Computer Network Intrusion Detection
Via Neural Networks Methods

Vu Dao
phongvu_98@yahoo.com
July 18, 2002
Outline

- Goals
- Background on Intrusion Detection Systems (IDS)
- Types of IDS
- Why Applying Neural Networks Techniques?
- User Profiling in the UNIX OS Environment
- Study of the Proposed Methods
  - Implementation of the Proposed IDS
  - Results
  - Trade-Offs of the Proposed Methods
- Summary
- Future Work
Goals

- Design and implement new intrusion detection systems that deal with changes in user profile (i.e. user behavior)
- Compare the proposed methods with other statistical methods to the intrusion detection problem, explain the trade-offs and the potential advantages of the proposed methods
Background on Intrusion Detection Systems

1999 DARPA Study [1]

Types of Evaluation
- U2R - User Illegally Became Root (eject, fdformat, ps, ...)
- DoS - Denial of Service (selfping, smurf, tcpreset, ...)
- R2L - Remote User Illegally Accessed a Local Host (guest, ftpwrite, xsnoop, ...)

Results of Detecting Intruders
- 80% Success for Old Attacks
- 25% Success for New and Novel Attacks

DARPA: Defense Advanced Research Project Agency
Types of IDS

- Audit-Trail IDS
- Network Monitoring IDS
- Others
Audit-Trail Methods

- Classical Artificial Intelligence (AI)
  - Statistical or Anomaly
  - Rule-Based, Signature or Misuse
- Soft-Computing Artificial Intelligence
  - Back Propagation (BP)
  - Radial Basis Function (RBF)
  - Genetic Algorithm (GA)
Research Concentration

- Previous Works Concentrate on System or Network
  - System Traffic or System Log
  - Goal is to Detect Intrusion on System or Network

- This Research Concentrates on User Account
  - Account Traffic or Account Log
  - Goal is to Detect Intrusion on a Specified Account
Why Applying Neural Network?

- Statistical Method
  - Used in Detecting New Attacks
  - Inaccurate
    - 75% Success Rate [2] for currently best research system

- Neural Network Has Self Learning Capability
  - Supervised Learning for Input-Output Mapping
  - Adapt Synaptic Weights to Changes in the Surrounding Environment

User Profiling in the UNIX OS Environment (1/2)

- Events Used in User Profiling[3]
  - Activities of the System as a Whole
  - Activities of Users
  - Activities of Particular Terminals
  - Transactions Involving Particularly Sensitive Files or Programs
  - Transactions Involving Particular Sensitive System Files or Programs

User Profiling in the UNIX OS Environment (2/2)

- Attributes of Users in Profiling
  - Command Sets, Time of Login, Host, CPU Time
  - Short-Term
    - Constant Profile
  - Long Term
    - Profile Drift
- Case Study

User Profiling -- Case Study (1/2)
User Profiling -- Case Study (2/2)
Study of the Proposed Methods

- Neural Network Methods
  - Back Propagation
    - Gradient Descent (GD)
    - Gradient Descent with Momentum
    - Variable Learning Rate GD with Momentum
    - Conjugate Gradient
    - Quasi Newton
Feed Forward Neural Networks

Input Layer

$X_1$

$X_2$

$X_3$

...$

X_n$

Hidden Layer

Output Layer

Feed Forward Neural Networks
Generated Data File

<table>
<thead>
<tr>
<th>Training Data (5000 Samples)</th>
<th>Testing Data (2000 Samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU1</td>
<td>CU2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File 1 CU = 5</th>
<th>File 2 CU = 6</th>
<th>File 3 CU = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25000</td>
<td>30000</td>
<td>35000</td>
</tr>
<tr>
<td>Testing Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>12000</td>
<td>14000</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35000</td>
<td>42000</td>
<td>49000</td>
</tr>
</tbody>
</table>
Data set 1, CU = 5
Data set 2, CU = 5
Data set 3, CU = 5
Results (BFGS - 5 Samples) Test Data 1

BFGS = Broyden, Fletcher, Goldfarb, Shanno
Results (BFGS - 6 Samples) Test Data 1
Results (BFGS - 7 Samples) Test Data 1
## BFGS Result

<table>
<thead>
<tr>
<th></th>
<th>CU = 5</th>
<th>CU = 6</th>
<th>CU = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Error</strong></td>
<td>2.7%</td>
<td>2.5%</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>18%</td>
<td>19%</td>
<td>21%</td>
</tr>
<tr>
<td><strong>CPU Error</strong></td>
<td>3.6%</td>
<td>1.8%</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>16%</td>
<td>19%</td>
<td>21%</td>
</tr>
<tr>
<td><strong>Memory Error</strong></td>
<td>3.1%</td>
<td>3.9%</td>
<td>3.3%</td>
</tr>
<tr>
<td></td>
<td>18%</td>
<td>16%</td>
<td>20%</td>
</tr>
<tr>
<td><strong>Combined Error</strong></td>
<td>1.9%</td>
<td>2%</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>18%</td>
<td>19%</td>
<td>21%</td>
</tr>
</tbody>
</table>
## CGP Result

<table>
<thead>
<tr>
<th></th>
<th>CU = 5</th>
<th>CU = 6</th>
<th>CU = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Error</strong></td>
<td>0.4% 20%</td>
<td>1.1% 21.4%</td>
<td>2.84% 20%</td>
</tr>
<tr>
<td><strong>CPU Error</strong></td>
<td>2.0% 2.9%</td>
<td>2.23% 18.6%</td>
<td>2.0% 17%</td>
</tr>
<tr>
<td><strong>Memory Error</strong></td>
<td>0.7% 18.5%</td>
<td>0.5% 20%</td>
<td>3.1% 19%</td>
</tr>
<tr>
<td><strong>Combined Error</strong></td>
<td>0% 20%</td>
<td>0.31% 21.43%</td>
<td>2.0% 20%</td>
</tr>
</tbody>
</table>
Summary Result

Performance of CGP & BFGS Methods

Percentage Error [cgr = blue, bfgs = red]

Number of CUs
Trade-Offs of the Proposed Methods

- **Advantage**
  - Adaptive to Profile Drift
  - Software Based Neural Networks
  - Added protection to critical account / system

- **Disadvantage**
  - Requires More Computing Resources
  - Require Negative Samples to Train Neural Networks
  - Must be configured to each user
Summary

- Profile Computer Users Successfully via Basic Attributes
- Neural Networks Capable of Classifying Users
Future Work

- Implement Other Neural Network Techniques
  - Radial Basis Functions
    - Weights has local affect on neuron
- Use Other User Profile Attributes
- Analyze Results to Improve Performance