Introduction to Intelligent Security Systems

Alvin Lin

August 2018 - December 2018

Intrusion Detection and Prevention Systems

Phases of Intrusion:

- Intelligence Gathering: attacker observes the system to determine vulnerabilities
- Planning: attacker decides what resource to attack (usually the least defended component)
- Attack: attacker carries out the plan
- Hiding: attacker covers tracks of attack
- Future Attacks: attack installs backdoors for future entry points

Intrusion detection and prevent has three major parts:

- Intrusion Prevention: protect system resources
- Intrusion Detection: (second line of defense) discriminates between intrusion attempts and regular system usage
- Intrusion Recovery: effective recovery models

Many layers of defense are required because there are often many security flaws in systems. Secure systems are often expensive and not user friendly, and there is still the an insider threat. They need to be continually improved because the skills and tools of hackers are also continually improving.

Terminology

- **Audit**: activity of looking at user/system behavior, its effects, or the collected data
- **Profiling**: looking at users or systems to determine what they usually do
- **Anomaly**: abnormal behavior
- **Misuse**: activity that violates the security policy
- **Outsider**: someone without access rights to the system
- **Insider**: someone with access rights to the system
- **Intrusion**: misuse by outsiders and insiders

IDS Operation

- **Host Based**: deployed on a specific host to monitor and gather information from that host
- **Network Based**: focuses on network attacks and attempts to identify unauthorized, illicit, and/or anomalous behavior based on network traffic patterns
- **Network Behavior Analysis**: an extension of network based intrusion detection systems that inspects the packets and network information gathered from routers and other physical devices on the network
- **Wireless**: examines the wireless network for suspcious activity and analyzes the wireless networking protocols

IDS	Main Components	Security Tasks
Host Based	DMZ Switch, Host IDPS Appliances, Agents	examines data flow and system files related to host
Network Based	Sensors, Load Balancer, Switch, Router	logging data, protocol analysis, inspects data in segments. Signature and anomaly-based detection
Network Behavior Analysis	NBA Management switch, console, and management server	examines packets from network segments
Wireless	sensors, management switches, and management server	sampling of traffic, logging information, wireless protocol analysis

IDS	Advantages	Limitations
Host Based	detects local attacks before they hit the network, bandwidth independent, low false positive rate	delay in detection, agents use the resources of hosts, installation of agents may conflict with existing security software
Network Based	no overload, signature and anomaly-based detection	inability to detect encrypted information, fails to analyze during high load
Network Behavior Analysis	excellent detection capabilities, ability to log information, can often determine attack origin	collects data from network devices, detects attacks after damage occurs
Wireless	able to monitor different types of attacks, can detect physical threat location, can perform prevention techniques	unable to detect passive monitoring or DDoS attacks, cannot do evasion techniques using channel scanning

Knowledge Classification Models

Explicitly Given Models:

- Traditional Search Algorithms
- Traditional Symbolic Logical Reasoning
- Fuzzy Logic Reasoning

Implicitly Derived Models:

- Neural Networks
- Support Vector Machines
- Other Techniques

The term **rule** in AI can be defined as an if-then structure that relates given information or facts in the *if* part to some action in then *then* part. A rule provides some description of how to solve a problem and a relatively easy to create and understand. They are the most commonly used type of knowledge representation. The *if* part is commonly known as the antecedent, while the *then* part is called the consequent.

Entropy

Entropy of a set of examples S relative to a binary classification is:

$$E(S) = -p_1 \log_2(p_1) - p_0 \log_2(p_0)$$

where p_1 is the fraction of positive examples in S and p_0 is the fraction of negatives. If all examples are in one category, entropy is zero. If examples are evenly mixed ($p_0 = p_1 = 0.5$), then entropy is a maximum of 1. For multi-class problems with c categories, entropy generalizes to:

$$E(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

The **information gain** of a feature F is the expected reduction in entropy resulting from splitting on this feature.

$$G(S, F) = E(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} E(S_v)$$

where S_v is the subset of S having value v for feature F. Entropy of each resulting subset is weighted by its relative size.

Artificial Neural Networks

The artificial neuron is a mathematical construct that emulates the more salient function of biological neurons, namely signal integration and threshold firing behavior.

Just as in the biological case, such neurons are bound together by various connection weights that determine how the outputs from one neuron are to be algebraically weighted before arriving at receiving neurons. The intelligence within these collective structures of artificial neurons is stored within these connection weights.

All of the information stored within an artificial neural network takes the form of connection strengths between neurons. These are values by which the signals from one artificial neuron to another are multiplied before being summed up within the receiving neuron. Special programs mathematically change the weights in the net until it consistently yields the correct outputs for any given set of inputs.

Artificial neural networks successively apply all known inputs to the net, propagating signals in the forward direction. They observe network output and backwardly propagate corrections to the respective connections in the net. This process continues until the net yields the correct output for all known test cases. The most important aspect of this process is that the network discovers on its own what the underlying rules of the conceptual space are.

Backpropagation

1. Assign random weights to all the linkages to start the algorithm.
2. Find the activation rate of the hidden nodes using the inputs and the linkages.
3. Find the activation rate of the output nodes using the activation rate of the hidden nodes and linkages.
4. Find the error rate at the output node and recalibrate all the linkages between the hidden nodes and output nodes.
5. Cascade the error down to the hidden nodes using the weights and error found at the output nodes.
6. Repeat the process until the convergence criteria is met.
7. Score the activation rate of the output nodes using the final linkage weights.

Support Vector Machine

A support vector machine tries to build a hyperplane that separates classes in the training dataset. A good hyperplane has the biggest margin.

K-nearest Neighbor

K-nearest neighbor is a classification algorithm that computes the distance to every training example x_i, selecting the k closest instances to the current node and using the majority label to label the current node.

Naive Bayes

Based on the Bayes rule of conditional probability, consider all the attribute values but independently for each other. We can then estimate the probabilities of events based on their frequencies over the training data. The learned hypothesis consists of the set of estimates.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech