On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches

Marek Ostaszewski

Advisor: prof. Pascal Bouvry

Faculty of Science, Technology and Communication
University of Luxembourg

09/02/2010 Ph.D. Thesis defense

Outline

Contents

Big, fast and dangerous Internet

- Estimated number of the users of the Internet is 25% of population of Earth [IWS'09]
- Connection speed grows by 50% per year [Nielsen'09]
- Large-scale attacks in the Internet
  - Benefit from resources of unaware Internet users
  - The most devastating attacks against enterprise networks [CSI/FBI'04, CSOS&R'09]

Information security vs Denial of Service (DoS)

Information Security

Integrity + Confidentiality + Availability

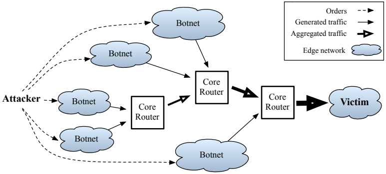
Distributed Denial of Service (DDoS)

- Blocks its target with artificially generated traffic
- Exploits resources of unaware/idle users (bots)
- Generates traffic difficult to process and classify

DDoS in action

"Democracy" of DDoS attack

- Everybody can have a botnet
  - ... and a really big one! (180,000 bots[CSOS&R'09])
- The Internet is a DDoS-friendly environment



Security mechanisms vs DDoS

Intrusion Detection Systems (IDSs)

- Monitor a computer system or network
- Search for attacks

Classification of IDSs

- **Location**
  - **Network**
  - Host
- Detection
  - Signature
  - Anomaly

Characteristics: Location/Network

Monitor the entry point of the protected network

- + Broad scope
- − Limited information

Introduction Scope

Security mechanisms vs DDoS

Intrusion Detection Systems (IDSs)

- Monitor a computer system or network
- Search for attacks

Classification of IDSs

- **Location**
  - Network
  - **Host**
- Detection
  - Signature
  - Anomaly

Characteristics: Location/Host

Monitor the computer system of the end-user

- + Detailed information
- − Limited scope

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 7 | 42

Security mechanisms vs DDoS

Intrusion Detection Systems (IDSs)

- Monitor a computer system or network
- Search for attacks

Classification of IDSs

- Location
  - Network
  - Host
- **Detection**
  - **Signature**
  - Anomaly

Characteristics: Detection/Signature

Rely on the pre-constructed database of attack signatures

- + Precise
- − Reactive

Security mechanisms vs DDoS

Intrusion Detection Systems (IDSs)

- Monitor a computer system or network
- Search for attacks

Classification of IDSs

- Location
  - Network
  - Host
- **Detection**
  - Signature
  - **Anomaly**

Characteristics: Detection/Anomaly

Rely on the pre-constructed model of normal behavior

- + Proactive
- − Imprecise

Security mechanisms vs DDoS

Intrusion Detection Systems (IDSs)

- Monitor a computer system or network
- Search for attacks

Classification of IDSs

- Location
  - **Network**
  - Host
- Detection
  - Signature
  - **Anomaly**

IDS considered in DDoS case

- Network-based
- Anomaly-based

Contents

What is so difficult about DDoS?

Volume

Sizes of botnets reach 100 000 bots

- Analysis of detailed information leads to packet drop
- IDS choke

Variability

Instances of the attack differ one from another

- The sources, the path and the targeted resources
- Impossible to construct a signature

Similarity

DDoS resembles regular traffic in structure and behavior

- Flash event/crowd resembles stateful DDoS
- Typical anomaly-based IDS go blind

Problem analysis | Features of DDoS

What is so difficult about DDoS?

Volume

Sizes of botnets reach 100 000 bots

- Analysis of detailed information leads to packet drop
- IDS choke

Variability

Instances of the attack differ one from another

- The sources, the path and the targeted resources
- Impossible to construct a signature

Similarity

DDoS resembles regular traffic in structure and behavior

- Flash event/crowd resembles stateful DDoS
- Typical anomaly-based IDS go blind

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 9 | 42

Problem analysis | Features of DDoS

What is so difficult about DDoS?

Volume

Sizes of botnets reach 100 000 bots

- Analysis of detailed information leads to packet drop
- IDS choke

Variability

Instances of the attack differ one from another

- The sources, the path and the targeted resources
- Impossible to construct a signature

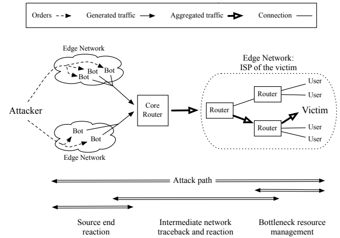
Similarity

DDoS resembles regular traffic in structure and behavior

- Flash event/crowd resembles stateful DDoS
- Typical anomaly-based IDS go blind

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 9 | 42

DDoS countermeasures



Problem definition

Representation of network traffic data

- Reduction of the network traffic volume
- Emphasis of inherent DDoS properties

Optimization of classification performance

- Minimization of the number of misclassified data samples
- Adaptation to changes in the traffic

State of the art

Representation

Jin'03	IP Time-to-Live clustering
Peng'04	Cumulative Sum analysis of IP Source address
Karas.'07	K-means clustering of botnet communications

Classification

Feinst.'03	IP Source address distribution
Chen'05	Frequency of inter-arrivals of TCP connections
Carl'06	Wavelet transform of traffic statistics
Xie'06	Hidden Markov Model for browsing behavior

Representation of the traffic using clustering

Goal

- The set of packets S^p is represented as a set of clusters S^c
- Number of clusters is always lower than an upper bound U

Data stream

- A sequence of data items $x_1, \ldots, x_i, \ldots, x_n$
- All the items are read only once
- Properties
  - One-pass requirement
  - Concept drift

Problem analysis Problem statement

Representation of the traffic using clustering

Goal

- The set of packets S^p is represented as a set of clusters S^c
- Number of clusters is always lower than an upper bound U

Data stream

- A sequence of data items $x_1, \ldots, x_i, \ldots, x_n$
- All the items are read only once
- Properties
  - One-pass requirement
  - Concept drift

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 13 | 42

Representation of the traffic using clustering

Goal

- The set of packets S^p is represented as a set of clusters S^c
- Number of clusters is always lower than an upper bound U

Data stream

- A sequence of data items $x_1, \ldots, x_i, \ldots, x_n$
- All the items are read only once
- Properties
  - One-pass requirement
  - Concept drift

Classification of the network traffic

Goal

- Classification of the clusters as *Attack* and *Normal*
- Minimization of incorrectly classified data samples

Binary classification [Fawcett'03]

- Assigns classified data sample to one of two classes
- Possible results are TP, TN (+) and FP, FN (−)

Performance metrics

- Sensitivity $= \dfrac{\text{TP}}{\text{TP+FN}}$
- Specificity $= \dfrac{\text{TN}}{\text{TN+FP}}$

Classification of the network traffic

Goal

- Classification of the clusters as *Attack* and *Normal*
- Minimization of incorrectly classified data samples

Binary classification [Fawcett'03]

- Assigns classified data sample to one of two classes
- Possible results are TP, TN (+) and FP, FN (−)

Performance metrics

- Sensitivity $= \dfrac{\text{TP}}{\text{TP+FN}}$
- Specificity $= \dfrac{\text{TN}}{\text{TN+FP}}$

Classification of the network traffic

Goal

- Classification of the clusters as *Attack* and *Normal*
- Minimization of incorrectly classified data samples

Binary classification [Fawcett'03]

- Assigns classified data sample to one of two classes
- Possible results are TP, TN (+) and FP, FN (−)

Performance metrics

- Sensitivity $= \dfrac{\text{TP}}{\text{TP+FN}}$
- Specificity $= \dfrac{\text{TN}}{\text{TN+FP}}$

Classification of the network traffic

Goal

- Classification of the clusters as *Attack* and *Normal*
- Minimization of incorrectly classified data samples

Binary classification [Fawcett'03]

- Assigns classified data sample to one of two classes
- Possible results are TP, TN (+) and FP, FN (−)

Performance metrics

- **Sensitivity** $= \dfrac{\text{TP}}{\text{TP+FN}}$
- Specificity $= \dfrac{\text{TN}}{\text{TN+FP}}$

Classification of the network traffic

Goal

- Classification of the clusters as *Attack* and *Normal*
- Minimization of incorrectly classified data samples

Binary classification [Fawcett'03]

- Assigns classified data sample to one of two classes
- Possible results are TP, TN (+) and FP, FN (−)

Performance metrics

- Sensitivity $= \dfrac{\text{TP}}{\text{TP+FN}}$
- **Specificity** $= \dfrac{\text{TN}}{\text{TN+FP}}$

Search for classification functions

Classification function [Ferreira'06]

Let x be an input data sample, then we define a function

$$\text{classify}(x) = \begin{cases} \textit{Attack}\ (1), & \textit{if}\ g(x) \geq C \\ \textit{Normal}\ (0), & \textit{otherwise} \end{cases}$$

The search for classification function

- Function $g(x)$ should maximize sensitivity and specificity
- The problem of finding $g(x)$ can be
  - Single-objective when two metrics are combined
  - Multi-objective when they are separate objectives

Search for classification functions

Classification function [Ferreira'06]

Let x be an input data sample, then we define a function

$$\text{classify}(x) = \begin{cases} \textit{Attack}\ (1), & \textit{if}\ g(x) \geq C \\ \textit{Normal}\ (0), & \textit{otherwise} \end{cases}$$

The search for classification function

- Function $g(x)$ should maximize sensitivity and specificity
- The problem of finding $g(x)$ can be
  - Single-objective when two metrics are combined
  - Multi-objective when they are separate objectives

Search for classification functions

Classification function [Ferreira'06]

Let x be an input data sample, then we define a function

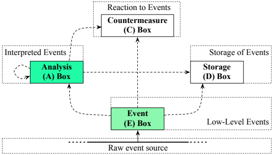
$$\text{classify}(x) = \begin{cases} \mathit{Attack}\ (1), & \mathit{if}\ g(x) \geq C \\ \mathit{Normal}\ (0), & \mathit{otherwise} \end{cases}$$

The search for classification function

- Function $g(x)$ should maximize sensitivity and specificity
- The problem of finding $g(x)$ can be
  - Single-objective when two metrics are combined
  - Multi-objective when they are separate objectives

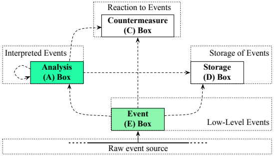
Contributions

- **Dynamic clustering** (representation)
  - Application and adaptation of Idiotypic Networks model
- Metaheuristic search (classification optimization)
  - Single and multi-objective Gene Expression Programming
- Integration and evaluation
  - Common Intrusion Detection Framework (CIDF)[Ptacek'98]



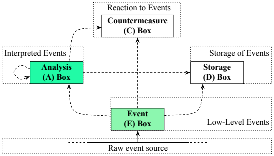
Contributions

- Dynamic clustering (representation)
  - Application and adaptation of Idiotypic Networks model
- **Metaheuristic search** (classification optimization)
  - Single and multi-objective Gene Expression Programming
- Integration and evaluation
  - Common Intrusion Detection Framework (CIDF)[Ptacek'98]



Contributions

- Dynamic clustering (representation)
  - Application and adaptation of Idiotypic Networks model
- Metaheuristic search (classification optimization)
  - Single and multi-objective Gene Expression Programming
- **Integration and evaluation**
  - Common Intrusion Detection Framework (CIDF)[Ptacek'98]



Contents

The Idiotypic Networks (IN) Paradigm

Artificial Immune Systems

- Paradigms inspired by a human immune system (HIS)
- Representation in HIS
  - Efficient compression: 10^8 antibodies - 10^{16} threats
  - High recognition ratio and adaptiveness are preserved

Idiotypic Networks theory

- Introduced by Niels Jerne (1974)
- Explains mechanisms of representation
- Inspiration to many IN models

The Idiotypic Networks (IN) Paradigm

Artificial Immune Systems

- Paradigms inspired by a human immune system (HIS)
- Representation in HIS
  - Efficient compression: 10^8 antibodies - 10^{16} threats
  - High recognition ratio and adaptiveness are preserved

Idiotypic Networks theory

- Introduced by Niels Jerne (1974)
- Explains mechanisms of representation
- Inspiration to many IN models

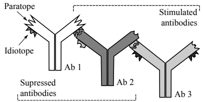
Idiotypic Networks (IN): theory and practice

IN theory in detail

- Antibodies (Ab) interact with other antibodies
- Directly - suppression
- Indirectly - stimulation

IN theory in practice [Mohr et al.'04]

- Ab = Data
- Interaction = Distance
- Suppression = Aggregation
- Stimulation = Similarity



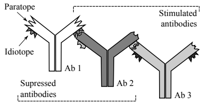
Idiotypic Networks (IN): theory and practice

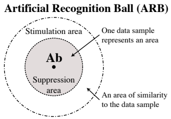
IN theory in detail

- Antibodies (Ab) interact with other antibodies
- Directly - suppression
- Indirectly - stimulation

IN theory in practice [Mohr et al.'04]

- Ab = Data
- Interaction = Distance
- Suppression = Aggregation
- Stimulation = Similarity





Proposed solution | Representation of the network traffic

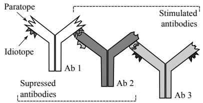
Idiotypic Networks (IN): theory and practice

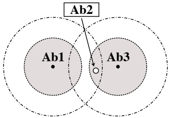
IN theory in detail

- Antibodies (Ab) interact with other antibodies
- Directly - suppression
- Indirectly - stimulation

IN theory in practice [Mohr et al.'04]

- Ab = Data
- Interaction = Distance
- Suppression = Aggregation
- Stimulation = Similarity



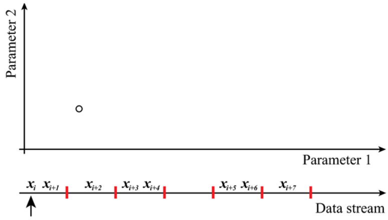


On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 19 | 42

IN model for data clustering and compression

Dynamics of the clustering process

lifetime = stimulation − decay

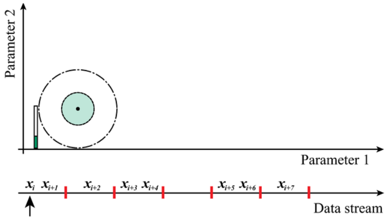


Proposed solution | Representation of the network traffic

IN model for data clustering and compression

Dynamics of the clustering process

lifetime = stimulation − decay

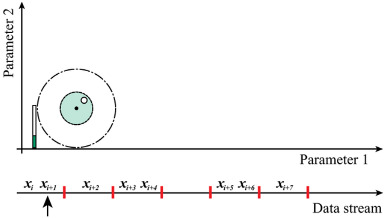


On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 20 | 42

IN model for data clustering and compression

Dynamics of the clustering process

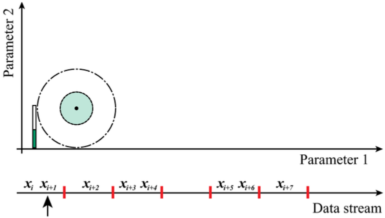
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

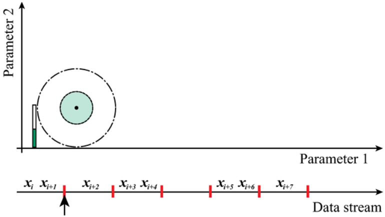
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

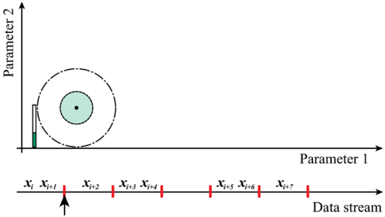
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

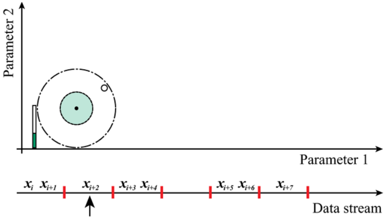
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

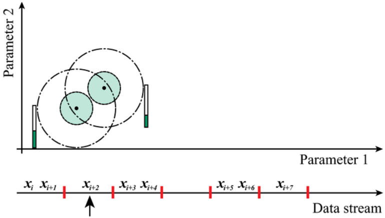
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

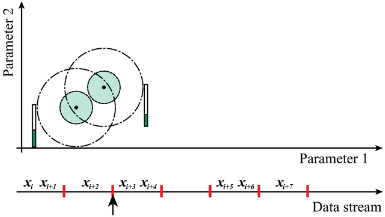
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

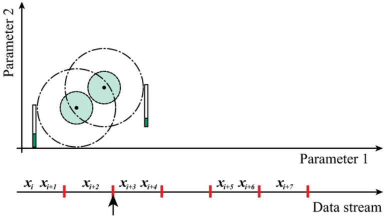
lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

lifetime = stimulation − decay



IN model for data clustering and compression

Dynamics of the clustering process

lifetime = stimulation − decay

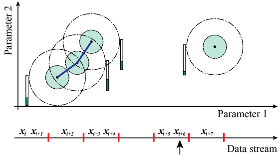
IN model for data clustering and compression

Dynamics of the clustering process

lifetime = stimulation − decay

Clustering

- **link** - connects neighbor ARBs
- **cluster** - linked ARBs



Representation of the network traffic - summary

ARB

- Aggregates data
- Represents of repetitive and intensive traffic
- Allows to preserve the upper bound U

Cluster of ARBs

- Groups ARBs
- Adapts to the changes in the data stream

Representation of the network traffic - summary

ARB

- Aggregates data
- Represents of repetitive and intensive traffic
- Allows to preserve the upper bound U

Cluster of ARBs

- Groups ARBs
- Adapts to the changes in the data stream

Introduction to Evolutionary Algorithms

Evolutionary Algorithms (EAs)

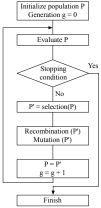
- Problem solving techniques
- Inspired by adaptation and evolution in the nature

Elements

- Chromosome
- Individual
- Population

Variants

- Genetic Algorithms
- Genetic Programming



Introduction to Evolutionary Algorithms

Evolutionary Algorithms (EAs)

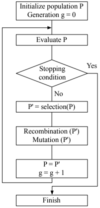
- Problem solving techniques
- Inspired by adaptation and evolution in the nature

Elements

- Chromosome
- Individual
- Population

Variants

- Genetic Algorithms
- Genetic Programming



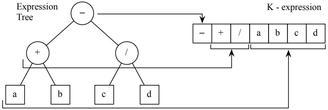
Gene Expression Programming (GEP) [Ferreira'06]

GEP = GA + GP

- Upper-bounded solution size
- Intensive exploration of the search space

Dual role of an individual

- Linear (k-expression): modifications
- Program tree (expression tree): evaluation



The individual and fitness calculation

The individual

- Program tree used as $g(x)$
- Individual is a classifier

Classification function

$$\text{classify}(x) = \begin{cases} 1, \ if\ \mathbf{g(x)} \geq C \\ 0, \ otherwise \end{cases}$$

Single-objective fitness (maximization)

$$so\ fitness = \text{sensitivity} * \text{specificity}$$

The individual and fitness calculation

The individual

- Program tree used as $g(x)$
- Individual is a classifier

Classification function

$$\text{classify}(x) = \begin{cases} 1, \textit{ if } \mathbf{g(x)} \geq C \\ 0, \textit{ otherwise} \end{cases}$$

Single-objective fitness (maximization)

$$\textit{so fitness} = \text{sensitivity} * \text{specificity}$$

Proposed solution | Optimization of classification performance

The individual and fitness calculation

The individual

- Program tree used as $g(x)$
- Individual is a classifier

Classification function

$$\text{classify}(x) = \begin{cases} 1, \textit{ if } \mathbf{g(x)} \geq C \\ 0, \textit{ otherwise} \end{cases}$$

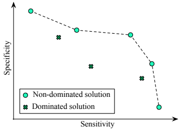
Single-objective fitness (maximization)

$$\textit{so fitness} = \text{sensitivity} * \text{specificity}$$

Multi-objective fitness Why?

- Separate objectives
- Solution - set of classifiers
- Hypervolume (HV) and Crowding Distance (CD)

[Zitzler'99,Nebro'08]



On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 24 | 42

Proposed solution | Optimization of classification performance

The individual and fitness calculation

The individual

- Program tree used as $g(x)$
- Individual is a classifier

Classification function

$$\text{classify}(x) = \begin{cases} 1, \textit{ if } \mathbf{g(x)} \geq C \\ 0, \textit{ otherwise} \end{cases}$$

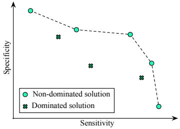
Single-objective fitness (maximization)

$$\textit{so fitness} = \text{sensitivity} * \text{specificity}$$

Multi-objective fitness

- Separate objectives
- Solution - set of classifiers
- Hypervolume (HV) and Crowding Distance (CD)

[Zitzler'99,Nebro'08]



On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 24 | 42

Proposed solution | Optimization of classification performance

The individual and fitness calculation

The individual

- Program tree used as $g(x)$
- Individual is a classifier

Classification function

$$\text{classify}(x) = \begin{cases} 1, \textit{ if } \mathbf{g(x)} \geq C \\ 0, \textit{ otherwise} \end{cases}$$

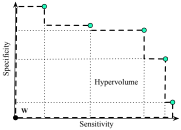
Single-objective fitness (maximization)

$$\textit{so fitness} = \text{sensitivity} * \text{specificity}$$

Multi-objective fitness

- Separate objectives
- Solution - set of classifiers
- Hypervolume (HV) and Crowding Distance (CD)

[Zitzler'99,Nebro'08]



On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 24 | 42

The individual and fitness calculation

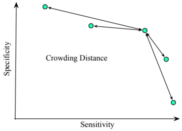
The individual
- Program tree used as $g(x)$
- Individual is a classifier

Classification function
$$\text{classify}(x) = \begin{cases} 1, \; if\; \mathbf{g(x)} \geq C \\ 0, \; otherwise \end{cases}$$

Single-objective fitness (maximization)
$$so\ fitness = \text{sensitivity} * \text{specificity}$$

Multi-objective fitness
- Separate objectives
- Solution - set of classifiers
- Hypervolume (HV) and Crowding Distance (CD)

[Zitzler'99,Nebro'08]



Classification performance - summary

Search for optimal classification
- GEP searches for $g(x)$ of upper-bounded size
- Input
  - Parameters of the IN model
  - Behavior expressed via time series

Fitness function
- Incorporates sensitivity and specificity
- Single and multi-objective search is considered

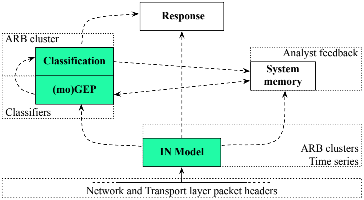
Classification performance - summary

Search for optimal classification
- GEP searches for $g(x)$ of upper-bounded size
- Input
  - Parameters of the IN model
  - Behavior expressed via time series

Fitness function
- Incorporates sensitivity and specificity
- Single and multi-objective search is considered

Idiotypic Network-based IDS (INIDS) architecture

Integration
- IN model placed in E-box
- Classifier search and classification process placed in A-box



IN model for network traffic clustering

STCP (Stream TCP) ARB

- Groups finished TCP connections
- Parameters
  - TCP/IP header fields
  - Termination code

Distance function

- Weighted sum of distance functions
- Bias towards outside traffic

Proposed solution | Proposed architecture

IN model for network traffic clustering

STCP (Stream TCP) ARB

- Groups finished TCP connections
- Parameters
  - TCP/IP header fields
  - Termination code

Distance function

- Weighted sum of distance functions
- Bias towards outside traffic

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 27 | 42

The input of GEP algorithm

Parameters of IN model

- **Lifetime** (L) - sum of lifetimes of ARBs in a cluster
- **Size** (S) - number of ARBs in a cluster
- **Ratio** (R) - Lifetime to a cluster capacity ratio

Time series

- Sliding window method ($\Delta t = 1$s)
- Data sample $x = \{tw(L_i^c),\ tw(S_i^c),\ tw(R_i^c)\}$

Proposed solution | Proposed architecture

The input of GEP algorithm

Parameters of IN model

- **Lifetime** (L) - sum of lifetimes of ARBs in a cluster
- **Size** (S) - number of ARBs in a cluster
- **Ratio** (R) - Lifetime to a cluster capacity ratio

Time series

- Sliding window method ($\Delta t = 1$s)
- Data sample $x = \{tw(L_i^c),\ tw(S_i^c),\ tw(R_i^c)\}$

On Capturing Vital Properties of Denial of Service Attacks Using Metaheuristic Approaches 28 | 42

Contents

IN model evaluation outline

Data sets

- Real-world traffic: **MIT-LL**
  One week of regular traffic, two weeks of attacks
- Simulated traffic: **ns2**
  Real-world topology, HTTP traffic clouds, 800s of traffic

Performance analysis

- Thresholding anomaly detection
- The visual representation and information content

Thresholding anomaly detection

Normal/Abnormal modelling

- Regular traffic is used to set up thresholds of L, S and R
- Exceeding of any of the thresholds is taken as an attack

Method of comparison

Thresholding the typical parameters of network traffic

- General
  - Packets and bytes per second
- TCP specific
  - Control flags (SYN, FIN, RST) per second

Thresholding anomaly detection: results

The **MIT-LL** data set

	IN model	Method of comparison
Sen	**0.83**	0.33
Spe	**0.996**	0.99

- Unreported anomalies
  - 2 in normal traffic
  - 400 in attack traffic
- Normal $\gg$ Attack

The ns2 data set

	IN model	Method of comparison
Sen	**0.831**	0.153
Spe	0.794	**1.0**

- "Clean" traffic
- DDoS + flash event

Thresholding anomaly detection: results

The MIT-LL data set

	IN model	Method of comparison
Sen	**0.83**	0.33
Spe	**0.996**	0.99

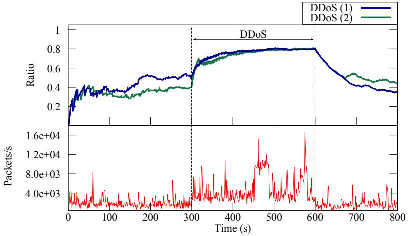
- Unreported anomalies
  - 2 in normal traffic
  - 400 in attack traffic
- Normal ≫ Attack

The ns2 data set

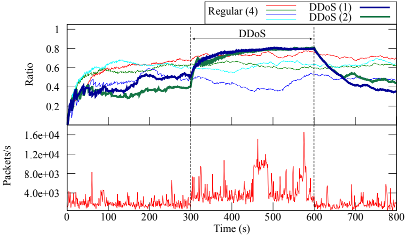
	IN model	Method of comparison
Sen	**0.831**	0.153
Spe	0.794	**1.0**

- “Clean” traffic
- DDoS + flash event

Visual representation and information content



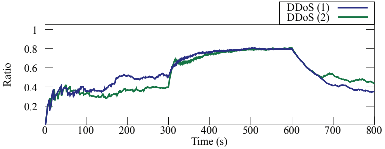
Visual representation and information content



IN model evaluation - summary

Results

- Sensitivity = 0.831
- Specificity = 0.794
- Place for improvement for GEP
  - Functions describing cluster behavior



GEP evaluation outline

Learning process

- Learning set: output of IN model for ns2 data set
- Time series for sliding window $w = 5, 10 - 60$
- 100 independent runs for each GEP setup

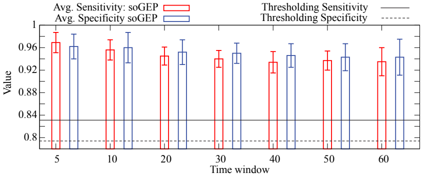
Performance analysis

- Single-objective GEP vs thresholding
- Multi-objective GEP vs soGEP vs reference methods
- Reference methods
  - Support Vector Machine (SVM)
  - Bayesian Network (BN)
  - Self-Organizing Map (SOM)

Single-objective GEP

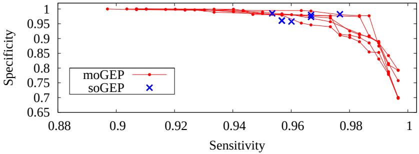
Results

- Relative improvement
  - Sensitivity: 12.4% – 16.6%, Specificity: 18.8% – 21.2%
- Performance influenced by w



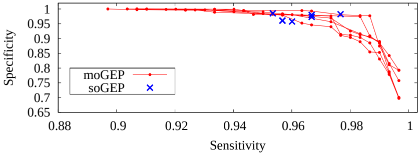
Multi-objective GEP

- Set of solutions instead of a single classifier
- soGEP explores a part of search space
- soGEP outperformed part of moGEP for 0.7% cases



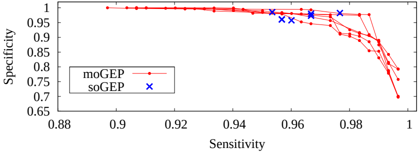
Multi-objective GEP

- Set of solutions instead of a single classifier
- soGEP explores a part of search space
- soGEP outperformed part of moGEP for 0.7% cases



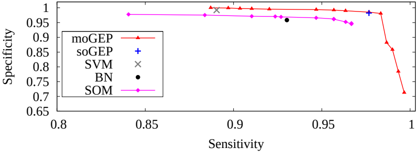
Multi-objective GEP

- Set of solutions instead of a single classifier
- soGEP explores a part of search space
- soGEP outperformed part of moGEP for 0.7% cases



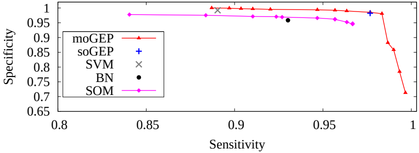
Comparison with other classification methods

- SVM, BN, SOM more sensitive to w than GEP
- soGEP and moGEP are better for $w = 5$ and 10
- For $w > 10$ the results of moGEP are complemented



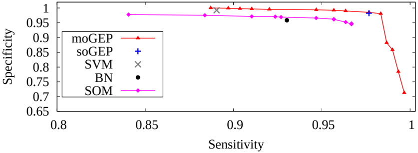
Comparison with other classification methods

- SVM, BN, SOM more sensitive to w than GEP
- soGEP and moGEP are better for $w = 5$ and 10
- For $w > 10$ the results of moGEP are complemented



Comparison with other classification methods

- SVM, BN, SOM more sensitive to w than GEP
- soGEP and moGEP are better for $w = 5$ and 10
- For $w > 10$ the results of moGEP are complemented

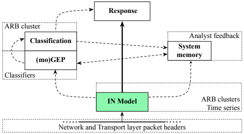


Contents

Conclusions

IN model for traffic clustering

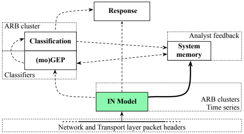
- **IN parameters emphasize the DDoS properties**
  - DDoS detectable by thresholding approach
  - IN more accurate (but slower) than regular thresholding
- Information feed useful for traffic analysis



Conclusions

IN model for traffic clustering

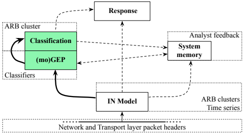
- IN parameters emphasize the DDoS properties
  - DDoS detectable by thresholding approach
  - IN more accurate (but slower) than regular thresholding
- **Information feed useful for traffic analysis**



Conclusions

GEP for traffic classification

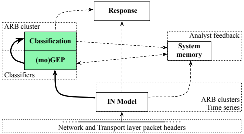
- **Offers adaptive generation of traffic classifiers**
- Outperforms SVM, BN and SOM for $w \leq 10$
- Proposed moGEP offers versatile solutions
  - Complemented by SVM, BN and SOM for $w > 10$



Conclusions

GEP for traffic classification

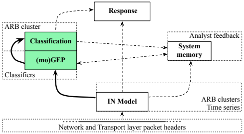
- Offers adaptive generation of traffic classifiers
- **Outperforms SVM, BN and SOM for** $w \leq 10$
- Proposed moGEP offers versatile solutions
  - Complemented by SVM, BN and SOM for $w > 10$



Conclusions

GEP for traffic classification

- Offers adaptive generation of traffic classifiers
- Outperforms SVM, BN and SOM for $w \leq 10$
- **Proposed moGEP offers versatile solutions**
  - Complemented by SVM, BN and SOM for $w > 10$



Summary of contributions

- IN model adapted and applied to cluster the network traffic
- GEP adapted and applied to search for traffic classifiers
- Multi-objective GEP proposed for more versatile solutions
- IN and GEP integrated within the CIDF framework
- IN and GEP evaluated on real-wold traffic and simulations
- Publications: 1 journal, 7 conference papers

Perspectives

- Extending the scope of the architecture
  - Distributed approach for in-depth forensics
  - More general traffic classification and profiling
- Parallel design of the IN model for better performance
  - Current speed estimated to ~375 Mb/s on QC Intel Xeon 3.2 GHz, 16 GB RAM (MacPro v3.1)
- Application of GEP to dynamic optimization problem
  - "online analysis, online learning" scenario

Thank you for your attention

Awaiting distributed attacks

Time for the board to find vulnerabilities and attempt distributed flooding attacks...

Publications

Journal M. Ostaszewski, F. Seredynski and P.Bouvry, “Coevolutionary-based Mechanisms for Network Anomaly Detection”, JMMA 2007

Conf. rank A M. Ostaszewski, F. Seredynski and P.Bouvry, “Immune Anomaly Detection Enhanced With Evolutionary Paradigms ”, GECCO 2006

Conf. rank A M. Ostaszewski, F. Seredynski and P.Bouvry, “A Nonself Space Approach to Network Anomaly Detection”, NIDISC(IPDPS) 2006

Conf. rank A M. Ostaszewski, P. Bouvry and F. Seredynski, “An Approach to Intrusion Detection by Means of Idiotypic Networks Paradigm”, CEC 2008

Conf. rank A M. Ostaszewski, P. Bouvry and F. Seredynski, “Denial of Service Detection and Analysis Using Idiotypic Networks Paradigm”, GECCO 2008

Conf. rank A M. Ostaszewski, F. Seredynski and P.Bouvry, “Adaptive and Dynamic Intrusion Detection by Means of Idiotypic Networks Paradigm”, NIDISC(IPDPS) 2008

Conf. rank A M. Ostaszewski P. Bouvry and F. Seredynski, “Multiobjective Classification with moGEP: An Application in the Network Traffic Domain”, GECCO 2009

References

Nielsen’09 J. Nielsen,“Nielsen’s Law of Internet Bandwidth”, 2009 (online) `http://www.useit.com/alertbox/980405.html`

IWS’09 Internet World Stats, 2009 (online) `http://www.internetworldstats.com/stats.htm`

CSI/FBI’04 L. Gordon et al., “CSI/FBI Computer Crime and Security Survey”, Computer Security Institute, 2004

CSOS&R’09 W. Brenner, “DDoS Attacks Are Back (and Bigger Than Before)”, CSO Security & Risk, 2010 (online) `http://www.csoonline.com/article/515614/DDoS_Attacks_Are_Back_and_Bigger_Than_Before_`

Fawcett’03 T. Fawcett, “ROC graphs: Notes and practical considerations for data mining researchers”, HP Tech. Rep., 2003

Ferreira’06 C. Ferreira, “Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence”, Springer, 2006

Jin’03 C. Jin et al., “Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic”, CCS 2003

Peng’04 T. Peng et al., “Protection From Distributed Denial of Service Attacks Using History-based IP Filtering”, ICC 2003

Karas.’07 A. Karasaridis et al., “Wide-scale botnet detection and characterization”, First Workshop on Hot Topics in Understanding Botnets, 2007

References, continued

Feinst.’03 L. Feinstein et al., “Statistical Approaches to DDoS Attack Detection and Response”, DISCEX 2003

Chen’05 Y. Chen et al., “Filtering of Shrew DDoS Attacks in Frequency Domain”, LCN 2005

Carl’06 G. Carl et al., “Wavelet based Denial-of-Service detection”, Computers & Security, 2006

Xie’06 Y. Xie et al., “A Novel Model for Detecting Application Layer DDoS Attacks”, IMSCCS 2006

Ptacek’98 T. H. Ptacek, T. N. Newsham, “Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection”, Secure Networks, Inc., 1998

Zitzler’99 E. Zitzler et al., “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach”, IEEE Transactions on Evolutionary Computatio, 1999

Nebro’08 A. Nebro et al., “AbYSS: Adapting Scatter Search to Multiobjective Optimization”, IEEE Transactions on Evolutionary Computatio, 2008

Processing speed of the architecture

Processing speed of the IN model

- Avg. speed ~375 Mbit/s
  - Quad-Core Intel Xeon 3.2 GHz, 16 GB RAM (MacPro v3.1)
  - ns2 data set
- Further speedup: parallel design

Processing speed of the classification methods

- Time of GEP learning process varies from 427 s — 554 s
- The three methods are in general faster than GEP
- Sufficient for “online analysis, offline learning” scenario