

## STAT Tool Suite

Sonya Harley  
January 22, 2002

## Intrusion Detection Systems

- Analyzes information about the activities performed in a computer system or network
- Methods to Detect Attacks
  - Anomaly Detection (Behavior-based)
  - Misuse Detection (Knowledge-based)

## Misuse Detection

- Known attacks (signatures) are matched against incoming data looking for evidence of malicious activity
- Advantage
  - Low number of false positives
- Disadvantage
  - Can detect only the attacks that have been modeled

## Anomaly Detection

- Based on models of the "normal" behavior of a computer system
- Advantage
  - Able to detect previously unknown attacks
- Disadvantages
  - Large number of false positives
  - Difficulty training a system in a dynamic environment

## State Transition Analysis Technique (STAT)

- Method to describe computer penetrations as attack scenarios; represented as a sequence of transitions
- Mitigates the disadvantages of plain signature-based approaches
- USTAT – UNIX host-based IDS(Sun); difficult to modify or to extend to match new environments

## State Transition Analysis Technique (STAT) (cont.)

- NSTAT – uses a client-server architecture to collect records from different sources (hosts)
- NetStat – performs monitoring at the network level in real time
- Mechanisms used by the STAT-based tools suggested they could be redesigned as a family of systems

## STAT-based ID framework

- Embodies the domain-independent characteristics of the STAT approach
- Provides a well-defined way to extend the core into a complete intrusion detection system tailored to the domain and environment

## STAT-based ID framework

- STATL – language used to represent attack scenarios using states and transitions
- Off-line architecture to translate attack scenarios into executable modules
- Runtime architecture for the creation of intrusion detection monitors
- The STAT core

## STATL

- Provides constructs to define the domain-independent entities of an attack scenario
- Must be extended by the intrusion detection developer
- Consists of states and transitions
- Provides the built-in concept of timers

## The STAT core


- Supports the preparation of the attack scenarios to be loaded into an IDS
- 4 components
  - Scenario Editor – allows user to construct scenarios in text or GUI format; produces STATL scenarios
  - Parser – reads a STATL scenario and transforms it into an internal format to be used by the Analyzer and the Translator

## The STAT core(cont.)

- Analyzer – reads a scenario in the common internal format; possibly can modify scenario
- Translator – reads a scenario in the common internal format and produces C code which is linked into the runtime architecture

## IDS Application

- The runtime architecture of a complete IDS
- Components
  - Audit stream provider – log files; network sniffer
  - Preprocessor – filters and creates STAT events; these events are sent to the STAT core for processing



## IDS Application(cont.)

- A STAT core – connected to a number of scenario plugin components
- Scenario plugins - contain the executable representation of an attack scenario
- Extension component – gives plugins access to the application-specific characteristics of the processed events and may provide functions to be used in reacting to an attack