Meta-IDS Environments: An Event Message Anomaly Detection Approach

3rd IEEE International Information Assurance Workshop

College Park, Maryland, USA

Jens Tölle, Marko Jahnke, Michael Bussmann, Sven Henkel
{toelle|jahnke|bussmann|henkel}@fgan.de

March 23-24, 2005
Overview

- Introduction
- Basic Ideas
- Structure Recognition
- Application: Worm Detection
- Summary, Further Work
System Overview

Data Flow

Warning Flow

Sensor

Local IDS

Local Security Tool

GW

Console

GW

GW

GW

GWs under domain control

Console(s) under control of trustworthy party

Domain A

Domain B

Domain C

Research Institute for Communication, Information Processing and Ergonomics

Computer Networks
Intrusion Detection Message Exchange Format (IDMEF)
Anomaly Detection Component

Domain A

Local Security Tool

GW

Local IDS

Sensor

Sensor

Domain C

Local Security Tool

GW

Local Security Tool

GW

Local Security Tool

Sensor

Sensor

Sensor

Console

Message Dispatcher

Analyzer

Storage

Control Flow

Data Flow

Warning Flow
Warning

Anomaly Detection approaches may cause
- **false positives** (false alarms)
- and **false negatives** (unreported serious events).

This holds for the system presented here as well.

It is not intended to be a standalone system!

The method supports other intrusion detection approaches.
Basic Idea

Several domains generate different kinds and amounts of event messages.

The basic idea is

not to focus on the meaning of single event messages,

but

to regard all event messages together as a stream of data and check for noticeable behavior.
Event Message Collection

Event message properties may depend on
- number and kind of installed systems
  (usage of different products)
- local configuration of systems
- local information sharing policies
  (anonymization, omission of details, filtering of messages)

Different domains may send event messages differing in form and quantity as a reaction to the same event.

Focus on typical event message structure.
Typical Structure

Typical properties (the structure) of incoming event messages can be visualized using graphs.

Measurements indicate, that these structures stay quite stable during normal system usage.

Fundamental changes are unusual (and therefore an anomaly).
Graph Construction

Most event messages contain information on

- **assumed source** of a security relevant action (IP address)
- **target** of a security relevant action (IP address)
- **affected service** (port)

![Diagram showing source IP, target IP, and port connected to form a resulting graph.](image-url)
Graph Construction cont’d

Real-life event messages led to the following graphs:

Without…

…and with service (port) information.
Graph clustering (grouping of strongly connected nodes, omission of minor edges) reveals basic structure.
Definition: Clustering

Definition:

A Clustering $\mathcal{R}$ of a Graph $G = (V,E)$ is a partitioning into Cluster $C_i$

- $C_i \subseteq V, C_i \neq \emptyset, 0 \leq i \leq n-1$
- $C_0 \cup C_1 \cup \ldots \cup C_{n-1} = V$ and
- $\forall 0 \leq i, j \leq n-1, i \neq j : C_i \cap C_j = \emptyset$

Clustering algorithms which need the number of clusters to be detected as input parameter are not suitable for this purpose.
Detection of abnormal System Behavior

Estimation of distance between current and set of $n$ preceding clusterings:

\[ t_{-n-2} \quad t_{-n-1} \quad t_{-n} \quad \ldots \quad t_{-2} \quad t_{-1} \quad t_{\text{current}} \]

comparison

time $t$
How to measure distance?

To calculate the **distance** between two clusterings, one feasible method is to estimate the minimum number of **elementary operations** needed to convert clustering 1 to clustering 2.

Elementary operations are:

- **splitting** a partition

- **combining** a partition

- (moving a subset from one partition to another)
In case of newly discovered and exploited vulnerabilities, signature databases need some time to be updated. Therefore, misuse detection systems might fail to discover a worm.

The method described here can be used to detect internet worms early and without the help of signatures.

The following slides present an example where real-life data is merged with data of a worm spreading.

- worm behavior comparable to code red v2
- 360000 vulnerable systems on the internet
Application: Worm Detection cont’d

Visualization:

Worm spreading, start… …after one hour…
Application: Worm Detection cont’d

…after two hours and…

…after three hours.
The previous slides gave a visual impression of the spreading of a worm. This graph presents the calculated cluster differences which are used for automatic worm warning.
Summary

- Presentation of an anomaly detection approach based on event message flows
- Building graphs from event messages
- Calculate clustering of graphs to detect typical structures
- Compare consecutive clusterings using distance measures
- Warn administrator if abnormal deviations are detected
Further work

- Optimized automatic determination of reasons for anomalies
- Information extracting for decision and reaction support
- Estimation of influence of anonymized event messages
The End…

Thank you very much for your attention!

Questions?

toelle@fgan.de