

# Genetic Algorithm Approach for Intrusion Detection

---

Y. B. Reddy

# Contents

---

- Introduction
  - Intrusion and Intrusion Detection System (IDS)
  - Various types of Intrusion Detection Systems
  - Present status of IDS and Our Goals
- Current Approaches
  - Misuse Detection vs. Anomaly Detection
  - Shortfalls with Current IDS
- Our Proposed Approach
- Genetic Algorithms
- Relevance of our Model
- Results and Discussion
- Conclusion
- References

# Introduction

---

## Intrusion and Intrusion Detection System

- Any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource.
- Identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems.

# Introduction --

---

## Various Types of Intrusion Detection Systems

- Host-Based IDS
  - Collects and analyzes locally & highly effective
  - On large network with several end points (web-server) may downsize the system performance
- Network-Based IDS
  - Monitor on particular network – distributed in nature
  - Uses ‘packet-sniffing’ technique – to pull data from TCP/IP or data related to bandwidth theft / denial of service
- Application-Based IDS
  - Monitor only specific applications (DBMS, accounting systems, etc)
  - Can detect attacks through analysis of application log files and many types of suspicious activity

# Introduction --

---

## Present Status of IDS

- Commercially available IDS are Network-Based (mostly)
- Lot of false alarms
  - ◆ False positives – generate alarm when there is no true intruder
  - ◆ False negative – no alarm when actual intrusion happens
- Lack of effectiveness, adaptability, and extensibility
- More than 200 types of systems are available in the market

# Introduction - -

---

## Our Goal

- ❑ User program activities can be monitored and modeled
- ❑ Resources to be protected
- ❑ Models of the normal or legitimate behavior on the resources
- ❑ Efficient methods that compare real-time activities against the models and report probably 'intrusive' activities

# Introduction

---

## IDS Functions

- Monitoring and analyzing both user and system activities
- Analyzing system configurations and vulnerabilities
- Assessing system and file integrity
- Ability to recognize patterns typical of attacks
- Analysis of abnormal activity patterns
- Tracking user policy violations

# Current Approaches

---

## Misuse Detection

- Record the specific patterns of intrusions
- Monitor current audit trails (event sequences) and pattern matching
- Report the matched events as intrusions
- Advantage - Detects accurately and efficiently
- Disadvantage – writing signatures of all possible variations of the pertinent attack
- Ex:- Rule-based systems

# Current Approaches

---

## Anomaly Detection

- Deviated from stored patterns
- Systems needs to train with lot of examples
- Involves various intelligent techniques
- Presently statistical tools were used to detect unusual and unexpected behavior

# Current Approaches

---

## Shortfalls with Current IDS

Variants	Intrusions change easily and frequently
False Positive	generate alarm when there is no true intruder (alarms for uninterested signature)
False negative	no alarms when intrusion happens (missing events of interest)
Data overload	amount of data grows rapidly

# Our Proposed Approach

---

A systematic framework to:

Build good models – select appropriate features of audit data to build ID models

Build better models – architect a hierarchical detector system that combines multiple detection models

Build updated models – dynamically update and deploy new detection system as needed

# Our Proposed Approach

---

Support for the future selection and model construction process

- Apply data mining algorithms to find consistent inter and intra audit record (event) patterns
- Use features and time windows in the discovered patterns to build detection models
- A support environment to semi-automate this process

# Our Proposed Approach

---

## Combine multiple detection models

- Each (base) detector model monitors one aspect of the system
- They can employ different techniques and be independent of each other
- The learned (meta) detector combines evidence from a number of these detectors

# Our Proposed Approach

---

## Design intelligent agent-based architecture

- Learning agents – continuously compute (learn) the detection models
- Detection agents – use the (updated) models to detect intrusions

# Data Mining Models

---

- Find interesting patterns that deviate from the regular patterns
  - ◆ Create original Patterns (includes intrusion and normal data)
  - ◆ From the incoming patterns – remove the un-effected data
  - ◆ Create resulting database from this data using frequent episode rules
  - ◆ Use the data mining process to identify the anomalies
- 1. YBR and Ratan
  - Intrusion Detection Using Data Mining Techniques
- 2. Santosh, Phoha, and YBR:
  - CLIQUE Clustering Approach to detect Denial-of-service Attacks (5<sup>th</sup> Annual IEEE Information Assurance Workshop June 2004)

# Genetic Algorithms

---

- Family of computational models based on principles on evolution and natural selection
- Convert the problem in a specific domain into a model by using a chromosome-like data structure, and evolve the chromosomes using selection, recombination and mutation operators
- Process starts with randomly selected population of chromosomes (representations of problem to be solved)
- Each position of chromosome is a gene.
- Set of chromosomes is population

10011111 - example of chromosome

Two Basic Operators

crossover and mutation

**Fitness Function**: numerical value which is proportional to the ability or utility of individual represented by that chromosome

# Genetic Algorithms

---

## Crossover – Recombination operator

Takes two individuals (chromosomes) and cuts their chromosome strings at some randomly chosen position and swaps the tail positions.

<u>Original</u>	<u>Crossover (single Point)</u>
xxx xxxxx	xxx00000
000 0000	000xxxxx

## Mutation

Substitute one or more bits of an individual randomly by a new value (0 or 1)

011101101100	
011001101100	(mutate 4 <sup>th</sup> bit)

Other operators not discussed here

# Genetic Algorithms

---

## Sample Dataset

Time Stamp	Duration	Src-Port	Dst-Port	Src-bytes	Dst-bytes	Protocol	Intrusion
2345	1	1360	080	235	123	TCP/IP	true

Conversion: 2345,1,1360,080,235,123,TCP (use code for protocol)

Chromosome can be formed with decimals, integers, or binary  
(see ecp Package)

(Other data like Source-IP and Destination IP also helps)

# Genetic Algorithms

---

## Genetic Algorithm for fitness function

- A constant size population of individuals
- Each individual represents a point in the search space for a given problem through a suitable coding
- A fitness value is assigned to each individual in the population
- Individuals are ranked and selected according to their fitness in such a way that more fit individuals are more likely to enter the relevancy group
- Genetic operators such as crossover and mutation are applied to pairs of individuals or single individual in order to produce new individuals

# Genetic Algorithms

The evaluation function is one of the most important parameters in Genetic Algorithms (Crosbie and Spafford)

The outcome is calculated based on whether a field of the connection matches the pre-classified data set, and then multiply the weight of the field. The order of weights depend upon network connections.

$$Outcome = \sum_{i=1}^n Matched * Weight_i$$

The suspicious level of threshold is calculated as

$$\Delta = |outcome - suspicious\_level|$$

Once mismatch happens, the penalty value is computed using absolute difference:

$$penalty = \left( \frac{\Delta * ranking}{100} \right)$$

The fitness of a chromosome is computed using the above penalty

$$Fitness = 1 - penalty$$

The range of fitness value is between 0 and 1

# Genetic Algorithms

---

## Present Status

- Presently the research work was done blindly using fitness function or package programs
- Some of the work was used similar to neural network learning techniques or bucket brigade algorithm (classifier systems)
- Bioinformatics approach (GA approach) for a pair wise sequence alignment is one of the recent paper

(Intrusion Detection: a Bioinformatics Approach – Erick Breimer, 19<sup>th</sup> *International computer security Applications Conference, Dec 8-12, 2003*)

# Future Work

---

- How can we be sure it will detect a specific intrusion?
- Can we compute probability *a priori* of its effectiveness?
- What sort of overhead would such a system impose on a production system?

Suggestion:

- Use the following concept to find the potential intruder:

The prediction of the degree of exposure to solvent of amino acid residues via Genetic programming by Simon Handley

“Genetic Programming”, 1996, edited by J. Koja; D. Goldberg; F. Fogal and R Riolo

# References

---

- Wei Li., “A Genetic Algorithm Approach to Network Intrusion Detection”, SANS Institute 2004.
- Mark Crosbie and Gene Spafford., “Applying Genetic Programming to Intrusion Detection”, Working notes for the AAAI Symposium on Genetic Programming, 10-12, 1995.
- Bob Adolf., “New Paradigms for Intrusion Detection Using Genetic Programming”, Smallbusinesscomputing.com, May 25, 2003.
- Cris Sinclair, Lyn Pierce, and Sara Matzner., “An application of machine Learning to Network Intrusion Detection”, 15th Annual Computer Security Applications Conference, December 6-10, 1999 ,Phoenix, Arizona