Models

Intrusion detection systems determine if actions constitute intrusions on the basis of one or more models of intrusion. A model classifies a sequence of states or actions as good, meaning no intrusions, or bad, meaning possible intrusions. *Anomaly models* use a statistical characterization and actions or states that are statistically unusual are classified as bad. *Misuse models* compare actions or states with sequences known to indicate intrusions and classify those sequences as bad. *Specification-based models* classify states that violate the specifications as bad. The models may be *adaptive models* that alter their behavior on the basis of system states and actions. They may be *static models* that are initialized from collected data and do not change as the system runs.

1. Anomaly Modeling

Anomaly detection has been called the art of looking for unusual states. Anomaly detection uses the assumption that unexpected behavior is evidence of an intrusion. Implicit is the belief that some set of metrics can characterize the expected behavior of a user or a process. Each metric relates a subject and an object.

**Def:** *Anomaly detection* analyzes a set of characteristics of the system and compares their behavior with a set of expected values. It reports when the computed statistics do not match the expected measurements.

Denning has identified 3 statistical models to use in anomaly detection.

A. Threshold metric

- \( \text{min } m \) -- \( \text{max } n \) events are expected to occur
- if outside bound, behavior deemed anomalous
- difficult – different levels of sophistication
  - characteristics of users
  - culture

B. Statistical Moments

- mean \( 1^{\text{st}} \)
- standard \( 2^{\text{nd}} \)
- correlations \( 3^{\text{rd}} \)

- if behavior falls outside of expected intervals \( \rightarrow \) anomalous behavior
- need to change over time
- better estimator but more complex

C. Markov Model

- Establish probabilities of one state transitioning to another state
- Get a sequence of event’s probability
- better yet

- When low probability transition occur, then anomalous behavior
- time-based inductive learning to develop model
- the better the training data, the better the model
II. Misuse Modeling

In the context of intrusion detection, misuse modeling means *rule-based detection*.

**Def:** *Misuse detection* determines whether a sequence of instructions being executed is known to violate the site’s security policy. If so, it reports a potential intrusion.

Modeling of misuse requires knowledge of system vulnerabilities or potential vulnerabilities that attackers attempt to exploit. The intrusion detection system incorporates this knowledge into a rule set. When data is passed to the intrusion detection system, it applies the rule set to the data to determine if any sequences of data match any of the rules. If so, then it reports that a possible intrusion is underway.

Misuse-based intrusion detection systems often use expert systems to analyze the data and apply the rule set. These systems cannot detect attacks that are unknown to the developers of the rule set. More recent intrusion detection systems use adaptive methods involving neural networks and Petri nets to improve their detection abilities.

III. Specification Modeling

Misuse detection is the art of looking for states known to be bad. Specification detection takes the opposite approach. It looks for states known not to be good, and when the system enters such a state, it reports a possible intrusion.

**Def:** *Specification-based detection* determines whether or not a sequence of instructions violates a specification of how a program or system should execute. If so, it reports a potential intrusion.

Only those programs that change the protection state need to be specified and checked. Specification-based intrusion detection is in its infancy. Among it appealing qualities are the formalization of what should happen. This means that intrusions using unknown attacks will be detected. Balanced against this desirable feature is the extra effort needed to locate and analyze the programs that may cause security problems. The subtlety of this last point is brought home when one realizes that any program is a potential security threat when executed by a privileged user.