THREE APPROACHES TO INTRUSION DETECTION

Analysis and Enhancements

VI National Computer and Information Security Conference

ACIS – COLOMBIA

by

Pedro A. Diaz-Gomez and Dean F. Hougen

June 2006
Outline

- Goal
- Basic Concepts
- The Three Models
  - Denning — Intrusion Detection Model
  - Crosbie & Spafford — Genetic Programming
  - Mé — Genetic Algorithms
- Conclusions & Future Work
Goal

- To review and analyse three approaches to intrusion detection:
  - An Intrusion-Detection Model
    by Dorothy Denning
  - Applying Genetic Programming to Intrusion Detection
    by Mark Crosbie and Eugene H. Spafford
  - Security Audit Trail Analysis Using Genetic Algorithms
    by Ludovic Mé
Basic Concepts: Intrusion Detection Systems

- *Intrusion Detection System (IDS)*
  - system to detect intrusions in a computer or computer network

- *Intrusion*
  - unauthorized attempt to access a system

- *Security Auditing*
  - formal examination of actions taken by system users

- *Audit Data*
  - records of actions taken by identifiable and authenticated users
Basic Concepts: Evolutionary Computation (EC)

- Inspired by Biological Evolution
  - Biological Evolution
    - Creates and Modifies Species by Natural Selection
  - Evolutionary Computation
    - Creates and Modifies “Solutions” by Artificial Selection
- Genes — *Hereditary Units that* Determine Characteristics
- Chromosomes — Collections of Genes in Individuals
- Populations — Collections of Individuals
Basic Concepts:
Evolutionary Computation (EC)

- Gene
- Chromosome
- Population
- Parents
- Offsprings
- Crossover
- Mutation
Basic Concepts: Genetic Programming (GP)

- John Koza has used a form of EC to evolve Lisp programs
- Programs in Lisp can be expressed as *parse trees*
Outline

- Goal
- Basic Concepts
- The Three Models
  - Denning — Intrusion Detection Model
  - Crosbie & Spafford — Genetic Programming
  - Mé — Genetic Algorithms
- Conclusions & Future Work
The Three Models

**Denning’s Model**
- Audit records in real time
- Anomaly Detection

**Crosbie & Spafford Prototype**
- Audit records in real time
- Anomaly Detection
- Suspicion report
- Rules using GP. Anomaly Detection using Those Rules

**Mé’s Audit Trail Analysis**
- Off-Line Audit records
- Misuse Detection
- Possible attacks
- Intrusions Pre coded. Pattern Matching
The Three Models Complemented

- Distributed K. Base
- Using AI to generate them.
- Known Attacks.
- Real Time System

Audit records in real time

Possile attacks
Outline

- Goal
- Basic Concepts
- The Three Models
  - Denning — Intrusion Detection Model
  - Crosbie & Spafford — Genetic Programming
  - Mé — Genetic Algorithms
- Conclusions & Future Work
Denning's Intrusion Detection Model Components

- The model has six *components*:
  - Subjects,
  - Objects,
  - Audit records,
  - Profiles,
  - Anomaly records, and
  - Activity rules.
Outline

● Goal

● Basic Concepts

● The Three Models

● Denning — Intrusion Detection Model

✓ Crosbie & Spafford — Genetic Programming

● Mé — Genetic Algorithms

● Conclusions & Future Work
A *GP* Intrusion Detection Model

![Diagram](image)

### Training Scenarios

<table>
<thead>
<tr>
<th>Type of scenario</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 connections with 1 second delay</td>
<td>90%</td>
</tr>
<tr>
<td>10 connections with 5 second delay</td>
<td>70%</td>
</tr>
<tr>
<td>10 connections with 30 second delay</td>
<td>40%</td>
</tr>
<tr>
<td>10 connections every minute</td>
<td>30%</td>
</tr>
<tr>
<td>Rapid connections, then random pauses</td>
<td>80%</td>
</tr>
<tr>
<td>Intermittent connections</td>
<td>10%</td>
</tr>
<tr>
<td>Connections to privileged ports</td>
<td>90%</td>
</tr>
<tr>
<td>Connections to any port</td>
<td>70%</td>
</tr>
</tbody>
</table>

Crosbie and Spafford. Applying Genetic programming to Intrusion Detection.
A GP Intrusion Detection Model

Suspicion reported by Agents

Suspicion Value

<table>
<thead>
<tr>
<th>Activities</th>
<th>Agent 1</th>
<th>Agent 2</th>
<th>Agent 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connections to privileged ports</td>
<td>83%</td>
<td>100%</td>
<td>98%</td>
</tr>
<tr>
<td>Login then long pauses then logins</td>
<td>31%</td>
<td>26%</td>
<td>0%</td>
</tr>
<tr>
<td>Logins and ftp with long pauses</td>
<td>73%</td>
<td>47%</td>
<td>25%</td>
</tr>
</tbody>
</table>

Training Scenarios

<table>
<thead>
<tr>
<th>Type of scenario</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 connections with 1 second delay</td>
<td>90%</td>
</tr>
<tr>
<td>10 connections with 5 second delay</td>
<td>70%</td>
</tr>
<tr>
<td>10 connections with 30 second delay</td>
<td>40%</td>
</tr>
<tr>
<td>10 connections every minute</td>
<td>30%</td>
</tr>
<tr>
<td>Rapid connections, then random pauses</td>
<td>80%</td>
</tr>
<tr>
<td>Intermittent connections</td>
<td>10%</td>
</tr>
<tr>
<td>Connections to privileged ports</td>
<td>90%</td>
</tr>
<tr>
<td>Connections to any port</td>
<td>70%</td>
</tr>
</tbody>
</table>

Crosbie and Spafford.
Applying Genetic programming to Intrusion Detection.
Outline

● Goal

● Basic Concepts

● The Three Models

● Denning — Intrusion Detection Model

● Crosbie & Spafford — Genetic Programming

✓ Mé — Genetic Algorithms

● Conclusions & Future Work
A Genetic Algorithm Approach Architecture

In Audit trail → Event classification → Analysis engine

WV Weighted Vector → AE Known Attacks

Observed vector OV

Out Attacks
A Genetic Algorithm Approach
Fitness Function

- The fitness function proposed by Ludovic Mé

\[ F(I) = \alpha + \sum_{i=0}^{N_a-1} W_i * I_i - \beta \cdot T^2 \]

- Hypothesis \( I \) is found such that:

  - \( \sum_{i=0}^{N_a-1} W_i * I_i \) is maximized, and
  
  - \( \sum_{j=0}^{N_a-1} AE_j * I_j \leq O_j \), for all \( 1 \leq i \leq N_e \). Constraint.
A Genetic Algorithm Approach
Fitness Function – Our Proposal

Problems

- $\sum_{i=0}^{Na-1} I_i$ is guiding incorrectly.
- Penalty is not well calculated

Solution proposed

$F(I) = \alpha + \sum_{i=0}^{Na-1} W_i * I_i - \beta \cdot T^2$

$F(I) = Ne - T$
A Genetic Algorithm Approach Fitness Function – Our Proposal

\[ F(I) = N_e - T \]

<table>
<thead>
<tr>
<th>User</th>
<th>False +</th>
<th>False -</th>
<th>Detected</th>
<th>False +</th>
<th>False -</th>
<th>Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>2051_7</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>2051_11</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>2506_15</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Zero Vector</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>One Intrus.</td>
<td>0</td>
<td>0.1</td>
<td>0.9</td>
<td>0%</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>Two Intrus.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Three Intrus.</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Outline

- Goal
- Basic Concepts
- The Three Models
  - Denning — Intrusion Detection Model
  - Crosbie & Spafford — Genetic Programming
  - Mé — Genetic Algorithms
- Conclusions & Future Work
Conclusions & Future Work

- Denning's Model
  - The assumption of abnormal as deviation from normality is a good start

- Our proposals
  - Could be complemented with *misuse* detection
  - Use of *sentinel profiles* for non common activity
  - Use of tendency of the mean
Conclusions & Future Work

- Denning's Model
  - Use of profiles, metrics and models is a great idea

- Our proposals
  - Try to overcome the heavy of the system
    - Distributivity
    - Maintenance
    - Risk analysis
Conclusions & Future Work

- Denning's Model
  - Use of Classes of Profiles is prominent

- Our proposals
  - Take into account the number of users in each Class
  - Complement with the use of the tendency of the mean
Conclusions & Future Work

- Crosbie & Spafford Model
  - Idea of use of distributed Agents is excellent

🌟 Our proposals
  - Take into account
    - The *control* of those
    - The *overload* impose in the system
Conclusions & Future Work

- Crosbie & Spafford Model

  - Use of GP in order to improve the capture of novel attacks

- Our proposals

  - Specification of
    - Fitness function and its parameters
    - Parameters of the evolution
    - SAL and MUX
Conclusions & Future Work

- Crosbie & Spafford Model
  - Use of GP in order to improve the capture of novel attacks

- Our proposals
  - More test
  - Compare future results with other approaches
Conclusions & Future Work

- Mé Model
  - Join objective and constraint in the fitness function

- Our proposals
  - *New fitness function* that uses only the constraint and
  - The objective is obtained with a new operator: the *union operator*
Conclusions & Future Work

- Mé’s Model
  - Good idea to use a matrix of misuse, and to encode intrusions as a chromosome

- Our proposals
  - Augment the system with the possibility of
    - Consider different users
    - Consider more intrusions
    - report user activity not considered in the analysis
Conclusions & Future Work

- Mé’s Model
  - Use a matrix of misuse, and to encode intrusions as a chromosome

- Our proposals
  - Augment the system with the possibility of
    - capture novel attacks
    - capture abnormal activity
    - disaggregate intrusions as exclusive
Thanks!

I would like to thank The University of Oklahoma for sponsoring this trip to Acis-Colombia.

I would like to thank Dr. Dean Hougen, for his advising, support and patience. From him, I learned not only in the classroom, but with his example, the way to do science and be better.

I would like to thank ACIS for this opportunity to share our research experiences and learn from a selected group of Panellists.
Bibliografy


- P. Diaz-Gomez and D. Hougen
  - Analysis and Mathematical Justification of a Fitness Function used in an Intrusion Detection System In Proceedings of the Seventh Annual Genetic and Evolutionary Computation Conference 2005
QUESTIONS