritate este critică. Capturile efectuate însă pe această rețea sunt interesante însă pentru categoriile de anomalii bazate pe periodicitatea normală a unor elemente de trafic.

A doua rețea pe care am folosit-o pentru a obține încă un set cu serii de date ca să existe o varietate mai mare de condiții este o rețea folosită în învățământ la conectarea unor laboratoare cu rețeaua de campus și finalmente cu Internet-ul. Această rețea o vom numi în continuare rețeaua EDU și este prezentată în Figura 69.

Rețeaua principală pentru scopurile noastre este rețeaua marcată în figură ca rețea de laborator și deservește mai multe săli de laborator (în total peste 50 noduri). Această rețea este conectată la rețeaua de campus printr-un gateway GW-D. Rețeaua de campus conține diverse stații de lucru și alte noduri de rețea (W-Xn) și este conectată la Internet printr-un router R-I. Prin rețeaua de campus de poate accesa server-ul de mail intern S-M.

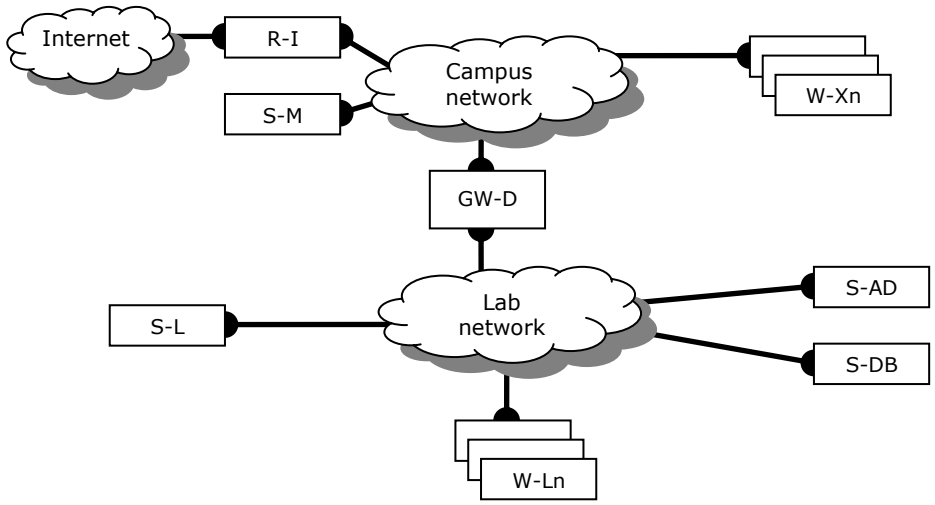


Figura 69. Diagrama reţelei private EDU

Din punct de vedere fizic rețeaua de laborator este o rețea Ethernet organizată pe doua nivele ierarhice. Primul nivel ierarhic folosește switch-uri Gigabit Ethernet la care sunt conectate serverele S-xx, al doilea nivel conține switch-uri Fast Ethernet la care sunt conectate stațiile de lucru W-Ln. Switch-urile au capabilități limitate de management dar suportă STP. Cu toate acestea, deoarece topologia rețelei nu necesită STP, protocolul nu a fost activat.

Rețeaua conține un server de domeniu Windows Active Directory S-AD și un server pentru baze de date S-DB care rulează de asemenea Windows. Server-ul S-AD îndeplinește toate funcțiile tipice unui astfel de server, între altele funcția de server DNS pentru domeniul local, server DHCP pentru rețeaua locală, server SMB/CIFS, etc. Există de asemenea un server Linux (S-L) care funcționează ca server HTTP și gazdă pentru mașini virtuale.

Captura de trafic a fost făcută pe interfețele internă și externă a GW-D. Acesta serveste ca router cu NAT pentru traficul cu originea în rețeaua laboratorului. De asemenea, oferă funcționalitate de proxy-server (care însă nu este folosită), server DNS (cache pentru AD și primar pentru un segment de adrese al laboratorului de rețele), server de time (NTP), server de licențiere.

5.4. Sistemul NEAR – platformă de captură și analiză a anomaliilor din rețea

Pentru a putea verifica și perfecționa ideile descrise în capitolole anterioare am implementat câteva instrumente care să permită captura de trafic, extracția în timp real a seriilor de date, prezentarea lor în formă grafică pentru o explorare manuală mai ușoară și analiza offline a acestora. Aceste instrumente sunt parte dintr-o platformă de captură și analiză care urmează să fie dezvoltată pe viitor sub forma unui sistem complet semiautomat de detecție a anomaliilor (NEAR).

La nivel principial, un astfel de sistem ar avea structura din Figura 70

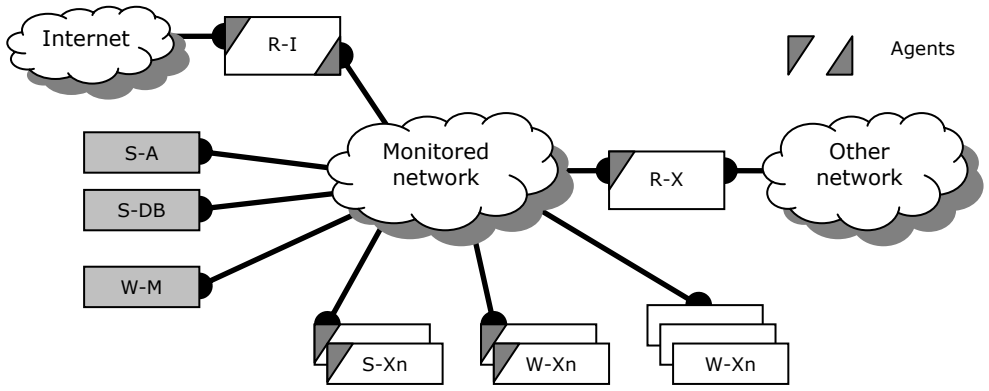


Figura 70. Sistem de detecție a intruziunilor NEAR

Sistemul de monitorizare este compus din:

- agenți care capturează trafic de rețea și extrag serii de date,
- agenți care colectează alte mărimi și extrag serii de date,
- un sistem server de analiză (S-A),

- un sistem server bază de date pentru stocarea istoricului de evenimente și a elementelor de configurare (S-DB)
- stații și instrumente de consultare/monitorizare/întreținere (W-M).

In figură sunt reprezentate de asemenea elemente care fac parte din rețeaua pe care o monitorizăm:

- routere care conectează rețeaua cu Internet-ul și cu alte rețele private; aceste routere au o parte din interfețe monitorizate de agenți de captură
- servere (S-Xn) care pot fi toate puncte de interes dotate cu agenți de captură și colectare date
- stații de lucru (W-Xn) din care o parte pot fi monitorizate pentru îmbunătățirea rezoluției sistemului

Agenții de captură rețea și agenții care colectează alte date pot fi module software direct integrate în nodurile pe care le monitorizează sau pot fi implementați prin intermediul unor funcționalități de monitorizare existente în hardware-ul și firmware-ul nodurilor interesante, accesate prin intermediul unor protocoale standard. Abordarea care folosește funcționalități existente deja în hardware/firmware este mai simplu de implementat (este de fapt modelul folosit de multe sisteme de management via SNMP). Această abordare are însă dezavantajul că trebuie să ne limităm la setul de informații existent și nu putem să experimentăm noi direcții de extracție și analiză. Din acest motiv, acolo unde e posibil, considerăm că metoda cu modul software e preferabilă în cazul rețelelor de dimensiune moderată și administrare compactă.

Sistemul de analiză are rolul de a agrega seriile de date de la agenți și de a executa operațiile de analiză. Acestea operații sunt mai mari consumatoare de resurse decât simpla captură și extracție, deci nu puteau fi integrate direct în agenți. In plus, o parte din eficiența detecției rezultă din corelațiile care se pot face între informații provenite din diferite puncte ale rețelei, cee ce ne conduce spre un server specializat de analiză. Acest server mai poate fi folosit și pentru a oferi informații sintetice despre starea rețelei și pentru a servi ca punct de plecare pentru notificări în timp real în cazul detecției unor anomalii grave.

Datele rezultate in urma analizei pot fi stocate pemtru consultare informativă ulterioară sau chiar pentru analiză post-factum cu algoritmi mai detaliați sau chiar manual, în vederea perfecționării continue a capabilităților de detecție. Un astfel de server de date poate să funcționeze în cazul unei rețele de dimensiuni moderate pe același nod cu server-ul de analiză deci nu necesită resurse suplimentare.

Interacțiunea utilizatorilor cu sistemul de detecție a anomaliilor se face prin stații de lucru care folosesc interfața de tip browser îndreptat spre interfața HTTP a server-ului de analiză sau prin unelte dedicate care accesează direct agenții. Aceste din urmă instrumente pot servi la depanarea și diagnosticarea însăși a sistemului sau la activitate de cercetare pentru îmbunătățirea acestuia.

5.5. NEAR-agent - instrument de colectare trafic și extracție serii de date

Conceptul pe care l-am avut în vedere pentru un astfel de sistem de monitorizare este parțial intruziv, bazat pe ideea de a instala agenți de captură și extracție pe diferite noduri din rețea (server, stații de lucru, routere) în măsura în care sistemul de operare al respectivelor noduri permite. Presupunem că aceste module agent au un consum redus de resurse și efortul de instalare a lor este

rezonabil comparativ cu avantajele aduse. Trebuie observat că nu este necesară instalarea agenților pe toate calculatoarele din rețea ci este suficientă instalarea pe un număr limitat dar bine ales de noduri interesante.

Pentru cazurile în care sistemul de operare al unui nod interesant nu permite instalarea unui astfel de modul agent dezvoltat special se pot găsi metode alternative. De exemplu putem să facem analiza doar pe informații rezultate din agregatoare de fluxuri sau să instalăm elemente de captură explicită pe interfețele interesante (de exemplu porturi de switch în regim mirroring sau alte dispozitive specializate de captura pasiva de trafic).

In procesul de dezvoltare ale acestei lucrări am ales să implementăm un agent de captură care funcționează pe platformă Linux folosind biblioteca de largă răspândire libpcap. Datorită faptului că această bibliotecă este disponibilă și pe alte platforme (în mod notabil pe Windows) portarea codului pentru agentul de captură și extracție ar trebui să fie ușor de făcut și extinderea sistemului sa fie suficient de simplă.

Obiectivele principale pe care l-am stabilit la dezvoltarea acestui modul-agent sunt:

- trebuie să poată efectua captură de pachete în timp real
- trebuie să analizeze structura pachetului (de fapt a cadrului de date) și să îl clasifice conform principiilor descrise în capitolele anterioare
- trebuie să întrețină contoare de pachete pe categorii de clasificare și interval de timp. Granularitatea intervalului de timp trebuie să fie suficient de mică pentru a permite distingerea unor evenimente de scurtă durată
- trebuie să rețină toate valorile contoarelor o perioadă suficient de mare de timp pentru a permite unui sistem centralizat să efectueze colectări mai rar, în cazul în care nu se dorește o monitorizare permanentă
- trebuie să colecteze statistici despre distribuția pe porturi a pachetelor, destinate identificării modificărilor în tipologia traficului care necesită reconfigurarea categoriilor
- trebuie să ofere acces la setul curent de date reprezentat de contoare și statistici prin intermediul unei interfețe de transfer machine-to-machine simple și comode
- trebuie să aibă un consum redus de resurse, astfel încât să poată funcționa permanent fără impact semnificativ asupra nodului pe care este instalat

Pe lângă aceste obiective principale care sunt esențiale pentru folosirea modulului în sistemul NEAR, din experiența de exploatare au rezultat și cerințe adiționale:

- trebuie să poată filtra doar o parte din trafic. Această opțiune este utilă în cazul procesării în timp real pentru a elimina de exemplu traficul normal dar neinteresant direcționat chiar spre interfața de transfer a agentului sau pentru a elimina o anumită componentă de trafic de volum mare cu intenția de a reduce efortul de procesare
- trebuie să poată salva seriile de date rezultate în fișiere zilnice pentru a acoperi cazul de exploatare fără server principal de analiză
- trebuie să poată funcționa și asupra unei surse de pachete offline (fișier de captură PCAP) pentru a putea implementa analiza seturilor de date colectate din alte surse
- trebuie să fie simplu de rulat și de configurat

Implementarea modulului a fost făcută în limbajul C++ pentru a obține o performanță bună și consum redus de resurse. Limbajul oferă și o portabilitate suficient de bună astfel încât o parte din efortul de scriere a modulului să fie recuperabilă și pentru alte sisteme de operare.

Pentru captura propriu-zisă am folosit biblioteca libpcap. Mecanismul de apelare al bibliotecii presupune următorii pași:

- deschiderea unei surse de captură care poate fi o interfață direct disponibilă în sistem sau un fișier în care a fost salvată anterior o captură (de exemplu cu o comandă de tipul `tcpdump ...-w fisier...`)
- configurarea unui eventual filtru care să fie folosit în cursul capturii
- invocarea în mod repetat a metodei `pcap_dispatch(...)` care solicită bibliotecii libpcap să extragă următorul pachet și să îl supună unei funcții de analiză

Pentru partea de inițializare a sursei de captură în cazul capturii în timp real avem nevoie de numele unei interfețe de rețea a nodului pe care rulează agentul. In lipsa acestui parametru se va folosi ca interfață de captură interfața implicită determinată via funcția `pcap_lookupdev()`. Odată determinată interfața se obține detalii despre adresare prin interogarea sistemului de operare. Avem nevoie cel puțin de adresa MAC a interfeței, de adresa IP și de mască. Pentru cazul în care avem de-a face cu o interfață la care au fost asociate multiple adrese IP am prevăzut posibilitatea memorării unei liste de elemente adresă/mască.

In cazul în care inițializarea se face cu o sursă de captură offline (fișier) informația despre adresa MAC și adresa IP (adresele IP) considerate a fi ale interfeței de captură nu sunt disponibile direct și trebuie obținute din altă sursă (vezi situația seturilor DARPA). Acest caz este acoperit de parametri obligatorii de configurare.

Analiza se face pachet-cu-pachet și este implementată cu o funcție proprie, apelată de libpcap. Analiza pachetului este făcută incremental și datele interesante sunt acumulate într-o structură intermediară, adaptată scopurilor de decizie în una din categoriile descrise în capitolele anterioare.

Structura principală a procesării care culege datele esențiale pentru discriminare în categorii este:

- se verifică sursa și destinatarul la nivel 2 ISO pe bază de adresă MAC. Incadrarea este OTH/ME/BCAST, cu semnificația alt_nod/acest_nod/broadcast
- se verifică prezența unui tag VLAN 802.1q și se marchează faptul că e trafic pe bază de VLAN. Momentan această informație nu a fost folosită în discriminarea pe categorii
- se separă în tipuri de trafic (802.3-v1/IP/ARP/ altele). Traficul IPv6 este deocamdată încadrat la „altele” datorită semnificației speciale pe care o pot lua familiile de adrese locale și care nu au fost suficient studiat la împărțirea „intern”/ „extern” descrisă anterior
- pachetele care conțin trafic IP sunt analizate mai departe. Se rafinează decizia originală OTH/ME/BCAST prin adăugarea cazului SUBNET, decis din raportarea adresei sursă respectiv destinație la adresele interfeței după aplicarea de mască. Traficul este de asemenea marcat în următoarele tipuri de trafic IP: TCP/UDP/ICMP/altele-IP și se extrag adresele de port sursă și destinație (dacă e cazul)
- pachetele care conțin trafic ARP vor avea doar decizia rafinată OTH/SUBNET/ME/BCAST

După ce au fost extrase aceste caracteristici avem o structură sintetică de forma:

```
struct t_pkt_info {
   u_char traffic_type; // type of traffic (ARP/TCP/UDP/IP....)
   u_char flags;        // special flags, bitmapped
   char src_ident;      // identification of the source
                        //    (NE_IDENT_...)
   char dst_ident;      // identification of the destination
                        //    (NE_IDENT_...)
   int src_sap_addr; // SAP address for the source (i.e. port)
   int dst_sap_addr; // SAP address for the destination
                     //    (i.e. port)
   int phylen;
   int loglen;
   int hash1;
   int hash2;
};
```

Această structură este folosită pentru discriminare în una din 64 categorii pentru a actualiza unul din cei 64 de contoare curente care vor produce cele 64 serii de date.

Din cauză că majoritatea traficului este IP-TCP sau IP-UDP criteriul de discriminare pe bază de port este semnificativ și datorită abordării „eu sunt server" respectiv „eu sunt client" vom dori o structurare oarecum regulată a setului de contoare. Sa presupunem că vom avea N sub-categorii pe baza de port, deci vom avea nevoie de minim 2N sub-categorii daca adăugăm dimensiunea „eu-client"/"eu-server". Deoarece există cazuri în care nu pot determina dacă sunt client sau server vom avea nevoie de un număr adițional de N contoare. Astfel de situații se întâlnesc dacă ambele porturi sunt binecunoscute (exemplu tipic trafic DNS server-server sau trafic pe port 137) respectiv în cazul router-elor care doar transportă traficul.

La acest moment există o decizie subtilă care trebuie să o luăm în cazul nodurilor care funcționează în regim de router și anume dacă să considerăm semantica lui „eu" ca fiind nodul în sine sau rețeaua internă pe care o izolează de restul lumii. Decizia este simplu de luat în baza informației agregate în structura t_pkt_info dar din păcate nu putem să întreținem și mai multe contoare pentru cele două variante de semantică a lui „eu". Singura soluție deocamdată pentru a păstra uniformă structura setului de contoare este configurarea diferită a agentului care montorizează un router în calitate de nod terminal față de cel care monitorizează un router în calitate de router. Varianta de agent pe care am folosit-o în practică aplică semantica care dă preferință caracterului de nod (testează valoarea ME, și nu valoarea ME|SUBNET și face o încadrare preferențial pe bază de adresă IP).

Cele 3N categorii la care asociem contoare le extindem la 4N pentru o structurare binară comodă și o aliniere care va permite eventual optimizări de calcul. Al patrulea set de N categorii îl vom folosi pentru cazurile speciale care nu se încadrează în schemele bazate pe port. Urmează să alegem o valoarea comodă pentru N. Structura de stocare a contoarelor este de dimensiune statică din motive de performanță așa încât N va fi fix pentru toate modulele implicate, atât agenți cât și alte componente de analiză. Va trebui să alegem un N astfel încât să avem

suficiente categorii bazate pe porturi și să putem oferi o rezoluție bună de analiză dar în același timp să nu avem prea multe date de stocat și transferat respectiv să nu încărcăm prea mult sistemul de analiză. Am ales N=16, adică vom avea 64 de contoare și 64 serii de date.

Cele 64 categorii sunt alocate conform tabelului Tabel 4:

Index	Destinație
0	Trafic IP, client, port neclasificat (de fapt nefolosit, vezi explicația din text)
1..15	Trafic IP, client, clasă de încadrare 1..15
16	Trafic IP, server, port neclasificat (de fapt nefolosit, vezi explicația din text)
17..31	Trafic IP, server, clasă de încadrare 1..15
32	Trafic IP, nedeterminat, port neclasificat
33..47	Trafic IP, nedeterminat, clasă de încadrare 1..15
48	Trafic IP, ambele porturi neclasificate
49	Trafic 802.3-v1 (tipic 802.2 LLC, 802.2 SNAP)
50	Trafic ARP
51	Trafic ICMP
52	Inițializări de sesiune de comunicație (de exemplu TCP-SYN)
53	Inițializări de sesiune cu port destinație neclasificat
54-62	Nefolosit
63	Trafic 802.3-v2/v3, cu protocol neidentificat (IPv6, WoL, PPPoE, ...)

Tabel 4. Categorii de trafic folosite de NEAR-agent și semnificația lor

Categoriile 0 și 16 de fapt au definiții fără sens pentru că în momentul în care știu că sunt server sau client conform definiției bazate pe porturi binecunoscute înseamnă ca portul nu mai este neclasificat. Din acest motiv seriile de date prezente pe aceste valori de index vor fi de fapt permanent zero și pozițiile ar putea de fapt să fie folosite la alte cazuri speciale.

Categoriile 1..15, 17..31, 33..47 sunt 3 seturi de 15 categorii discriminate pe bază de port pentru protocoale TCP și UDP. Putem deci să definim N-1 (adică 15) seturi de porturi cunoscute pe baza cărora să recunoaștem statutul de server în analiza unui pachet. Arondarea acestor seturi trebuie făcută în funcție de rețeaua pe care o monitorizăm. Incadrarea celor 65536 de porturi posibile în una din clase se face cu o tabelă internă configurabilă. Configurația inițială a acestei tabele este preluată din parametrii de configurare cu care este pornit programul (deci putem să adaptăm NEAR-agent la specificul rețelei pe care funcționează). Valorile conținute în această tabelă se pot modifica în timpul execuției așadar un sistem centralizat de analiză și administrare poate să îmbunătățească în timp modul de distribuție al porturilor pe clase. In Tabel 2 se prezintă un exemplu de configurare folosit la procesarea offline a capturilor DARPA. Se poate observa prin comparație cu tabelul similar prezentat în capitolul despre principii de captură că se urmăresc protocoale diferite, datorită faptului că specificul de trafic diferă între o rețea actuală și rețeaua DARPA.

Index	Porturi	Descriere
1	80, 443, 8080, 8443	Trafic de tip browser
2	22, 23	Trafic de administrare și CLI (SSH, TELNET)
3	25, 110	Trafic de mail (SMTP, POP3)
4	21, 20	Trafic FTP
5	53	Trafic DNS
6	37, 123, 161, 514	Trafic-sistem (TIME, NTP, SNMP, SYSLOG)
7	6000, 6001, 6002, 6003, 6004	Trafic X-Windows
8	6665, 6666, 6667, 6668	Trafic IRCU (Internet Relay Chat)
9	79, 113	Trafic FINGER și AUTH/IDENT

Tabel 5. Incadrarea porturilor în clase folosită pentru analiza capturilor DARPA

Categoria 48 din Tabel 4 este importantă pentru că acoperă tot traficul IP neclasificat. Valorile de trafic pe această categorie nu vor fi niciodată nenule pentru că există protocoale care nu folosesc porturi binecunoscute (de exemplu sesiunea TNS de Oracle după dialogul inițial sau diferite variante de RPC). Obiectivul nostru este însă să reducem cât mai mult valorile constatate aici prin arondarea pe cât posibil a porturilor stabile în una din cele 15 categorii disponibile. Traficul care nu este clasificat poate fi justificat dar de cele mai multe ori este trafic neașteptat care poate fi chiar o anomalie semnificativă. Este important să reducem volumul de trafic de fond în această categorie pentru a ușura detecția a astfel de anomalii.

Pentru a avea un punct de sprijin în acest sens, NEAR-agent întreține o listă cu cele mai frecvent întâlnite porturi care nu au putut fi clasificate. Deoarece există o diferență principială între porturi care sunt neidentificate pentru că sunt nesemnificative (au fost porturi de plecare pe partea de client a conexiunii) și porturi care nu au fost identificate pentru că nu le-am cunoscut (încă) va trebui să întreținem încă o listă de porturi care au apărut frecvent ca destinație a unei deschideri de sesiune (de exemplu în cazul simplu porturi care au fost destinație într-un pachet TCP-SYN).

Construcția unei astfel de liste trebuie să țină seama de faptul că nu putem pur-și-simplu număra toate aparițiile de porturi deoarece în timp putem obține o listă cu 65536-k poziții unde k este numărul de porturi care sunt deja prinse în tabela de clasificare. Soluția pe care am adoptat-o este să limităm dimensiunea listei la o valoare rezonabilă și să dotăm elementele din listă cu un marcaj de timp al ultimei actualizări. Dacă e necesar să extindem lista peste dimensiunea maximă în loc să adăugăm un nou element se refolosește poziția cu cel mai mic contor (ultima poziție din listă) dar numai dacă marcajul de timp al ultimei sale incrementări este suficient de vechi. Algoritmul poate să piardă valori sau să numere imprecis dar oricum este o sursă aproximativă de informație iar lista poate fi periodic reinițializată. In practică rezultatele au fost satisfăcătoare și nu am simțit nevoia de a implementa un algoritm mai complicat.

Valorile rezultate pe cele 64 de contoare trebuie să fie stocate per interval de timp, astfel încât să obținem serii de date potrivite pentru analiză. Din experiența

anterioară de administrare dimensiunea temporală rezonabilă a fenomenelor pe care dorim să le evidențiem este o zi. Periodicitatea zi/noapte este semnificativă în trafic și poate fi o ancoră de consistență pe care o putem folosi în analiză. Există și fenomene periodice de durată mai lungă (de exemplu cicluri săptămânale) dar stocarea unei cantități așa de mare de date în agent nu prezintă avantaje semnificative. Pentru analiza la astfel de scări de timp se poate folosi stocarea în baza de date.

Ciclul de descărcare al datelor este un alt criteriu care este semnificativ în legătură cu adâncimea de stocare necesară. Chiar dacă analiza ar trebui să se facă cât mai frecvent posibil există situații în care server-ul de analiză este indisponibil pe o durată mai mare de timp. Am decis că o adâncime de stocare rezonabilă în memoria agentului poate să fie de 1 zi (24 ore).

Granularitatea intervalului minim de timp este de asemenea importantă și are directă legătură cu volumul de date care trebuie stocat. Dacă luăm în considerare că o zi are 86400 secunde o dimensiune a intervalului primar aferent unui eșantion din seria de date de o secundă conduce la 64*4*86400 octeți deci aproximativ 20 MB memorie. Pentru sistemele curente acest consum de memorie nu este neapărat important dar intenționăm să rulăm agentul și pe noduri mai puțin capabile, la limită chiar pe noduri care rulează Linux embedded. Pentru a obține un consum de memorie mai redus tebuie să acceptăm un interval mai mare, de 10 secunde. Din punct de vedere al rezoluției temporale eșantionarea la 10 secunde este rezonabilă. Durata atacurilor tipice de portscan sau DoS este mult peste 10 secunde.

Modulul NEAR-agent întreține așadar un buffer circular de seturi de 64 contoare de 32 biți în care la orice moment sunt disponibile seriile de date pe ultimele 24 ore. Mai întreține de asemenea liste cu cele mai frecvent întâlnite porturi neclasificate și histograme cu frecvența apariției diverselor dimensiuni de pachete respectiv cu frecvența apariției porturilor pe întraga gamă. Histogramele sunt colectate pentru scopuri strict experimentale și nu le vom discuta în continuare.

Tot acest set de date este pus la dispoziția unor module-client capabile de afișare/gestionare/analiză prin intermediul unei interfețe HTTP în format JSON. Am ales această combinație pentru că este ușor de folosit chiar fără un server dedicat de analiză. Formatul JSON este un concurent important al formatului XML, mai ales în spațiul aplicațiilor moderne datorită simplității sale și ușurinței de folosire în corelație cu limbaje script din familia Javascript. In cazul nostru am preferat acest format datorită faptului că are un overhead redus per element. Avem serii masive de valori care trebuie transportate și JSON este mai eficient în mod natural pentru acest caz.

Chiar și cu folosirea JSON, un pachet complet de date aferent ultimelor 24 ore ocupă peste 1 MB. Din acest motiv, agentul este capabil să servească imagini parțiale a setului de date (de exemplu ultimele k seturi de eșantioane sau toate seturile mai noi decât un anumit prag de timp). Un sistem de analiză care funcționează permanent poate să folosească această capabilitate pentru a face un transfer de informație incremental.

5.6. NEAR-GUI - instrument de vizualizare serii de date

Pentru a explora potenţialul de de detecţie a metodei bazate pe separarea de trafic am avut nevoie de un instrument care să permită în mod simplu şi comod

preluarea seriilor de date de la agenţii disponibili sau din fişiere procesate anterior şi afişarea acestora într-o formă intuitivă. In acelaşi instrument am inclus şi capabilităţi simple de transformare în domeniul frecvenţă respectiv de afişare a altor categorii de date furnizate de agenţi (de exemplu setul celor mai frecvente porturi neclasificate). Un astfel de instrument poate fi util şi în cadrul unui sistem complet, ca unealtă de diagnostic sau interogare punctuală, în special în absenţa server-ului de analiză.

Pentru implementare am ales limbajul Java şi o abordare grafică bazată pe SWT[84]. Acest mediu de programare asigură o bună portabilitate. Unealta dezvoltată va putea fi rulată cu uşurinţă ca aplicaţie locală pe multe sisteme de operare (Windows, Linux, OSX, ...).

Interfaţa aplicaţiei NEAR-GUI este prezentată în Figura 71. Aplicaţia permite deschiderea mai multor ferestre de vizualizare a seriilor de date. Fiecare fereastră este dedicată unui set de serii de date care provine fie din conectarea în timp real la un agent de captură şi extracţie, fie din deschiderea offline unui fişier produs de un astfel agent. Alinierea ferestrelor este aleasă intenţionat cu aspect de „stivuire”. In acest mod se poate deschide în mod repetat aceeaşi sursă de date dar cu perspective diferite astfel înct să se poată investiga vizual corelaţia între multiple serii.

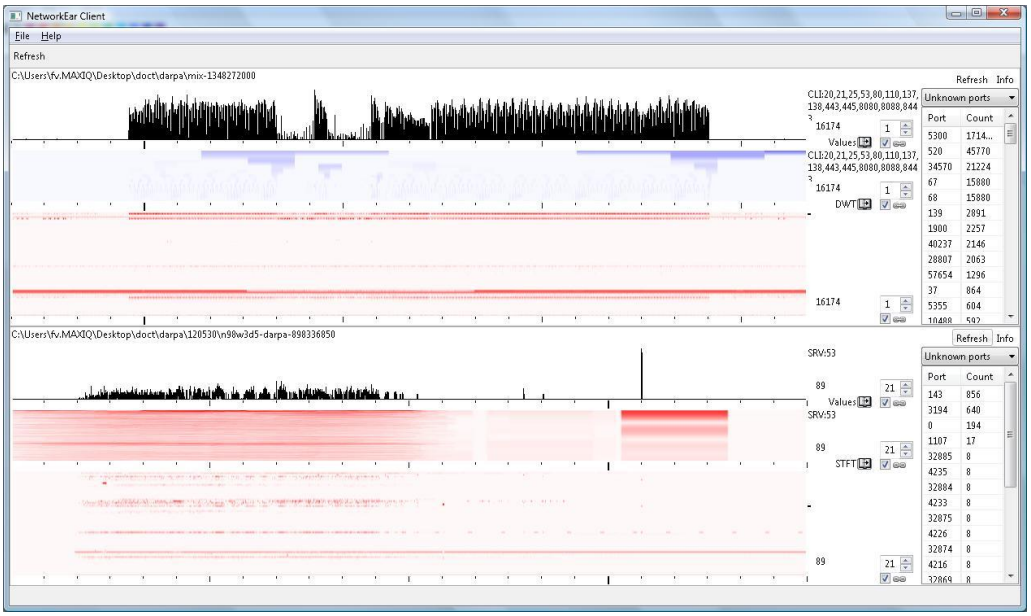


Figura 71. Interfaţa aplicaţiei NEAR-GUI

Exemplul din Figura 71 prezintă aplicaţia în care au fost deschise două astfel de ferestre care conţin date din fişiere produse anterior. Partea de sus conţine o captură generată prin monitorizarea server-ului S-CB din reţeaua RD. Partea de jos conţine o captură generată prin procesarea offline a capturilor din reţeaua DARPA (în particular ziua de vineri a săptămnii 3 din sesiunea DARPA’98).

Toate ferestrele au aceeaşi structură, destinată a oferi acces rapid şi comod la volumul mare de date conţinut de un set de serii de valori. Partea principală a

ferestrei este dedicată pentru trei reprezentări grafice. Cele două grafice din partea superioară au aceleași capabilități și sunt destinate afișării de informații despre o anumită serie din cele 64. Graficul din partea de jos a ferestrei reprezintă toate cele 64 serii de date pentru o localizare rapidă a zonelor de interes.

Ferestrele care afișează o serie de date pot să afișeze direct valori (sub forma de grafic de bare combinat cu „anvelopa" grafic poligonal) sau pot să afișeze rezultate ale unor transformări în domeniul timp-frecvență pe care le-am folosit la investigațiile inițiale. Aplicația suportă spectrogramă obținută via STFT și transformare DWT la care s-a compensat decimarea (pentru a păstra aspectul rectangular al graficului). Deoarece NEAR-GUI este deocamdată doar un instrument simplu de cercetare-explorare nu am prevăzut posibilitatea de a controla paremetrii transformărilor din interfața utilizator. Parametrii transformărilor se pot modifica prin ajustarea sursei și recompilare. Din același motiv nu există axe pe verticala graficelor. Acolo unde am avut nevoie de mai mult decât de o confirmare vizuală am folosit instrumente mai exacte (de exemplu analiza manuală în Matlab).

In figură se pot vedea atât exemple de serii de valori (graficele din partea de sus a ferestrelor) cât și exemple de reprezentări a rezultatelor transformărilor timp-frecvență. Pentru acestea din urmă, prezentarea valorilor pe axa amplitudine este făcută folosind intensitatea culorii.

Secțiunea din dreapta a ferestrei este dedicată elementelor de control. Putem să modificăm indexul seriei de date afișate, tipul de reprezentare și putem să legăm între ele controalele care selectează seria curentă astfel încât cu manevre simple (single-click sau săgeți sus/jos) să putem explora rapid întregul set de serii urmărind simultan valori și una din transformările disponibile. Se afișează și semnificația seriei curente corelat cu situația de configurare de la momentul capturii. In extrema dreaptă este prezentă lista cu cele mai frecvent întâlnite porturi (per total sau doar cele neclasificate).

5.7. Atacuri prezente în seturile DARPA, analizate cu NEAR

Pentru a putea reflecta atacurile identificabile prin mecanismele propuse și descrise mai sus am rulat procesul de analiză offline implementat de NEAR-agent asupra capturilor furnizate de seturile de date DARPA. Am obținut 99 fișiere care conțin reprezentarea JSON a analizelor zilnice care ar fi fost expuse de funcționarea NEAR-agent pe rețeaua DARPA. Din aceste fișiere 48 sunt rezultate pentru DARPA'98 și 51 pentru DARPA'99.

Fișierele DARPA'98 sunt doar pentru captură pe segmentul „exterior" iar cele din DARPA'99 sunt atât pentru segment „exterior" cât și pentru segment „interior". Am organizat aceste fișiere cu o convenție de nume care reflectă în numele fișierului anul, saptămâna, ziua, zona (dacă e cazul) și un timestamp (msec față de 00:00:00, 01 Ianuarie 1970 conform reprezentării tradiționale UNIX). Includerea acestui timestamp a fost necesară deoarece pentru anumite zile oficiale de simulare au rezultat mai multe fișiere cu rezultate de analiză.

Deși setul DARPA'98 ar fi trebui să conțină 35 de fișiere zilnice (7 săptămâni a 5 zile) există unele anomalii care au făcut să rezulte 48 fișiere. Există capturi aferente unor zile care în realitate conțin date pentru mai mult decât o singură zi (Tabel 6). Deoarece NEAR-agent produce câte un fișier offline la fiecare traversare a orei 11:00 GMT și unul la încheierea capturii, aceste zile au produs mai mult decât un singur fișier cu rezultate de achiziție serii de date. Ora 11:00 GMT a fost aleasă

pentru că se reflectă în 06:00 EST, oră la care „începe ziua de captură" conform descrierii experimentului.

Săpt.	Ziua	Interval acoperit de captură
2	2 (marți)	Conține de fapt pachete din 3 zile diferite, de la 1998-06-09T07:56:12-05:00 până la 1998-06-12T05:40:30-05:00[4]
3	1(luni)	Conține de fapt pachete din 5 zile diferite, de la 1998-06-15T07:51:46-05:00 până la 1998-06-19T03:37:20-05:00
3	4(joi)	Conține de fapt pachete din 4 zile diferite, de la 1998-06-18T07:59:18-05:00 până la 1998-06-21T03:21:35-05:00
5	5(vineri)	Conține de fapt pachete din 2 zile diferite, de la 1998-07-03T07:56:21-05:00 până la 1998-07-04T04:10:27-05:00
6	1(luni)	Conține de fapt pachete din 4 zile diferite, de la 1998-07-06T07:59:01-05:00 până la 1998-07-09T03:35:50-05:00
7	2(marți)	Conține de fapt pachete din 2 zile diferite, de la 1998-07-14T07:56:33-05:00 până la 1998-07-15T05:59:03-05:00
7	4(joi)	Conține de fapt pachete din 3 zile diferite, de la 1998-07-16T07:54:28-05:00 până la 1998-07-19T05:38:25-05:00

Tabel 6. Capturi care depășesc o zi în setul DARPA'98

Și setul de date din 1999 conține câteva astfel de situații speciale, detaliate în Tabel 7

Săpt.	Ziua	Zonă	Interval acoperit de captură
1	1(luni)	IN	Conține de fapt pachete din 2 zile diferite dar mai puțin de 24 ore, de la 1999-03-01T08:00:05-05:00 până la 1999-03-02T06:00:16-05:00. Din cauza orei de salvare automată rezultă un fragment suplimentar, nerelevant, între 06:00 și 06:16
2	1(luni)	OUT	Conține de fapt pachete din 2 zile diferite dar mai puțin de 24 ore, de la 1999-03-08T08:00:01-05:00 până la 1999-03-09T06:00:49-05:00. Din cauza orei de salvare automată rezultă un fragment suplimentar, nerelevant, între 06:00 și 06:49
4	2 (marți)	IN	Captura pentru „inside" lipsește complet. Lipsa ei este documentată în observațiile setului DARPA'99

Tabel 7. Capturi care prezintă situații speciale în setul DARPA'99

[4] Seturile DARPA au fost colectate la MIT (Massachusetts) in timezone EST, cu offset -5 față de GMT. În fișierele de captură PCAP sunt stocate marcaje de timp UTC/GMT.

Față de alte surse de capturi de pachete, seturile de date DARPA prezintă avantajul că atacurile prezente sunt (cel puțin în mare parte) inventariate și documentate. Am profitat de acest avantaj și am investigat enumerarea de atacuri din perspectiva analizei pe care o putem face folosind instrumentele descrise.

In urma acestei analize am constatat că există o serie de atacuri pentru care avem o șansă bună de detecție (folosim nomenclatura originală din documentele DARPA): arppoison, ipsweep, mailbomb, neptune, nmap, PoD, portsweep, resetscan, SATAN, smurf, tcpreset. Există de asemenea un număr de atacuri care ar putea fi eventual detectate dar numai în condiții favorabile: back, crashiis, dict, dosnuke, guest, httptunnel, insidesniffer, land, ls_domain, ncftp, netbus, phf, queso. Există în final și un număr de atacuri care prin natura lor nu se pot detecta folosind doar instrumente orientate către procesarea traficului de rețea și pe care nu le enumerăm aici.

In continuare vom analiza sumar atacurile detectabile și vom prezenta unele serii procesate care ilustrează respectivele atacuri. Descrierile atacurilor sunt preluate din documentația online DARPA [78] și din teza lui K. Kendall [77]. Pentru cazurile unde există referințe despre momentul în care sunt prezente instanțe ale atacurilor enumerăm seriile care conțin atacuri sub forma compactă nYYwWWdDD pe care am folosit-o și pentru etichetarea fișierelor de analiză, unde:

- YY este anul (98 sau 99)
- WW este săptămâna simulării (1..7 pentru DARPA'98 și 1..5 pentru DARPA'99)
- DD este ziua din săptămână (1..5 pentru luni..vineri)

Spre exemplu n98w6d4 se referă la date colectate din setul DARPA'98, săptămâna 6, ziua 4 (joi).

5.7.1. arppoison

Descriere DARPA/Kendall: *„An arp-level denial of service, where the attacker sends out bogus responses to "arp-who-has" requests for the victims mac address. In order to carry out this attack, the attacker must gain access on a machine on the victim's subnet, so it often involves a remote attacker logging into a local machine, then running the attack against another machine (the victim)."*

Atacul a apărut doar în setul de date din 1999 și din păcate poziția atacurilor nu a fost explicit prezentată. Putem presupune doar că acest atac, pentru a fi eficient, produce un volum semnificativ de trafic ARP cu origine pe o singură stație. In modelul propus de noi, atacul ar putea fi detectat prin volum crescut de trafic pe seria ARP colectată de unul din nodurile de rețea. Dacă am trata separat traficul ARP incoming și outgoing (care actualmente sunt agregate pe index 50) și agentul poate fi prezent pe stația atacatoare, șansele de detecție sunt semnificativ mai bune. Deoarece nu am avut confirmare explicită despre poziția atacului nu am investigat această posibilitate.

5.7.2. ipsweep

Descriere DARPA/Kendall: *„Surveillance sweep performing either a port sweep or ping on multiple host addresses."*

Atacul apare atât în setul 1998 cât și în setul 1999, în mai multe poziții documentate: n98w2d2, n98w3d3, n98w4d3, n98w4d5, n98w5d2, n98w5d3, n98w6d3, n98w6d4, n98w7d3, n98w7d4, n99w2d2, n99w2d3, n99w2d4. Din

descriere, din nume și din faptul că există un atac numit portsweep înțelegem că este o explorare pe același port a mai multor adrese IP diferite sau o accesare ICMP a unei serii de adrese. In funcție de intensitatea unui astfel de atac și de porturile vizate ne putem aștepta să fie vizibil pe seriile de port aferente, pe seria de porturi neclasificate sau pe seria de ICMP, sub forma unei platforme sau sub forma unor apariții periodice. Pentru datele DARPA'99 putem să urmărim și seria ARP, care ar putea să conțină un trafic ARP sporit, din cauza explorării secvențiale a unui spațiu mai mare de adrese IP.

In Figura 72 sunt prezentate două astfel de cazuri. In partea de sus avem un atac ipsweep detectabil clar pe seria ARP, prezentat pe fișierul de analiză n99w2d2-darpain. Se observă din diagrama exploratorie din partea de jos a ferestrei superioare că atacul ar fi vizibil și pe alte serii de date, dar rezultatul cel mai bun se obține cu seria ARP. Dacă se aplică un proces de voting sau de corelație între serii rezultatele pot fi și mai bune.

In partea de jos avem un alt caz de ipsweep unde se poate profita de principiul SNR pentru a detecta atacul cu succes pe seria porturilor neclasificate deoarece traficul neclasificat a fost deosebit de redus. Atacul este prezentat pe seria n98w2d2 pentru care informația din seria ARP nu este relevantă (este vorba de captură externă iar atacul se derulează asupra nodurilor din interior).

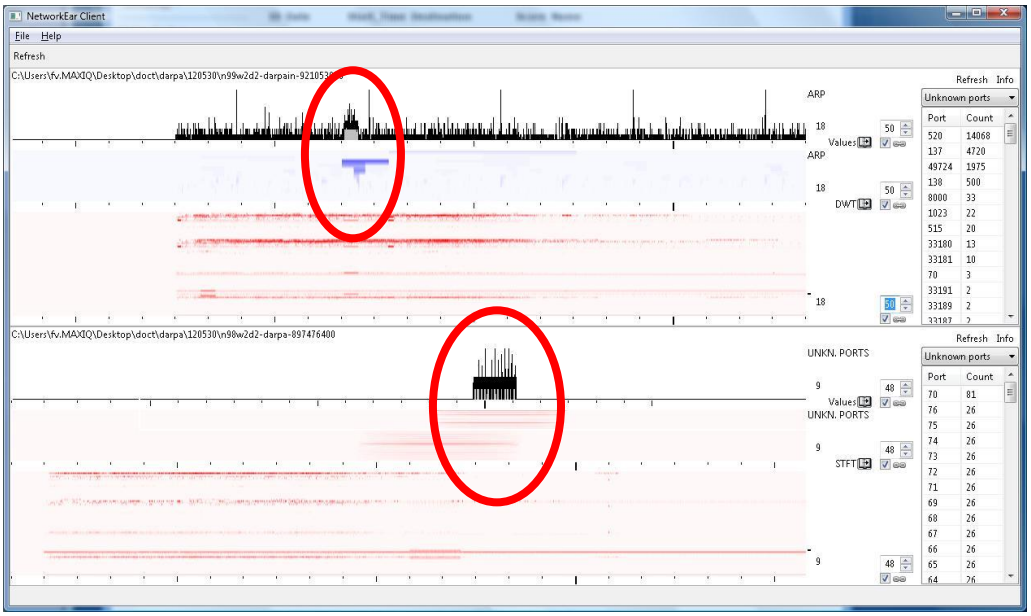


Figura 72. Atacuri ipsweep evidenţiate în NEAR-GUI

Am încercat aplicarea algoritmilor de detecție descrișI anterior asupra acestor serii de date. Această parte a procesării a fost scrisă în Matlab și este pur experimentală (nu are o interfață de operare ci este doar cod). Rezultatele se regăsesc în Figura 73 şi Figura 74. Fiecare figură conţine 4 grafice notate cu a), b), c) şi d).

Graficele a) reprezintă seriile de valori rezultate din achiziţia executată de NEAR-agent. Pentru Figura 73 este seria de index 50 adică seria contoarelor pentru

apariţii de pachete ARP. Pentru Figura 74 este vorba de seria de index 48, adică seria contoarelor pentru apariţii de pachete cu porturi neidentificate. Abscisa este index în cele 8640 valori pentru că în analiza Matlab nu am mai aplicat calculul de interval de timp prezent în NEAR-GUI.

Graficele b) sunt o reprezentare aproximativă a magnitudinii coeficienţilor la cele 9 nivele de descompunere folosite, după aplicarea pragurilor. Reprezentarea serveşte la prezentarea intuitivă a distribuţiei coeficienţilor în încercarea de a găsi un algoritm cât mai bun de detecţie.

Graficele c) sunt o reprezentare a semnalului reconstituit după filtrul care folosește coeficienții trunchiați și nu include componenta de aproximare. Sunt prezentate pragurile 6-sigma și pragul 8-sigma sub formă de linii punctate.

Graficele d) reprezintă ieșirea detectorului cu prag la 6-sigma. Pentru cazul atacului ipsweep din DARPA'99 (Figura 73) se observă că detectorul este declanșat atât de anomalia produsă de ipsweep în zona abscisei 3700-3900 cât și de vârfurile prezente în pozițiile aproximative 2400, 3900, 5250, 6800, 8200. Din analiza manuală a fișierului de captură rezultă că în acele poziții au existat secvențe compacte de interogare ARP generate de router spre stații din interior. Aspectul periodic sugerează că este vorba de o expirare forțată periodică a cache-ului de ARP din router.

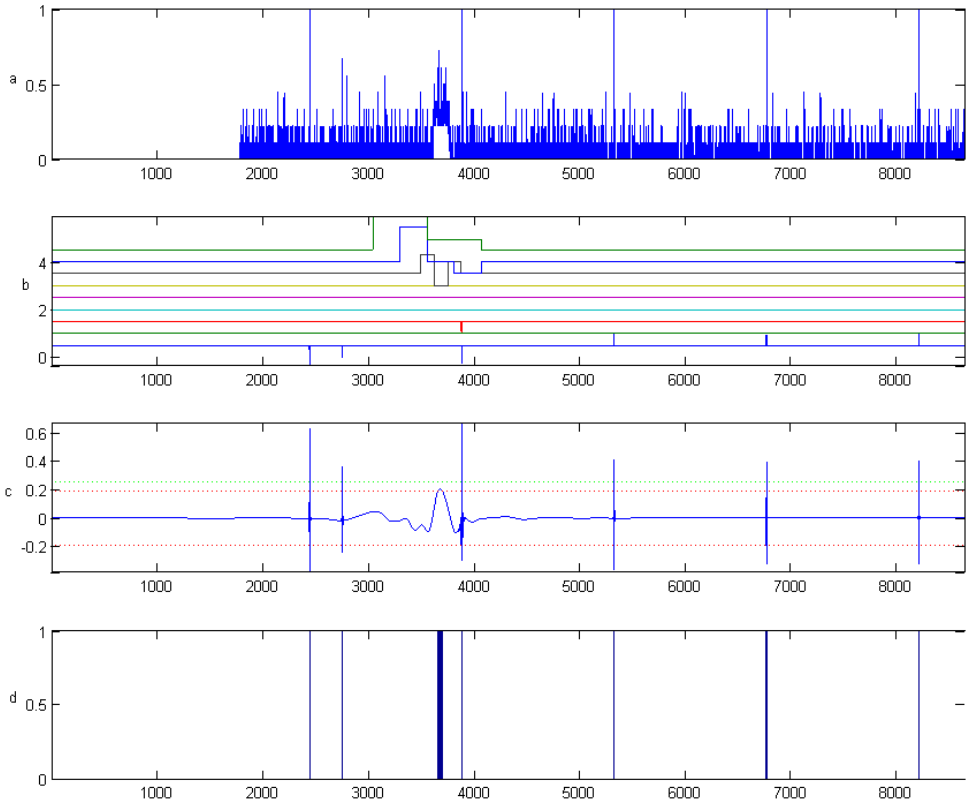


Figura 73. Detecţia de anomalie pentru seria 50 (ARP) din n99w2d2-darpain.

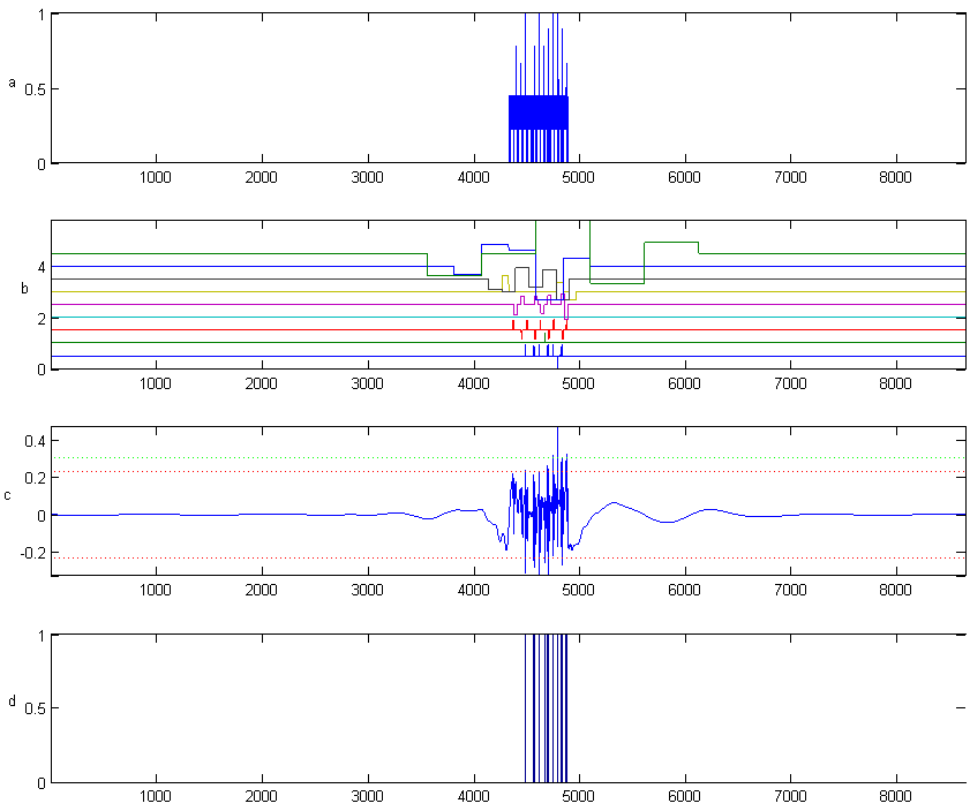


Figura 74. Detecţia de anomalie pentru seria 48 (UNKN PORT) din n98w2d2

Cazul atacului ipsweep din n98w2d2 este mai simplu pentru că în afara atacului practic nu a existat trafic pe această serie de date şi principiul SNR se aplică în mod ideal. In consecinţă se observă o suprapunere foarte bună a detecţiei peste perioada perturbată, fără detecţii nedorite. Dacă inspectăm vizual diagrama c). se observă că aplicarea pragului 6-sigma este suficientă în acest caz dar şi că un prag mai agresiv (8-sigma) nu ar fi mascat perturbaţia ci doar ar fi întârziat detecţia.

5.7.3. mailbomb

Descriere DARPA/Kendall: *„A Denial of Service attack where we send the mailserver many large messages for delivery in order to slow it down, perhaps effectively halting normal operation."*

In mod neașteptat atacul nu este descris ca nou în setul DARPA'99 dar nici nu este nominalizat exact în setul DARPA'98. In setul DARPA'99 el apare nominalizat în n99w2d2 și n99w2d3. Deoarece atacul implică un volum mare de date care să înece capabilitatea de procesare a server-ului de mail ne așteptăm ca acesta să se manifeste ca o platformă sau un impuls pe seria care clasifică portul 25 dedicat transferului SMTP. Această comportare se confirmă pe datele reale, după cum se vede în Figura 75.

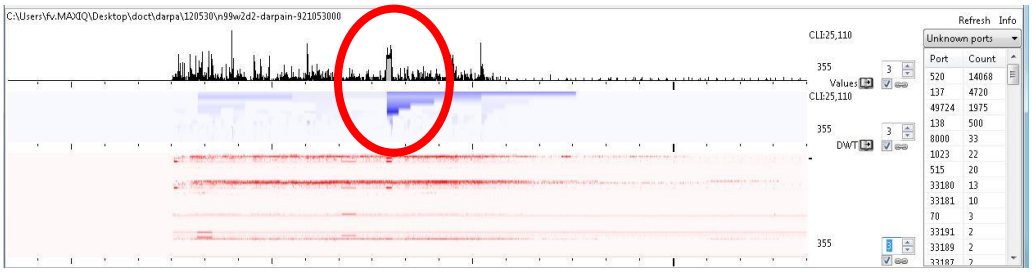


Figura 75. Atac mailbomb pe seria porturilor de e-mail (25, 110)

5.7.4. neptune

Descriere DARPA/Kendall: *„Syn flood denial of service on one or more ports."*

Atacul este prezent în multe poziții pentru ambele simulări: n98w1d1, n98w3d4, n98w4d2, n98w5d4, n98w5d5, n98w6d1, n98w6d3, n98w6d4, n98w6d5, n98w7d5, n99w2d4, n99w2d5. Forma de atac este o inundare cu pachete SYN care ar trebui să se vadă pe seria aferentă portului atacat. Dacă e un atac care acoperă mai multe porturi, mai multe serii vor fi afectate, ceea ce face atacul mai ușor de identificat.

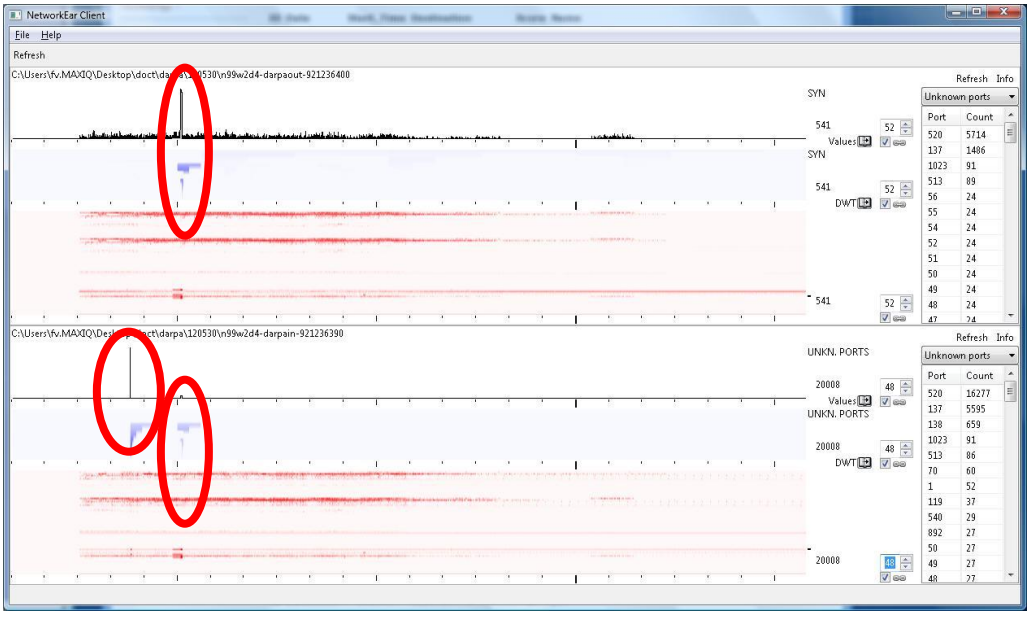


Figura 76. Atac neptune din exterior şi SATAN din interior pe seriile n99w2d4. Seria de sus este capturată în exterior, seria de jos este capturată în interior. Atacul din stânga pe seria de jos este SATAN.

Pentru exemplificare am ales n99w2d4 pe care am investigat atât seriile de interior cât și seriile de exterior. Rezultatul este prezentat în Figura 76. Se observă că atacul neptune pe seria din exterior se poate detecta ușor folosind seria SYN dar

este prezent și pe seria porturilor necunoscute deoarece nodurile de rețea din acea perioadă nu foloseau nici o tehnică stealth (deci răspundeau cu RST pe porturi închise). Același atac este mult mai greu de detectat pe captura din interior, dar numai din cauză că este acoperit ca amplitudine de un atac SATAN care a avut loc cu aproximativ 90 minute înainte.

5.7.5. nmap

Descriere DARPA/Kendall: *„Network mapping using the nmap tool. Mode of exploring network will vary - options include SYN."*

Atacul apare în n98w3d3, n98w3d4. Capabilitățile de scan ale utilitarului nmap sunt multiple și poate să ofere diverse grade de agresivitate. Nu toate vor fi detectabile, mai ales în condițiile în care prin configurare scanarea poate să fie făcută cu o rata de 1 pachet pe minut sau și mai lent. Există totuși posibilitatea ca scanarea să fie capturată ca un tren periodic de valori reduse cu perioadă lungă, pe seriile de porturi neclasificate. Spre exemplu atacul nmap din n98w3d3 este de lungă durată (12 ore) dar are o puternică componentă periodică cu perioada 60 secunde pe întreaga durată a atacului. Versiunile curente de nmap permit introducerea unei componente aleatoare atât în secvențierea pachetelor cât și în intervalul dintre pachete [85].

5.7.6. PoD

Descriere DARPA/Kendall: *„Denial of service ping of death."*

Atacul este istoric chiar din punct de vedere al constatărilor experiențelor DARPA pentru că se bazează pe un viciu de implementare în stiva TCP/IP a diverselor sisteme de operare care a fost între timp corectat. De fapt chiar în descrierea DARPA se menționează că nici una din mașinile atacate nu a fost afectată de acest atac. Atacul apare în multiple poziții documentate (chiar dacă momentele de timp documentate nu corespund întotdeauna cu conținutul capturilor): n98w1d4, n98w4d1, n98w4d2, n98w4d3, n98w5d2, n98w5d4, n98w6d2, n98w6d4, n98w7d2, n99w2d1, n99w2d5.

Ca implementare atacul presupune trimiterea către țintă a unui pachet IP supradimensionat. Datorită faptului că există o limitare clară a dimensiunii cadrului Ethernet acest pachet IP va fi fragmentat și vor rezulta multe cadre distincte (peste 40 cadre pentru un singur pachet). Procesul de captură nu face reasamblare deci din punctul de vedere al contoarelor colectate de NEAR-agent vom constata un număr mare de pachete ICMP care produc un impuls semnificativ și ușor de detectat pe seria corespunzătoare.

5.7.7. portsweep

Descriere DARPA/Kendall: *„Surveillance sweep through many ports to determine which services are supported on a single host."*

Ca principiu de manifestare acest atac este foarte apropiat de nmap și efectiv poate fi implementat folosind utilitarul nmap. In seturile DARPA atacul nu este descris în detaliu dar este documentat ca prezent în ambele instanțe: n98w2d1, n98w3d1, n98w4d3, n98w4d4, n98w5d2, n98w5d4, n98w5d5, n98w6d2, n98w6d4, n98w7d2, n98w7d3, n99w2d2, n99w2d4, n99w2d5.

Deoarece metoda de atac este o simplă încercare de conectare succesivă la toate porturile unui nod destinație atacul este vizibil pe seria porturilor neclasificate.

5.7.8. resetscan

Descriere DARPA/Kendall: *„ResetScan sends reset packets to a list of IP addresses in a subnet to determine which machines are active. If there is no response to the reset packet, the machine is alive. If a router or gateway responds with "host unreachable," the machine does not exist.”*

Atacul este menționat în lista de atacuri ca fiind atac nou în DARPA’99 dar nu este documentat ca poziție în lista de atacuri aferentă săptămânii 2, singura din set care are atacurile documentate. Ca prezentare atacul ar trebui să se manifeste ca un număr mare de pachete ARP generate de explorarea completă a spațiului de adrese intern și un număr mare de pachete ICMP cu raportul HOST-UNREACHABLE.

5.7.9. SATAN

Descriere DARPA/Kendall: *„Network probing tool which looks for well-known weaknesses. Operates at three different levels. Level 0 is light.”*

Atacul apare în multiple poziții documentate: n98w3d1, n98w4d2, n98w5d1, n98w5d4, n98w6d1, n98w6d4, n98w7d1, n98w7d4, n99w2d3, n99w2d4. Deoarece scopul este explorarea rețelei pentru a detecta vulnerabilități atacul se derulează pe un interval de timp scurt și generează impulsuri semnificative pe mai multe serii. Un exemplu relevant l-am prezentat deja în Figura 76, care ilustrează un nivel intens.

5.7.10. smurf

Descriere DARPA/Kendall: *„Denial of service icmp echo reply flood.”*

Atacul este documentat pentru DARPA’98 în pozițiile n98w1d3, n98w3d3, n98w4d1, n98w5d1, n98w5d4, n98w5d5, n98w6d4, n98w6d5, n98w7d4, n98w7d5. Am identificat prin explorare zi-cu-zi o anomalie pe seria ICMP și în n99w4d1-darpaout, chiar dacă săptămâna 4 nu are atacuri documentate (de fapt a fost chiar o săptămână de test). Am confirmat prezența unui atac prin analiza manuală a capturii aferente și am identificat atacul smurf. Cu această ocazie trebuie să observăm încă o dată că metodele descrise pot să semnaleze o anomalie chiar independent de cauza acesteia dar nu pot să identifice cauza chiar dacă e una binecunoscută. Sarcina identificării cauzei revine altor mecanisme, eventual de tip pattern-matching.

5.7.11. tcpreset

Descriere DARPA/Kendall: *„A Denial of Service where the attacker, generally sitting on the same subnet as the victim, resets any tcp connections that it sees go through the handshake phase with the victim. To do this it spoofs the victims ip on the reset packets.”*

Atacul este clasificat ca fiind nou în DARPA’99 dar nu există nici o instanță documentată pentru săptămâna 2. Ca prezentare ar putea să conțină o oarecare intensificare a traficului destinat portului pe care se execută DoS în cazul în care comportamentul tipic al clientului este de a reîncerca operația de conectare. Dacă

avem de-a face cu un client non-persistent atunci anomalia nu poate fi distinsă de un reset normal al conexiunii.

5.8. Secvențe din rețele proprii, analizate cu NEAR

Pentru a completa imaginea capabilităţilor de detecţie oferite de metoda de captură şi extracţie pe care o propunem în această lucrare vom prezenta în continuare câteva exemple de capturi. Aceste exemple ilustrează situații în care se pot evidenția evenimente care pot fi considerate anomalii și care pot necesita investigație suplimentară sau modificarea configurației agenților de captură pentru a obține mai multă informație.

5.8.1. Absenţă trafic în reţeaua EDU

O categorie de anomalie detectabilă este opusul impulsului prezentat în exemplele anterioare și anume lipsa completă sau reducerea semnificativă a valorilor din serie pe un interval compact în timp. O astfel de situaţie a fost capturată în reţeaua EDU şi este prezentată în Figura 77. In prezentarea situaţiei se folosesc denumirile din diagrama reţelei EDU (Figura 69).

Fereastra de sus conţine serii capturate pe interfaţa lui GW-D spre reţeaua de campus (eth0) iar fereastra de jos conţine serii capturate pe interfaţa lui GW-D spre reţeaua interioară de laborator (eth2). Se observă lipsa valorilor pe toate seriile din interfaţa eth0 şi pe multe serii din interfaţa eth2. Această lipsă de semnal poate fi uşor detectată ca anomalie.

Explicaţia lipsei complete a valorilor este o cădere de tensiune care a afectat probabil întreaga clădire. Nodul GW-D este protejat de o sursă UPS şi a agenţii instalaţi pe el au continuat să monitorizeze interfeţele. Interfaţa spre reţeaua de campus este conectată într-un switch neprotejat de UPS (parte a reţelei de campus) care nu a funcţionat pe durata căderii de tensiune şi deci pe eth0 nu s-a constatat nici un fel de trafic.

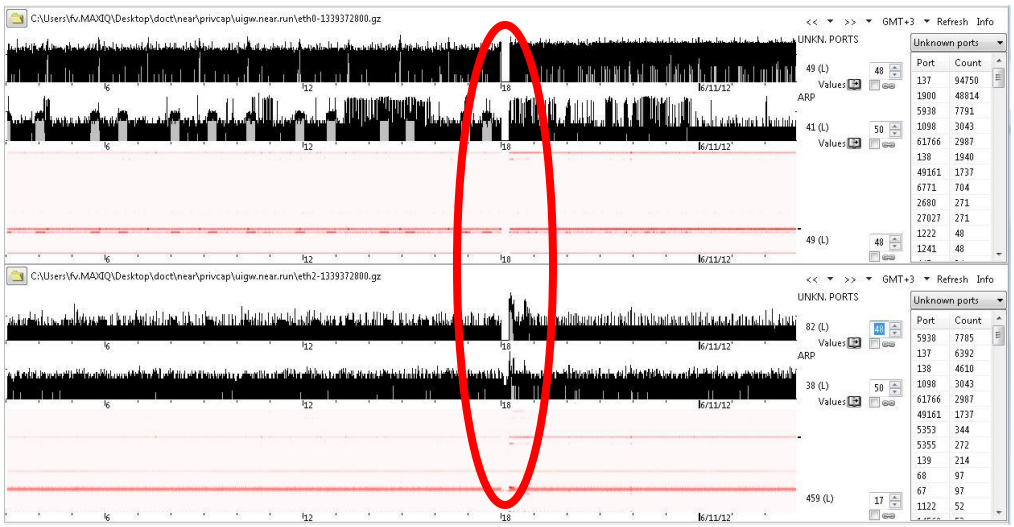


Figura 77. Cădere de tensiune temporară reflectată în seriile de date (reţeaua EDU)

Interfața spre rețeaua de laborator este conectată în switch-ul de prim nivel (conform descrierii rețelei EDU) care este protejat de UPS. Din păcate switch-urile de nivel doi nu sunt protejate așa încât accesul stațiilor W-Ln la restul rețelei a fost blocat pe durata căderii de tensiune. Din acest motiv traficul pe majoritatea seriilor are valori zero. Există totuși trafic ARP (se vede în graficul 2 al ferestrei de jos) generat de activitatea de fond a nodurilor care au fost protejate de UPS-uri și au fost conectate în switch-ul de prim nivel (serverele S-L, S-AD, S-DB). Pe durata căderii de tensiune nodul GW-D(eth2) mai detectează și trafic de sincronizare timp și syslog produs direct de server-ul Linux S-L (pe o serie care nu e reprezentată în figură).

Se observă de asemenea creșterea temporară a traficului pe seria porturilor neclasificate imediat după revenirea tensiunii de alimentare. Din analiza statisticilor pe porturile neclasificate o presupunere rezonabilă este că e vorba de trafic produs de stațiile Windows pe porturile NETBIOS 137/138[5] imediat după ce s-a restabilit conectivitatea pentru a confirma restabilirea disponibilității resurselor.

5.8.2. Atac fals în rețeaua EDU

Un exemplu de situație specială cu potențial de anomalie dar care își schimbă natura în urma corelației între agenți este prezentat în Figura 78. Și în această figură fereastra de sus conține capturi pe interfața eth0 din GW-D (spre rețeaua de campus) iar fereastra de jos conține capturi pe interfața eth2 din GW-D (spre rețeaua de laborator). Graficele care au numărul maxim de pachete urmat de marcajul „(L)" sunt reprezentate în scară logaritmică pentru a permite o vizualizare mai bună a valorilor mici din serie.

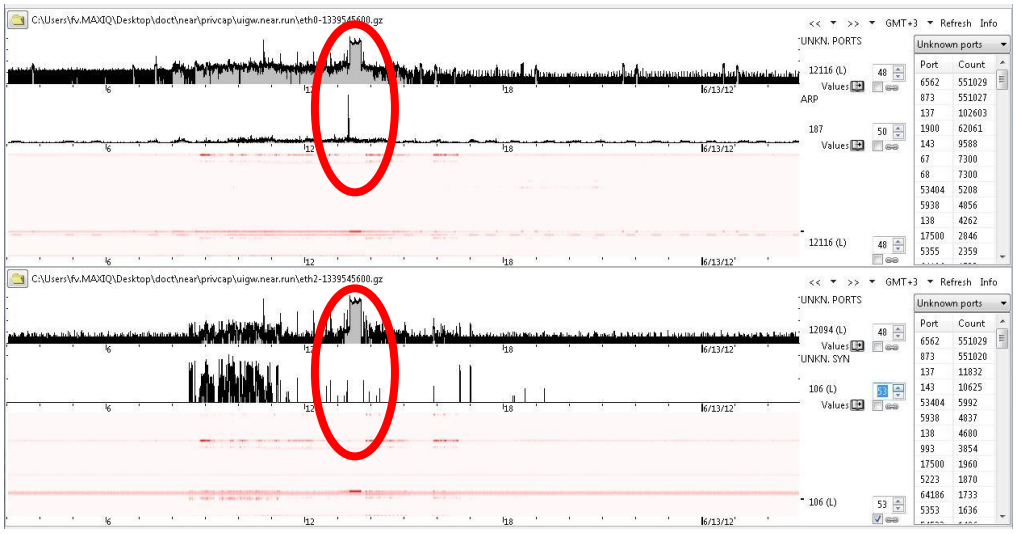


Figura 78. Trafic neașteptat pe port necunoscut. Sus: anomalie ARP şi trafic pe porturi neclasificate. Jos: Aceeaşi semnătură de trafic pe porturi neclasificate, fără corespondenţă în SYN-neclasificate,

[5] Datorită unei erori de configurare pentru aceste capturi, porturile 137 și 138 nu au fost corect configurate în clasă separată NETBIOS și apar pe seria porturilor neclasificate

Din analiza doar a seriilor produse de GW-D(eth0) (fereastra de sus) am putea să tragem concluzia că avem de-a face cu un atac explorator peste multiple IP-uri (din cauza impulsului pe ARP de la ~13:20) urmat aproape imediat de un atac cu scanare de porturi (valori semnificativ de mari pe seria porturilor neclasificate, cu o durată relativ lungă).

Dacă luăm în considerare și datele oferite de GW-D(eth2) constatăm că traficul pe porturi neclasificate provine din interiorul rețelei de laborator dar nu există deschideri de sesiune (SYN) pe aceste porturi neclasificate. Avem în schimb un număr mare de pachete alocate porturilor 6562 (anonim) și 873 (rsync) pe statistica porturilor neclasificate, cu valori relativ apropiate, ceea ce ne face să bănuim că este vorba de fapt de un transfer rsync. Ca măsură de îmbunătățire a calității monitorizării această constatare recomandă includerea portului 873 în clasa porturilor transfer de date care la momentul capturii conținea doar porturile 20 și 21 (FTP).

Explicaţia anomaliei bazată pe trafic eventual legitim de RSYNC nu explică şi anomalia de ARP care am interpretat-o ca scan de subreţea IP. Din analiza altor zile de captură am constatat un impuls similar ca amplitudine şi prezentare, la aceeaşi oră, cu o zi înainte. Nu am găsit o explicaţie rezonabilă pentru acest comportament care nu s-a repetat aşadar nu putem decât să îl considerăm o coincidenţă.

5.8.3. Trafic periodic normal în reţeaua RD

Am explorat şi seriile colectate din reţeaua RD unde am găsit mai multe exemple semnificative legate de periodicitatea traficului generată de procese periodice sau chiar de natura protocoalelor de nivel superior. In continuare ne referim la structura reţelei RD folosind denumiri introduse în Figura 68.

O pereche de serii de interes sunt seria de index 19 pe interfaţa internă a nodului de graniţă GW-I respectiv seria de index 3 pe interfaţa unică a server-ului S-CB, prezentate în Figura 79 aşa cum apar ele în interfaţa de explorare NEAR-GUI..

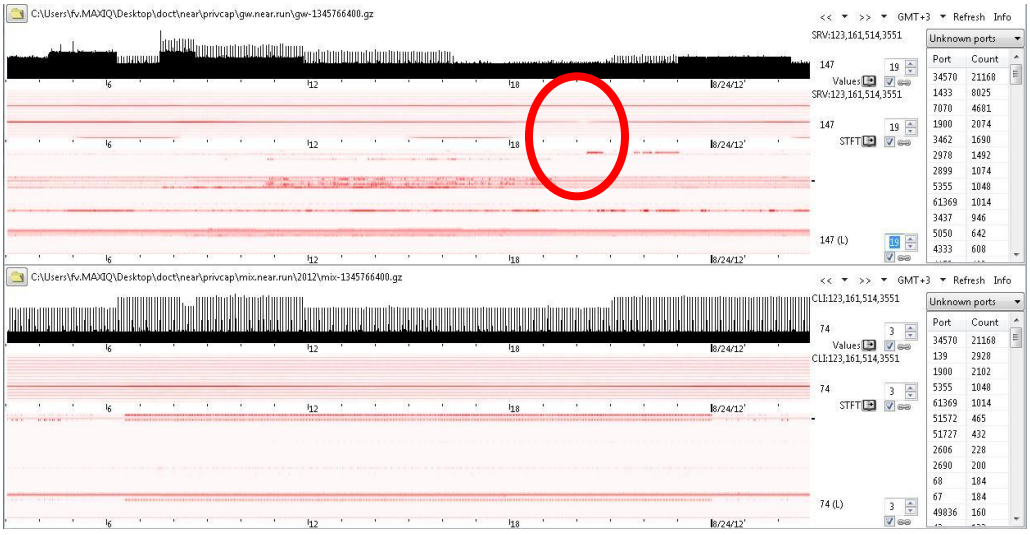


Figura 79. Trafic periodic între S-CB (jos) şi GW-I (sus). Marcajul indică o discontinuitate posibil interesantă.

Pentru ambele noduri am selectat clasa de trafic „sistem” care conținea la momentul capturii porturile 123, 161, 514, 3551. Seria selectată pentru GW-I este seria „server” iar seria selectată pentru S-CB este seria „client”. Nodurile monitorizate sunt într-o relație de tip client-server pe (o parte din) protocoalele aferente, așa că ar trebui să constatăm un oarecare grad de corelație între cele două serii. Așteptările pentru aceste serii sunt date de specificul traficului în și particularitățile în rețeaua RD și anume:

- traficul pe port 123 (ntp) între cele două mașini (ambele rulează Linux) ne așteptăm să fie vizibil doar pe seria SYM (index 35) pentru că Linux NTP folosește ca port de plecare tot 123. Există trafic pe port 123 între stațiile Windows și GW-I dar nu ne așteptăm să fie vizibil pentru că este extrem de scăzut
- ne așteptăm să existe trafic pe port 161 (snmp) generat de un proces mrtg care rulează pe S-CB și care interoghează periodic (la 5 minute) multiple servere, inclusiv GW-I. Traficul SNMP vizibil pe seria SRV la GW-I este doar o parte din traficul generat de S-CB pentru că acesta monitorizează mai multe servere.
- traficul pe port 514 (syslog) va fi vizibil doar pe seriile CLI (index 3) iar S-CB nu este configurat să facă remote-syslog.
- ne așteptăm să existe trafic pe port 3551 (apcupsd) generat de S-CB care verifică periodic disponibilitatea sursei UPS prin interogarea GW-I. Traficul apcupsd vizibil pe seria de jos (S-CB) este doar o parte din traficul primit de GW-I pentru că există mai multe servere care folosesc același UPS.

Din aspectul seriilor de valori privite ca evoluție în timp nu se poate desprinde o concluzie specială în afară de faptul că există o componentă periodică. Pe graficele exploratorii STFT se observă mult mai bine o componentă periodică importantă atât pentru GW-I cât și pentru S-CB și o componentă periodică adițională mai puțin prezentă și de frecvență mai joasă (frecvența zero pe graficul STFT este sus). Se mai observă o discontinuitate în a doua armonică a componentei semnificative de la GW-I și componente ocazionale de armonica 3.

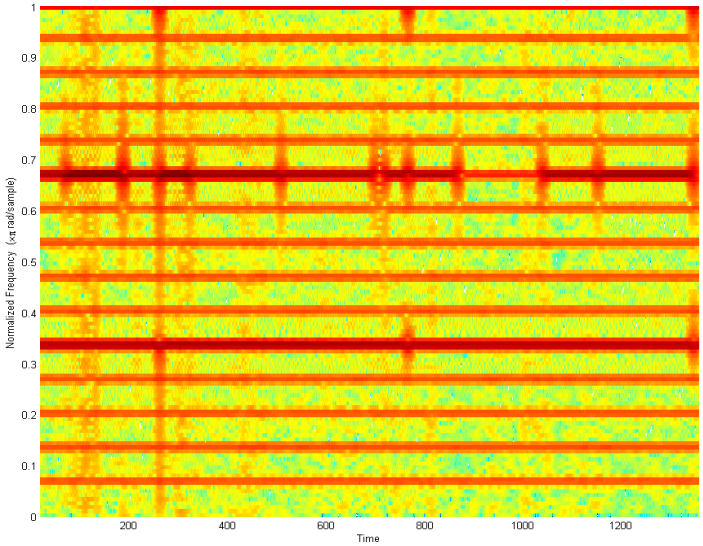


Figura 80. Transformata STFT pentru seria GW-I, index 19. Reprezentare implicită Matlab, normalizat.

Pentru că interfața NEAR-GUI are deocamdată capabilități limitate la explorare și nu poate prezenta detalii, am analizat seriile de date folosind manual capabilitățile de reprezentare mai performante din Matlab și o serie de script-uri dezvoltate special pentru procesarea comodă a datelor colectate de NEAR-agent.

Am extras în Matlab cele două serii și am aplicat pe fiecare STFT (funcția spectrogram) cu fereastră de 256 puncte, offset 250, 256 puncte FFT. In Figura 80 este reprezentată transformata STFT pentru GW-I index 19, în prezentarea implicită Matlab. Această serie este aceeași serie din diagrama de sus marcată în Figura 79. Se observă aceeași discontinuitate în una din armonicile prezente altfel sistematic pe toată durata de analiză, discontinuitate care poate fi suspectată ca un semn de anomalie în funcționare.

Reprezentarea este puțin mai clară decât cea oferită de NEAR-GUI datorită rezoluției sporite de reprezentare dar gradația de culoare nu oferă suficient suport intuitiv pentru o interpretare confortabilă. Din acest motiv am reluat reprezentarea în formă de suprafață 3D, în Figura 81, ținând cont de scalarea specifică perioadei originale de eșantionare.

Din reprezentarea mai intuitivă făcută în Figura 81 se observă imediat elemente clare de identificare a traficului care ar putea să fie exploatate automat. Pentru o claritate mai bună a reprezentării am eliminat componenta de medie care ar fi fost altfel prezentă în partea din stânga a graficului.

Se observă că există o componentă de tip „pieptene Dirac" în domeniul frecvență produsă de interogarea la 5 minute făcută de S-CB către GW-I (marcată cu C în figură). Componenta din spațiul timp este o serie de impulsuri de lățime 1 cu perioada de 5 minute, ceea ce dă aspectul foarte pronunțat de „pieptene" în spațiul frecvență. Se observă că această componentă este aproape permanent vizibilă pe durata întregii zile monitorizate.

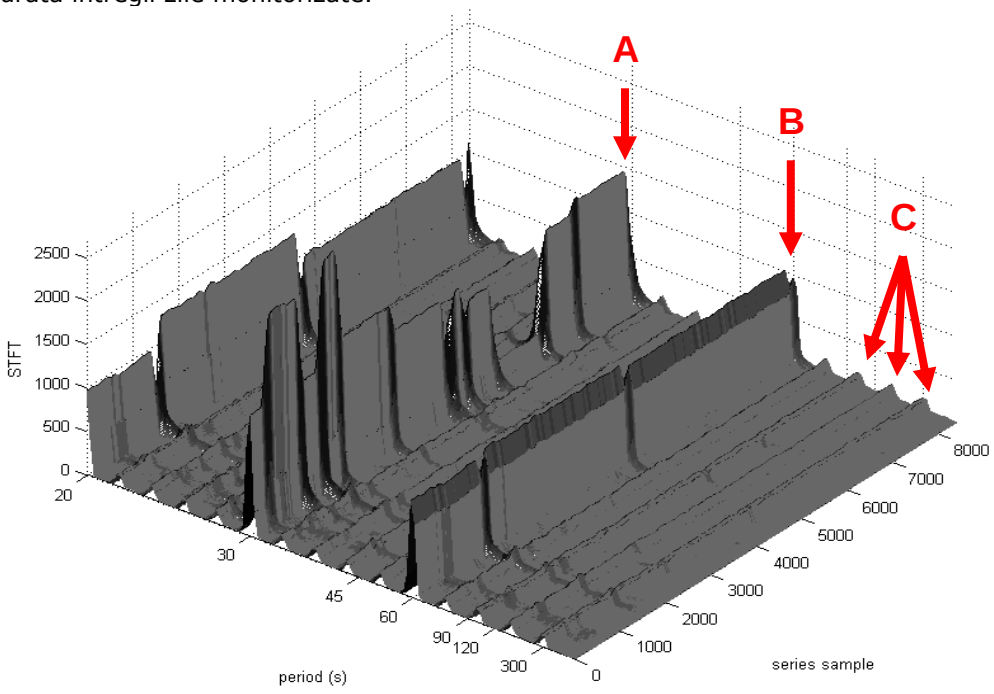


Figura 81. Transformata STFT pentru seria GW-I, index 19. A: Trafic apcupsd la 30 s; B: trafic apcupsd la 60 s; C: trafic mrtg la 5 min

Există de asemenea o componentă puternică de trafic cu perioada de 60 secunde (marcată cu B în figură, la componenta spectrală 43). Această componentă este datorată traficului de tip apcupsd (port 3551) generat de majoritatea serverelor *cu excepția* server-ului S-CB. Deoarece și acest tip de trafic are la origine aspect de tren de impulsuri în spațiul timp ne așteptăm să producă și el un efect de pieptene similar cu semnalul comentat anterior. Intr-adevăr, se regăsește un maxim relativ permanent în evoluția zilei atât la componenta spectrală 85 cât și la componenta spectrală 128. Dacă facem un calcul exact, 2560/60 = 42.(6) respectiv 2560/30 = 85.(3) unde 2560 este numărul de secunde acoperit de transformarea FFT de bază, deci perioada componentei fundamentale.

La prima vedere am considerat că lipsa componentei marcate cu A în figură este o anomalie. Pe respectiva componentă FFT se manifestă semnalul produs de clientul apcupsd al server-ului S-CB (vezi şi Figura 82), şi aşteptarea imediată este că lipsa componentei e cauzată de întreruperea funcţionării server-ului S-CB. In realitate, interpretarea corectă vine după ce ţinem seama că maximul B din Figura 81 este produs de multiple servere necorelate care toate execută polling la interval de 60 de secunde. Datorită evoluţiei aleatoare din punct de vedere a fazei între semnalele parţiale create de aceste servere (şi inclusiv prin compunere cu semnalul generat de S-CB la 30 secunde) există posibilitatea ca amplitudinea componentei spectrale 85 să nu mai aibă valori semnificative. Acelaşi lucru se aplică şi pentru componenta spectrală 128 care este vizibil variabilă în graficul produs de NEAR-GUI.

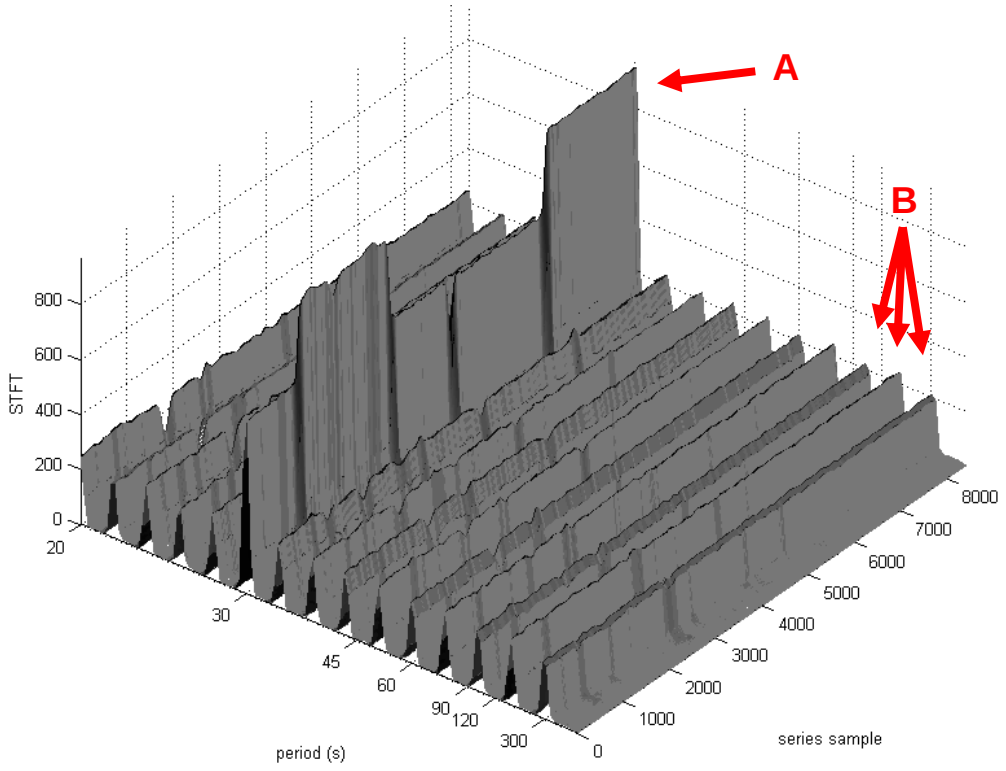


Figura 82. Transformata STFT pentru seria S-CB, index 3. A: trafic apcupsd la 30 s; B: trafic mrtg la 5 min

O întrebare legitimă este „de ce variază în timp prezentarea maximului A din Figura 82 dacă e produsă de trafic stabil tip client apcupsd al server-ului S-CB unde nu de manifestă fenomenul de compunere”. Din analiza detaliata cu tcpdump a traficului generat de clientul apcupsd am constatat că numărul de pachete care compun o secvenţă de dialog între S-CB şi GW-I este sau 15 sau 17, aparent din cauza modului în care GW-I face TCP-push prematur la ultima parte a răspunsului său. Nu avem o dovadă directă, dar cele două valori distincte constatate pentru maximul A ar putea să fie cauzate de acest fenomen.

Din analiza cazului prezentat mai sus se poate trage concluzia că descompunerea STFT poate produce informație utilă în cazul unor serii cu caracter puternic periodic dar că interpretarea datelor rezultate poate fi delicată și necesită atenție la artifactele de aliasing.

5.8.4. Variaţie a comportării periodice în reţeaua RD

Pentru a da o imagine completă asupra capabilităților analizei în timp-frecvență în cazul seriilor care prezintă (și) caracter periodic, adăugăm și alte exemple de situații reale.

In Figura 83 se regăseşte seria SYN care prezintă clar o activitate intensă în timpul programului de lucru (6:30 – 24:00) dar care nu poate fi desemnată ca structurată din spaţiul timp. Pe graficul STFT se observă însă o comportare periodică pe care o putem investiga similar cu cazul de mai sus.

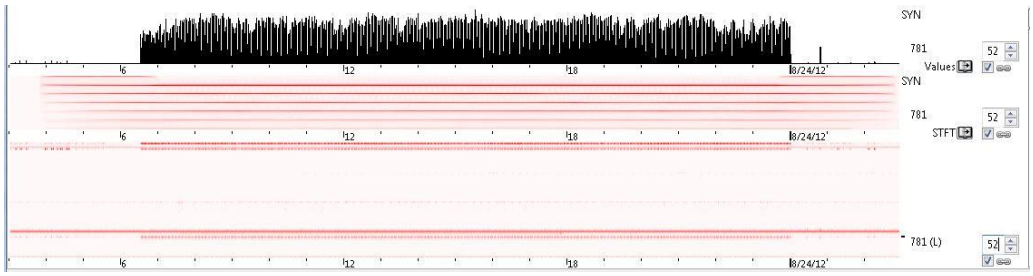


Figura 83. Activitate pe seria SYN la server de build periodic S-CB.

In Figura 84 și respectiv Figura 85 am aplicat aceleași metode de extracție și vizualizare pentru a ilustra mai bine situația menționată anterior în capitolul despre analiză și anume faptul că variația caracterului periodic poate fi un indicator de bună funcționare, în cazul în care este corelat cu alte informații despre sistemul monitorizat.

Din reprezentarea făcută în Figura 84 se observă imediat că seria SYN are o componentă continuă cu perioada de 2 minute (120 s) ceea ce corespunde foarte bine cu perioada buclei care verifică prezenţa unor noi modificări în baza de cod sursă folosind CVS. Captura a fost făcută în data de 23 August 2012 care a fost aparent o perioadă fără modificări de cod pentru că nu se constată ruperi ale structurii descompuse în frecvență pe axa evoluției temporale. O diagramă similară extrasă pentru ziua următoare (Figura 85) are o structură mai dezordonată deoarece în ziua respectivă au existat modificări care au necesitat cicluri de build și care au întrerupt periodicitatea de 2 minute.

Așadar în acest caz continuitatea caracterului periodic poate fi considerată o măsură a unei potențiale probleme (nu se efectuează build-uri) în cazul în care avem din alte surse informația că au existat modificări.

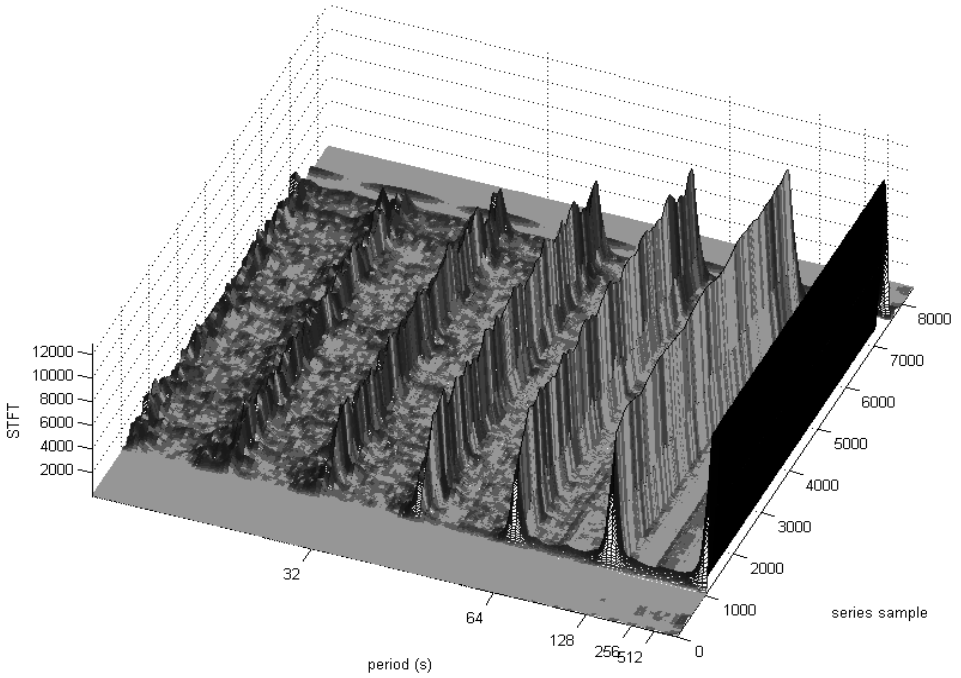


Figura 84. Transformata STFT pentru seria S-CB/52, 23-Aug-2012. Se observă efectul buclei de build nefolosite.

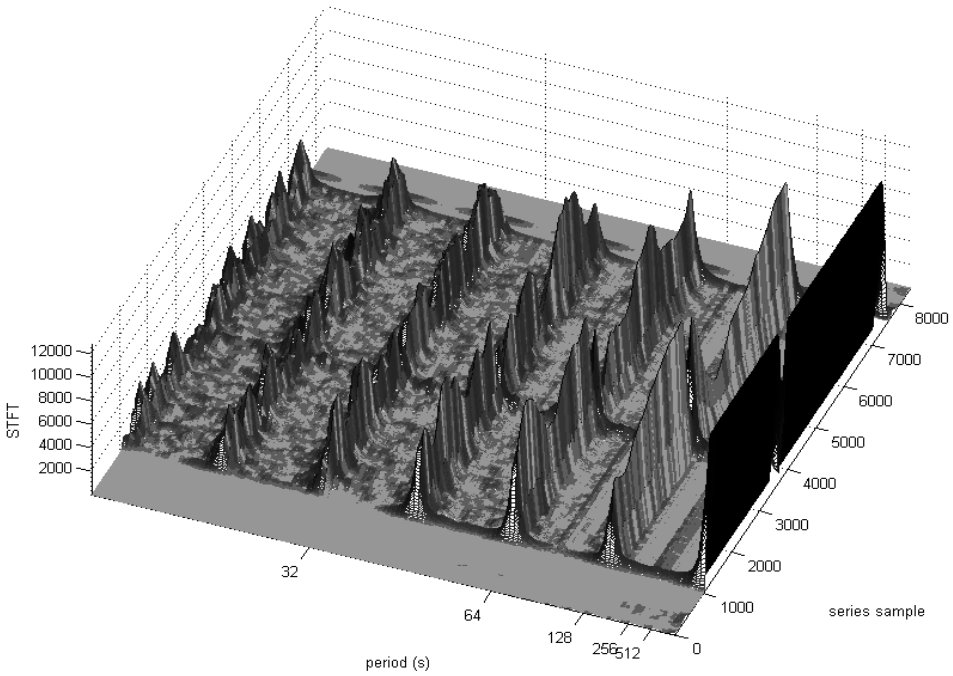


Figura 85. Transformata STFT pentru seria S-CB/52, 24-Aug-2012. Se observă efectul buclei de build active.

5.8.5. Evoluţie normală a protocolului NTP în reţeaua RD

Ultima diagramă pe care o prezentăm are mai mult rol de curiozitate dar este relevantă pentru că ilustrează cum procesarea timp-frecvență poate să identifice o comportare de protocol mai greu de evidențiat cu alte metode. In Figura 86 este reprezentată transformata STFT a seriei 35 colectată pe S-CB. Seria 35 este seria porturilor (123, 161, 514, 3551) dar din cauză că se urmăreşte varianta simetrică (sursa şi destinaţia sunt din aceeaşi clasă) traficul pe care îl conţine seria este cel mai probabil trafic NTP care pentru sistemul de operare Linux se face de pe port 123 pe port 123.

Descrierea protocolului NTP [86] presupune că nodul client (în cazul nostru S-CB) interoghează periodic nodul server (în cazul nostru GW-I) pentru a obține o referință de timp. Deoarece însăși tranzitul pachetului cerere și răspuns poate afecta referința de timp se face o estimare a duratei de tranzit și se aplică algoritmi de filtrare pentru a elimina valorile accidentale. In cursul acestor operații clientul NTP variază intervalul de polling în funcție de obiectivul urmărit (aliniere rapidă sau integrare bună). Acest interval de polling este exprimat în secunde și este o putere a lui 2, uzual între 64 și 1024. Un nod NTP în funcționare normală se plasează de obicei la 1024 s dar ocazional rata de poll se reduce pentru a compensa variații prea rapide în frecvența ceasului propriu.

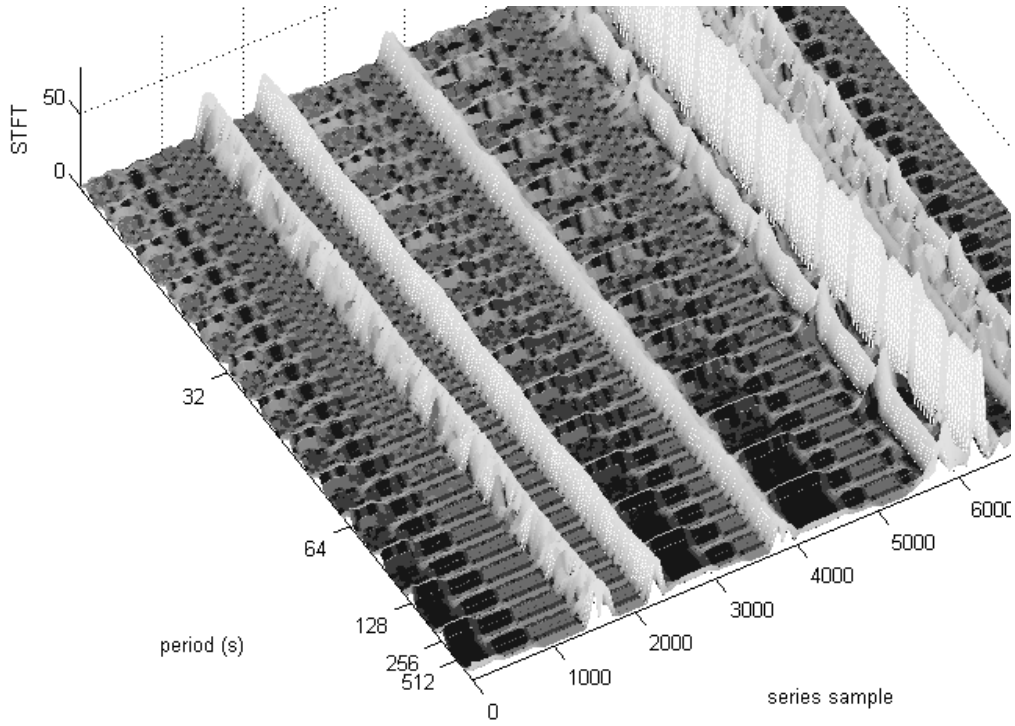


Figura 86. Transformata STFT pentru seria S-CB index 35 (NTP simetric).
Se observă evoluţia gradată şi repetată în timp a protocolului
NTP de la poll-interval 256 la poll-interval 1024

In Figura 86 se vede cum există intervale de timp în care perioada traficului NTP este clar 256, urmate de intervale în care perioada traficului este clar de 512 (între marcaj de timp 0 și aproximativ 900). Amplitudinea acestor caracteristici extrase este redusă și din această cauză intervalul cu perioada de 1024 nu mai este așa de bine vizibil dar există clar o zonă care poate fi atribuită unei funcționări cu poll-interval 1024 secunde. Secvența este absolut conformă cu comportarea teoretică a protocolului NTP.

Se observă de asemenea că această comportare de „auto-tuning” a protocolului NTP este repetată, probabil datorită faptului că frecvența oscilatorului propriu a fost perturbată (de exemplu undeva între marcajele de timp 1500 și 2500).

Evoluția în timp a diagramei pentru seria 35 ne invită la încă o ultimă constatare. Există pe această serie valori locale semnificativ mai mari decât cele considerate normale, valori care se comportă ca niște impulsuri și care produc crestele extinse peste tot spectrul de frecvențe. Din compoziția seriei ar trebui să fie pachete 123-123, 161-161, 514-514 sau 3551-3551. Din păcate nu există captura detaliată a acelei perioade din care să determinăm cauza exactă. Estimăm că ar putea fi vorba de o comportare neașteptată a implementării de NTP sau de utilizarea ocazională a portului de plecare 3551 pentru apel client apcupsd. Constatarea sugerează că un sistem real de detecție va trebui să folosească multiple surse de informație pentru a evita astfel de falsuri pozitive.

6. CONCLUZII ŞI CONTRIBUŢII PERSONALE

Obiectivul acestei teze este de a explora metode de îmbunătățire a sistemelor de detecție a intruziunilor în rețele ce calculatoare folosind tehnici de analiză a traficului privit ca întreg (spre deosebire de metoda de analiză pachet-cu-pachet).

In acest scop am pornit intuitiv de la evidența existenței unui conținut valoros de informație în seriile de valori care reprezintă traficul de rețea pe care am constatat-o din experiența personală câștigată prin urmărirea rezultatelor produse de unelte de monitorizare cum ar fi MRTG.

Am definit un set de principii de îmbunătăţire a ratei de detecţie a anomaliilor (p. 37) şi am căutat metode de aplicare practică a acestora. Pentru aceasta am definit o metodă de extracție a unor serii de valori din traficul brut capturat pe o interfață oarecare de rețea (p. 39). Am definit o tehnică de segregare a destinației traficului care este complementară sistemului bazat pe fluxuri existent deja în industrie, metodă bazată pe folosirea separării după domenii de autoritate (p. 39) şi a porturilor binecunoscute (p. 42). In același timp am subliniat importanța analizei multi-nivel a traficului (cu referire la nivelele ISO-OSI) pentru detecția unei game cât mai largi de anomalii (p. 41).

Am analizat în continuare categoriile de anomalii care se pot manifesta în seriile de date colectate și am evidențiat că în fapt analiza datelor colectate în diverse puncte din rețea poate conduce la detecția a mai mult decât atacuri de securitate și anume poate evidenția defecte de configurare, de operare sau probleme externe, chiar de natură hardware (p. 47).

In continuare am introdus și am evaluat metode de detecție a anomaliilor de tip impuls în seriile de date (p. 52). Deoarece elementul fundamental de detecţie este aplicarea unui prag am dicutat şi evaluat două tehnici de detecţie simplă cu prag, corelat cu specificul seriilor de date oe care intenţionăm să le procesăm (p. 55). Am descris şi analizat de asemenea şi două metode de detecție a impulsurilor care folosesc descompunerea wavelet: una bazată pe descompunere și cumulare de praguri (p. 65) și una bazată pe o variantă de filtrare cu descompunere și recompunere care folosește eliminarea componentei de aproximare (p. 67).

Deoarece seriile de date sunt în principal produse din procesarea volumelor de trafic de rețea am explorat în diferite puncte din lucrare impactul pe care îl are distribuția auto-similară asupra diverselor metode de detecție a anomaliilor impuls. Am constatat prin evaluare experimentală nivelul redus de auto-similaritate pe care îl prezintă seriile descompuse și am justificat acest fenomen relativ la specificul descompunerii pe care am executat-o (p. 26,52).

Am evaluat de asemenea aplicabilitatea diferitelor funcții wavelet curente pentru analiza semnalelor specifice de perturbație tip impuls și am determinat experimental parametrii care pot să producă rezultate mai bune dar am constatat în același timp că nu se poate obține o îmbunătățire semnificativă doar prin utilizarea funcțiilor wavelet uzuale, neadaptate (p. 79) .

Toate eforturile depuse sunt parte dintr-o dezvoltare mai largă care țintește spre construcția unui sistem (semi)automat de detecție a anomaliilor din rețele mici și mijlocii (p. 81). Din această cauză pe parcursul dezvoltării acestei teze am pus la punct o suită de instrumente de colectare și analiză care pot fi bază a unui astfel de viitor sistem. Am descris așadar particularitățile acestor unelte și modul în care

aplicarea lor pe seturi de date binecunoscute (DARPA) respectiv pe seturi de date private poate conduce la detecția unei game largi de anomalii.

Am descris detaliile de implementare pentru un agent de colectare şi extracţie care respectă principiile enunţate anterior (p. 90). Am descris de asemenea un instrument portabil de vizualizare a seriilor de date pe care l-am dezvoltat pentru explorarea confortabilă a datelor colectare de agent (p. 96). Am aplicat complet mecanismul de extracţie în mod offline pe întregul set de date DARPA'98 şi DARPA'99 (p. 98) dar am monitorizat şi reţele reale în timp real. Datele extrase au fost bază pentru exemplificarea unor situaţii tipice de detecţie a anomaliilor (p. 107)

Firește, până la implementarea completă a unui astfel de sistem mai trebuie depuse multe eforturi. Nu putem decât să descriem sumar în continuare câteva direcții posibile pe care l-am identificat în cursul dezvoltării de până acum.

In setul surselor de informații care pot fi relevante pentru comportarea anormală a unei rețele se pot include și mărimi independente de trafic dar care pot fi transformate în serii de valori în spațiul timp. Este vorba de mărimi cum ar fi încărcarea CPU, temperatura internă a nodurilor de rețea, numărul de accese la disc pe unitatea de timp pentru nodurile server. Pentru a putea folosi aceste informații în sistemul final e nevoie de agenți de colectare directă sau indirectă compatibili cu formatul oferit deja de NEAR-agent.

Pentru a putea detecta mai bine anomaliile este nevoie de continuarea eforturilor de a obține algoritmi potriviți. O direcție este îmbunătățirea capabilităților multirezoluție peste spațiul timp pe o gamă mai largă de scări temporale deoarece am constatat că există componente semnificative atât la scara de ordin minute cât și la scara de ordin zi sau săptămână (care actualmente este dincolo de spațiul analizat de uneltele pe care le-am dezvoltat). In acest scop va trebui dezvoltat inclusiv un mecanism de colectare a seriilor de date pe termen lung și eventual în formă deja scalată în timp, similar cu ce oferă biblioteca RRD/MRTG.

Am dovedit prin exemplele prezentate din capturile pe rețele reale că există un potențial semnificativ de detecție a anomaliilor prin corelația deciziilor peste multiple serii de date. In acest sens o direcție de perfecționare este îmbunătățirea algoritmului de detecție bazat pe corelație multirezoluție dar și construcția unor algoritmi de tip reguli configurabile sau alte soluții din categoria sistem expert care să poată să modeleze cu succes deciziile luate de un operator uman.

Pe direcția de transformare a sistemului într-o unealtă reală pentru utilizări imediate se pot lua măsuri de construcție a unor interfețe de prezentare mai prietenoase și a unor instrumente mai comode de gestionare a populației de agenți necesari achiziției de informație relevanta din rețea.

7. Bibliografie citată şi consultată

1. Florin Vancea, Codruţa Vancea - Network traffic flow analysis as a security tool, Analele Universităţii Oradea, Fascicola Electrotehnică, Secţiunea Ştiinţa Calculatoarelor şi Sisteme de Control, 24-26 Mai 2007, Oradea (EMES'07), Pag. 129-131,2007
2. Les Cottrell & Connie Logg, „Distributed Computing Environment Monitoring and User Expectations", SLAC-PUB-95-7008, Contributed to the International Conference on Computing In High Energy Physics '95 (CHEP95) Conference, Rio De Janiero, Brazil September 18-22, 1995
3. L. Todd Heberlein, Gihan V. Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood, David Wolber, „A Network Security Monitor", 1990 IEEE Computer Society Symposium on Research in Security and Privacy, May 7, 1990
4. Richard Bejtlich, „The Tao of Network Security Monitoring: Beyond Intrusion Detection", Addison-Wesley; July 2004, ISBN 0321246772
5. Jill Huntington-Lee, Kornel Terplan, Jeff Gibson, „HP Openview: A Manager's Guide", McGraw-Hill, Inc. New York, NY, USA, 1997, ISBN:0070313822
6. „Introduction to Cisco IOS® Flexible NetFlow", CISCO white paper, online, PDF, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/ps6965/prod_white_paper0900aecd804be1cc.pdf
7. „Introduction to Cisco IOS NetFlow - A Technical Overview", CISCO technical overview, online, PDF, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.pdf
8. John Gerber, „Three Open Source IDS/IPS Engines", in „Security Advancements at the Monastery 2010", online, blog, http://blog.securitymonks.com/2010/08/26/three-little-idsips-engines-build-their-open-source-solutions/
9. Les Cottrell, „Network Monitoring Tools", online, HTML, http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
10. George Varghese, „Network Algorithmics", Elsevier, 2005, ISBN: 0-12-088477-1, c.16-17
11. „Endace DAG 9.2X2 Datasheet", Endace, online, PDF, http://www.endace.com/assets/files/resources/END_Datasheet_DAG9.2X2_4.0.pdf
12. Luca Deri, „Improving Passive Packet Capture: Beyond Device Polling", Proceedings of SANE 2004
13. Luca Deri, Francesco Fusco, „Exploiting commodity multi-core systems for network traffic analysis", online, PDF, http://luca.ntop.org/MulticorePacketCapture.pdf
14. Lothar Braun, Alexander Didebulidze, Nils Kammenhuber, Georg Carle, „Comparing and Improving Current Packet Capturing Solutions based on Commodity Hardware", IMC'2010, November 1–3, 2010, Melbourne, Australia, ACM 978-1-4503-0057-5/10/11, pp 206
15. Florin Vancea, Codruţa Vancea – „Practical Security Issues for a Real Case Application", International Journal of Computers, Communications & Control,Vol. III,15-17 Mai 2008, Oradea (ICCCC 2008), Pag. 516-520,2008.

16. Martin Roesch, „Snort - Lightweight Intrusion Detection for Networks”, Proceedings of LISA '99, p.229-238
17. V. Paxson, „BRO: A System for Detecting Network Intruders in Real Time”, Proceedings of the 7th USENIX Security Symposium, 1998
18. Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, Wenke Lee, „BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation”, Proceedings of 16th USENIX Security Symposium, 2007, pp.167-182
19. RFC1271 - Remote Network Monitoring Management Information Base, online, http://tools.ietf.org/html/rfc1271
20. RFC2021 - Remote Network Monitoring Management Information Base Version 2 using SMIv2, online, http://tools.ietf.org/html/rfc2021
21. RFC3954 - Cisco Systems NetFlow Services Export Version 9, online, http://tools.ietf.org/html/rfc3954
22. RFC5101 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information, online, http://tools.ietf.org/html/rfc5101
23. RFC5102 - Information Model for IP Flow Information Export, online, http://tools.ietf.org/html/rfc5102
24. RFC5103 - Bidirectional Flow Export Using IP Flow Information Export (IPFIX), online, http://tools.ietf.org/html/rfc5103
25. RFC3176 - InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, online, http://tools.ietf.org/html/rfc3176
26. W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” IEEE/ACM Trans. Networking, pp. 1–15, 1994
27. Stilian Stoev, Murad S. Taqqu, Cheolwoo Park, J. S. Marron, „On the wavelet spectrum diagnostic for Hurst parameter estimation in the analysis of Internet traffic”, Computer Networks, vol.48, 2005, pp423-445
28. J.M. Peha, „Protocols Can Make Traffic Appear Self-Similar”, Department of Engineering and Public Policy. Paper 47., online, http://repository.cmu.edu/epp/47
29. J. Brutlag, “Aberrant behavior detection in time series for network monitoring”, in Proceedings of the USENIX Fourteenth System Administration Conference LISA XIV, New Orleans, LA, December 2000
30. Rudolf H. Riedi, Vinay J. Ribeiro, Matthew S. Crouse, and Richard G. Baraniuk, „Network Traffic Modeling using a Multifractal Wavelet Model”, Proceedings European Congress of Mathematics, Barcelona 2000
31. Paul Barford, Jeffery Kline, David Plonka and Amos Ron, „A Signal Analysis of Network Traffic Anomalies”, in Proceedings of ACM SIGCOMM Internet Measurement Workshop 2002
32. A. Isar, I. Nafornita. „Reprezentări timp-frecvenţă”, Editura "Politehnica", Timisoara, Romania,1998.
33. S. G. Mallat, „A Wavelet Tour of Signal Processing”,. San Diego / London: Academic Press, 2nd ed., 1999.
34. David Rincón Rivera, „Contributions to the Wavelet-based Characterization of Network Traffic”, PhD thesis, Universitat Politècnica de Catalunya, 2007
35. J.Gao, G. Hu,X. Yao, and R. K. C. Chang, “Anomaly detection of network traffic based on wavelet packet,” in Proceedings of the Asia-Pacific Conference on Communications (APCC ’06), pp. 1–5, Busan, Korea, August 2006

36. Wei Lu, Ali A. Ghorbani, „Network Anomaly Detection Based on Wavelet Analysis", EURASIP Journal on Advances in Signal Processing, Volume 2009, Article ID 837601, 16 pages, doi:10.1155/2009/837601
37. I. Adam, M. Oltean and M. Bora, "A New Quasi Shift Invariant Non-Redundant Complex Wavelet Transform", Scientific Bulletin of the "POLITEHNICA" University of Timisoara, number dedicated to the Symposium on Electronics and Telecommunications ETC 2006, Seventh Edition, Tom 51 (65), Fascicola 2, 2006 ISSN 1583-3380, pp.14-18, Timişoara, September 21-23 2006.
38. Marius Sălăgean, Ioana Firoiu, „Anomaly Detection of Network Traffic Based on Analytical Discrete Wavelet Transform", Proceedings of COMM2010, Bucharest, 2010
39. Chin-Tser Huang, Sachin Thareja, and Yong-June Shin, „Wavelet-based Real Time Detection of Network Traffic Anomalies", International Journal of Network Security, Vol.6, No.3, PP.309–320, May 2008
40. Yuliya Kopylova, Duncan A. Buell, Chin-Tser Huang and Jeff Janies, „Mutual Information Applied to Anomaly Detection", Journal of Communications and Networks, Vol. 10, No. 1, 2008, pp. 89-97
41. Gaia Maselli, Luca Deri, Stefano Suin, „Design and Implementation of an Anomaly Detection System: an Empirical Approach", in Proceedings of Terena Networking Conference, Zagreb, Croatia, 2003
42. Yogendra Kumar JAIN, Sandip S. PATIL, „Design of Hybrid Network Anomalies Detection System (H-NADS) Using IP Gray Space Analysis", Informatica Economică vol. 13, no. 2/2009, pp.110-119
43. Li Bo, David J. Parish, J.M. Sandford, P. J. Sandford, „Using TCP Packet Size Distributions for Application Detection", PGNET2006, Liverpool John Moores University, 2006
44. Marcell Perényi, Trang Dinh Dang, András Gefferth, Sándor Molnár, „Identification and Analysis of Peer-to-Peer Traffic", Journal of Communications, VOL. 1, NO. 7, November/December 2006, ISSN 1796-2021
45. Rohit Dhamankar, Rob King, „Protocol Identification via Statistical Analysis (PISA)", as Black Hat 2007 presentation, online, http://www.blackhat.com/presentations/bh-usa-07/Dhamankar_and_King/Whitepaper/bh-usa-07-dhamankar_and_king-WP.pdf
46. Cihangir Beşiktaş and Haci Ali Mantar, „Real-time Traffic Classification Based on Cosine Similarity Using Sub-Application Vectors", in Proceedings of Traffic Monitoring and Analysis 4th International Workshop, TMA 2012, Vienna, Austria
47. Team Cymru, „The Darknet Project", online, http://www.team-cymru.org/Services/darknets.html
48. Florin Vancea, Codruţa Vancea, "NEAR — Network extractor of anomaly records or traffic split-counting for anomaly detection", Proceedings of EUROCON 2013, ISBN 978-1-4673-2230-0, p. 60-64, 2013, DOI: 10.1109/EUROCON.2013.6624966
49. A.S. Tanenbaum, „Rețele de calculatoare", ed. 4, Byblos, 2003, ISBN 973-0-03000-6
50. M. Naforniță, „Arhitectura retelelor de calculatoare", Politehnica Timisoara Publishing House, 2007, 185 pag., ISBN 978-973-625-574-8

51. CAIDA – Cooperative Association for Internet Data Analysis, „The UCSD Network Telescope", online, http://www.caida.org/projects/network_telescope/
52. D. Moore, C. Shannon, G. Voelker, and S. Savage, „Network Telescopes: Technical Report", Tech. rep., Cooperative Association for Internet Data Analysis (CAIDA), Jul 2004
53. E. Cooke, M. Bailey, D. Watson, F. Jahanian, J. Nazario, The Internet Motion Sensor: A distributed global scoped Internet threat monitoring system, Technical Report CSE-TR-491-04, University of Michigan, Electrical Engineering and Computer Science, July 2004
54. CISCO white paper „Strategic Directions Moving the Decimal Point: An Introduction to 10 Gigabit Ethernet", online, PDF, http://www.cisco.com/warp/public/cc/techno/lnty/etty/ggetty/tech/10gig_wp.pdf
55. Friesen, G. M., Jannett, T. C., Jadallah, M. A., Yates, S. L., Quint, S. R., & Nagle, H. T. (1990). „A comparison of the noise sensitivity of nine QRS detection algorithms". *Biomedical Engineering, IEEE Transactions on*, *37*(1), 85-98.
56. Sudipta Mukhopadhyay, G. C. Ray, „A New Interpretation of Nonlinear Energy Operator and Its Efficacy in Spike Detection", IEEE Transactions on Biomedical Engineering, vol. 45, no. 2, February 1998
57. Kohler, B.-U., Hennig, C., Orglmeister, R., „The principles of software QRS detection", Engineering in Medicine and Biology Magazine, IEEE (Volume:21 , Issue: 1), Jan.-Feb. 2002, ISSN 0739-5175, pp. 42 – 57
58. J. F. Kaiser, "On Teager's algorithm and its generalization to continuous signals," *Proc. 4th IEEE Digital Signal ProcessingWorkshop*, Mohonk (New Paltz), NY, Sept. 1990.
59. I. Obeid, "A Wireless Multichannel Neural Recording Platform for Real-Time Brain Machine Interfaces", *PhD thesis,*Duke University 2004
60. F. Vancea, „On Performance of Simple Detection of Pulse-Shaped Anomalies in Data Series from NEAR Network Data Collection Tool", Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara, Tom 57(71), Fascicola 1-2, 2012
61. S. Uhlig and O. Bonaventure. „Understanding the Long-Term Self-Similarity of Internet Traffic", Proc. of QOFIS2001, Coimbra, Portugal, September 2001. Springer-Verlag LNCS2156, pages 286-298
62. Sergei Katsev, Ivan L'Heureux, „Are Hurst exponents estimated from short or irregular time series meaningful?" Computers & Geosciences 29, Elsevier 2003, 1085–1089
63. C. Narduzzi, P. Pegoraro, and S. Uhlig,. „Revisiting the multiscaling hypothesis at medium timescales",.Proc. of the 20thGRETSI symposium on signal and image processing, Louvain-la-neuve, Belgium, September 2005
64. Richard Clegg, „A Practical Guide to Measuring the Hurst Parameter", International Journal of Simulation: Systems, Science & Technology 7(2) pp 3-14 2006
65. Richard G. Clegg, Raul Landa, Miguel Rio, „Criticisms of modelling packet traffic using long-range dependence" (extended version), Journal of Computer and System Sciences Volume 77, Issue 5, September 2011, Pages 861–868
66. F. Vancea, "Intrusion detection in NEAR system by anti-denoising traffic data series using discrete wavelet transform", submitted to AECE - Advances

in Electrical and Computer Engineering (www.aece.ro), ISSN 1582-7445, e-ISSN 1844-7600, doi: 10.4316/aece
67. Boashash, B.: „Time Frequency Analysis”, Elsevier Science, 2003, Publication Date: October 30, 2003 | ISBN-10: 0080443354 | ISBN-13: 978-0080443355
68. S. Mallat, „A Wavelet Tour of Signal Processing”, Third Edition: The Sparse Way, Academic Press 2008, ISBN: 978-0123743701
69. Netresec, Publicly available PCAP files, online, http://www.netresec.com/?page=PcapFiles
70. Evil Fingers, PCAP repository, online, https://www.evilfingers.com/repository/pcaps.php
71. Robert K.Cunningham, Richard P. Lippmann, David J. Fried, Simson L. Garfinkel, Isaac Graf, Kris R. Kendall, Seth E. Webster, Dan Wyschogrod, and Marc A. Zissman. „Evaluating intrusion detection systems without attacking your friends: The 1998 DARPA intrusion detection evaluation”. MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 1999.
72. Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. "The 1999 DARPA off-line intrusion detection evaluation." *Computer networks* 34, no. 4 (2000): 579-595.
73. John McHugh,. "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory." ACM transactions on Information and system Security 3, no. 4 (2000): 262-294.
74. Thomas, Ciza; Sharma, Vishwas; Balakrishnan, N – „Usefulness of DARPA dataset for intrusion detection system evaluation”, in "Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008". Edited by Dasarathy, Belur V. Proceedings of the SPIE, Volume 6973, article id. 69730G, 8 pp. (2008)
75. G. Wilkinson, „Identification of Hostile TCP Traffic using Support Vector Machines”, Lincoln College, Oxford University Computing Laboratory, 2009
76. B. Sangster, T.J. O'Connor, T. Cook, R. Fanelli, E. Dean, J. Adams, C. Morrell, and G. Conti; "Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets;" USENIX Security's Workshop on Cyber Security Experimentation and Test (CSET); August 2009
77. Kendall, Kristopher. "A database of computer attacks for the evaluation of intrusion detection systems." PhD diss., Massachusetts Institute of Technology, 1999
78. MIT Lincoln Laboratory website, „1999 DARPA Intrusion Detection Evaluation”, online, http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/docs/
79. Haines, Joshua Wills, Richard P. Lippmann, David J. Fried, M. A. Zissman, and E. Tran. „1999 DARPA intrusion detection evaluation: Design and procedures”. No. TR-1062. Massachusetts Inst Of Tech Lexington Lincoln Lab, 2001.
80. Haines, Joshua W., Lee M. Rossey, Richard P. Lippmann, and Robert K. Cunningham. "Extending the darpa off-line intrusion detection evaluations." In DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings, vol. 1, pp. 35-45. IEEE, 2001.
81. Rossey, Lee M., Robert K. Cunningham, David J. Fried, Jesse C. Rabek, Richard P. Lippmann, Joshua W. Haines, and Marc A. Zissman. "Lariat:

Lincoln adaptable real-time information assurance testbed." In Aerospace Conference Proceedings, 2002. IEEE, vol. 6, pp. 6-2671. IEEE, 2002.
82. APC UPS Daemon – apcupsd site, online, http://www.apcupsd.com/
83. Tobi Oetiker, „Multi Router Traffic Grapher", site, online, http://oss.oetiker.ch/mrtg/
84. „SWT: The Standard Widget Toolkit" – Eclipse SWT site, online, http://www.eclipse.org/swt
85. Gordon Lyon et. al., „Nmap Security Scanner" – NMAP site, online, http://nmap.org
86. Mills, D., Jim Martin, Jack Burbank, and William Kasch. "RFC 5905: Network Time Protocol version 4: Protocol and algorithms specification." Internet Engineering Task Force (2010).