Multi agent event monitoring system

Marek Woda, Tomasz Walkowiak
Wroclaw University of Technology, Poland
{marek.woda, tomasz.walkowiak}@pwr.wroc.pl

ABSTRACT

The paper presents the architecture of a hybrid, AI based, event monitoring system. The aim of the system is to assist network administrator in the task of computer intrusion detection. We propose to combine AI techniques (fuzzy logic, data mining, neural networks and/or clustering techniques) to provide efficient intrusion detection system. The event monitoring is designed in the hierarchical agent architecture which augments the effectiveness of processing, filtering and gathering event occurrences. The recommended system is focused on monitoring business services and their components: technical services.

Key Words: Smart sensors, multi-agent event monitoring, AI based event monitoring, systems security

1. Introduction

Event monitoring is increasingly a key part of systems defense. It is so inevitably related to the intrusion detection [1], that in many case it is almost impossible to separate both aspects when considering system security. Various approaches to Event Monitoring are currently being used, but they are relatively ineffective [2]. To improve this, an Artificial Intelligence (AI) is introduced [3, 4] and plays a driving role in evolving event monitoring services[5,6].

Confidentiality, Integrity and Availability of information are major concerns in the development and exploitation of network based computer systems. Event monitoring system which supports intrusion detections, is able to detect, and in subsequent stage with help of intrusion detection prevent and react to the attacks. That's why event monitoring has become an integral part of the information security processes [6]. Worth mention is fact that it is not technically feasible to build a system with no vulnerabilities; event monitoring continues to be an important area of research.

2. Event monitoring

In systems security, event monitoring is a process responsible for harvesting and then scrutinizing collected data against predefined rules. And when the threatening state occurs it signals it to a appropriate subscriber (such like human operator, IDS or other security system). Source of signals come from a variety of sources in both software and hardware, and it may set accordingly to a monitored environment

Event collection is a complex process of gathering events (occurrences of some elemental actions that take place in computer software or hardware) in one place, usually a file. It happens frequently that mentioned above events are pre-filtered to avoid storing to too large files.

Pre-filtering is being done on a predefined set of rules basis; only significant and crucial event occurrences are stored and all other that are less significant are discarded.

Important part of event monitoring process is real time log analysis. This process scrutinizes records in the event log to aggregate them to make decision whether or not an event should trigger an alarm.

An observed object (e.g. an application, an operating system, a database, a hardware component) usually is being monitored by a small software entity called sensor.

Theses entities perform basic operations (collecting, analyze, and signaling) without any co-ordination between themselves or higher level of monitoring.

3. Research Area

Main aim of this paper is to present premises for three tier multi agent monitoring system that will augment effectiveness of processing, filtering and gathering event occurrences.

Such system shall combine AI techniques, be capable to deal with high traffic volume able process and interchange data in a distributed way, in order to reduce number of false positives and allow to signal alarms about threats throughout WAN in a secure way.

Current event monitoring systems are focused rather on a small networks monitoring without any cooperation with other security systems (like for examples SNORT). Lack of cooperation between other systems restricts their usability. Limited applicability to operate in only one environment or operating system is also a huge disadvantage.

The major requirements for a monitoring system are flexibility, modularity and the capability of processing all kind of data from the network in all kinds of ways to produce meaningful information.

Existing event monitoring systems especially commercial ones are based on misuse event detection approach, which means these systems will only be able to detect known event types and in most cases they tend to be ineffective due to various reasons like non-availability of patterns, time consumption for developing new patterns, insufficient data, etc.

We propose applying artificial intelligence methods for the development of EMS (event monitoring systems) that yields some advantages, compared to classical approach.

The AI provides new flexibility to the uncertain problem also in intrusion detection systems and allows much greater complexity for event monitoring systems. However, most of the AI based systems require human experts to refine their response. It is inevitable occurrence. Unfortunately these tasks are time consuming and human dependent.

Distributed sensors nature is a huge advantage, which may facilitate parallel disturbing computing hence the processing of data should be performed within each individual sensor, rather than at a central system controller as in most traditional systems. While a sensor in the traditional sense outputs raw data, described here sensors (smart sensors [7]) outputs only useful information. Furthermore, smart sensors may be dynamically programmed as user requirements change.

Nonetheless if the reaction rules are automatically generated, less time would be consumed for building a good event classifier and shortens the development time of building or updating a new event classifier.

A hybrid (AI based) system should be proposed for aiding network administrator in the task of event monitoring with pre-filtering methods and finally supporting computer intrusion detection.

We would like to combine AI techniques (fuzzy logic, data mining, neural networks and/or clustering techniques) to provide efficient technique for anomaly based, unknown event detection and utilize our approach as aid for host based intrusion detection. Our long term goal is to make this system implement in a real time environment.

We recommend AI based system for event monitoring and countermeasures on computer systems in a network environment using data mining technology. The system would be built using intelligent agents to apply a data

mining approach to also support intrusion detection.

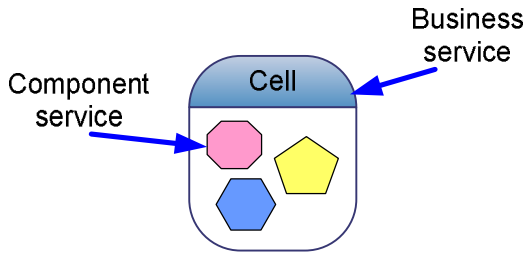

Fig. 1 Idea of a cell

4. Multi agent even monitoring system architecture

We propose to divide entire system that is being monitored into a set of easily controllable regions which will be called cells[1] [8]. It will facilitate monitoring not entire system as a whole (which maybe kind a difficult, resources, synchronization reasons etc.), but divided into small sub-elements.

Cell (Fig. 1.) is a collection of the low level services that make possible rendering business (high level) services. There should be at least one, duplicated, low level service in each cell that has vital meaning for entire system. Such approach would allow us to be services oriented and by the same to divide system for the sake of the business services that are being served.

Each cell should be able to interact with another one. Inner cell services can be used to support other similar cells, in case when one of low level services fails and there is no spare one to backup.

The system environment could be divided into three tiers (Fig 2):

- High level (business services)
- Middle level (component services)
- Low level
    a. operating system (local)
    b. smart sensors (remote)

[1] The idea of „cell" was worked out within DESEREC project [9], firstly proposed in [8]

Inner cell services can be used to support other similar cells, in case when one of low level services fails and there is no spare one to backup.

High level – is the end user (business) services level. Each main service in any system like e.g. internet shop can be perceived as a complex service. All distinguished and recognized, critical, services ought to be monitored for security reasons and with paying attention to their accessibility. None of these tasks can be entrusted to a human operator to monitor. An individual is no reliable enough, is such case, due to elaborateness and monotony of these tasks. So at this level User Virtual Representative agents will be utilized. Their main role will be administering lower tier agents in pursue to detect and recognize the main service unavailability.

Middle tier – is the component ("lower" level – functional, logical, and physical) services level. Each of the available main services ("internet shop") comprise of several component, usually more specific (lower level) services like e.g. DNS or data base services or physical like physical network connection.

We have differentiated two types of agents:

- cell agents,
- mobile agents (communication agents).

Cell agents are ones that are ascribed to one cell location or single component service in one cell and can't be recalled or relocated from the initial position. Cell agent assignment is either to entire cell or one, particular, low level service, and is strictly dependent of cell complexity (or rather business services). For the cells that are not business critical cell agents are rather not advisable to use (if our environment resource consuming aware) and in that case it's strongly recommended to use mobile agents.

Mobile agents work as communication agents, that mean most of the information exchange is being done by them or they facilitate it. Agent mobility feature comes form its ability to transport its state from one environment to another, with its data intact, and still being able to perform appropriately in the new environment. It infrequently happens that environment which agent operates became hostile or no longer handy to work in, then agent could move to another cell and restore its activity.
Mobile agents decide when and where to move next, which is evolved from RPC. Agent move like a common user does, namely it doesn't really visit a website but only make a copy of it, a mobile agent accomplishes this move through data duplication. When a mobile agent decides to move, it saves its own state and transports this saved state to next host and resume execution from the saved state.

Mobile agents perform two roles:

- cell administrators, where are able to manage entire (rather simple, not complex) cells, act in such case like cell agents, but more that one low-level service can be entrusted to them, only for cells without crucial meaning for entire system,
- the carriers of critical data, they move from one cell to another with information about low(-er) services availabilities, malfunctions, unexpected events, and passing data about false alarms, in order not to alarm entire system when similar action occurs (the receivers that receive reconfiguration actions from Policy & Reaction Server for low-level services).

Low level (OS level) – this is the lowest level, mainly oriented on interactions at kernel API (local approach). All processes, their operations, and also interactions between them are monitored at the root. Monitoring is directly pinned to the operating system.

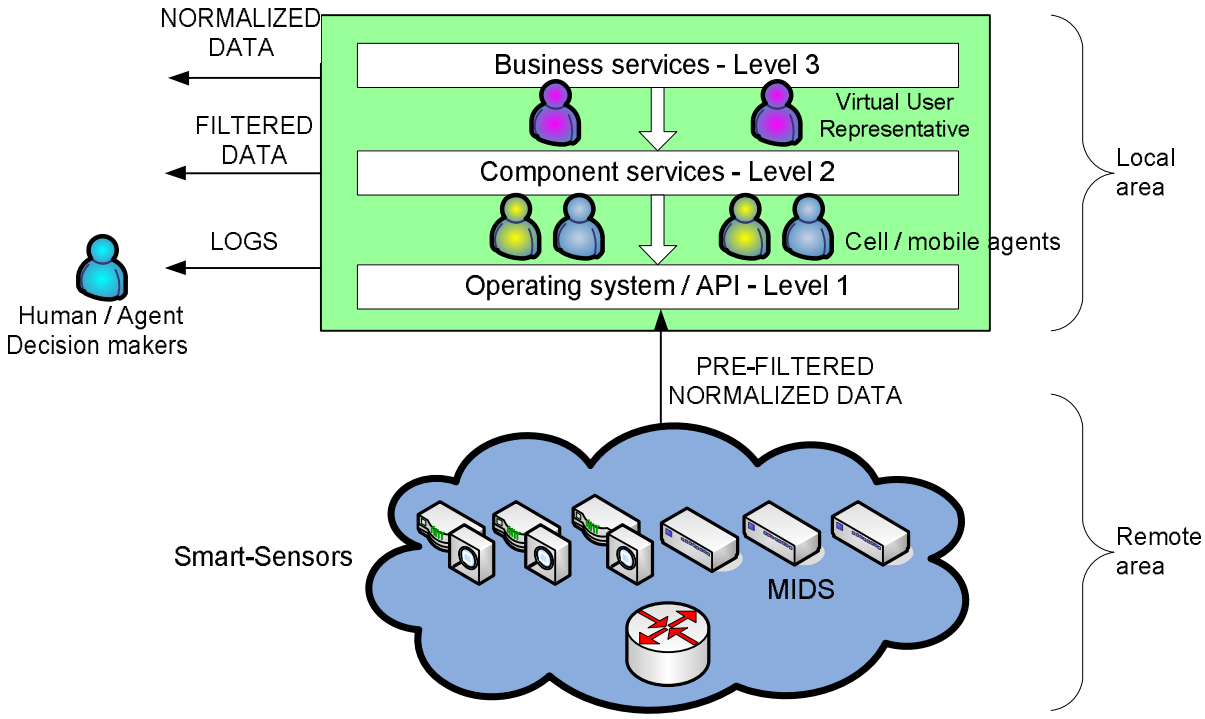

Fig. 2. General overview of multi agent event monitoring system

This approach is somewhat restricted because of the direct connection to operating system core and by that limited

to the implementation of specific OS (UNIX, Windows).

Nonetheless if properly implemented this might be best way to raise non-false alarms about malicious actions due to the fact that at this level "data noise" is the smallest and sensors are not disturbed (deafened) by the set of the working (higher level) applications.

Remote approach of smart sensors encompasses sending preconfigured smart sensor to the root source of events usually located far from system core (it could be a network node, or remote server etc.).

All events are being monitored by the smart sensors that are data-harvest oriented, mainly responsible for threat detection and filtering. Smart sensors will be equipped in AI techniques but only in vestigial form. These sensors should remain as small as possible, not to bring additional, unnecessary burden for the operating system.

High level agents called User Virtual Representatives (Fig. 3) function as substitutes of users, acts like human users, perform regular human actions in pursue to detect service unavailability. One UVR agent is ascribed always only to one business service. When service, which agent is ascribed to, is no longer responding, and UVR agent give can't itself recognize the culprit, it gives commands to lover tier agents in order to recognize the situation, which of component services.

Middle level agents will gather data, render complex event logs and activity data into common formats (normalized data) while low-level agents called smart sensors classify recent activities and provide data and current classification states to each other and to a higher level of agents that implement data mining over the entire knowledge and data sources of the system.

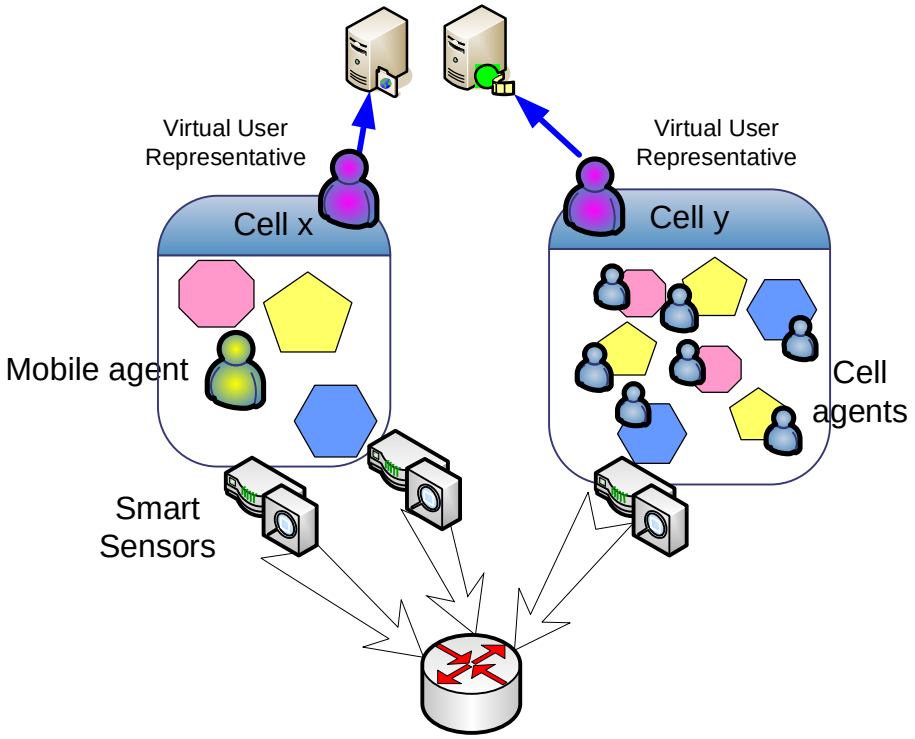

Fig. 3. High level multi-agent event monitoring system architecture

5. Conclusions

The implementation of a practical smart sensor event monitoring system requires the synthesis of several technologies. One must bring together knowledge in the fields of artificial intelligence, data processing, distributed systems, and networks. While extensive research has been conducted in all of these areas, a smart sensor system imposes some new design parameters which must be met.

Classical event monitoring approach is no longer applicable in modern networks. It lacks of flexibility and works effectively only in small nets. Our Agent based event monitoring system has additional value the current solutions:

1. Combines WMI & SNMP mechanisms along pre-filtering mechanisms that allows to collect only real hazardous events
2. Suspicious events can be freely distributed between similar agent based systems due to their independent communication layer
3. Security policies / authentication methods incorporated in the agent based systems prevent misuse or distort the information held by agents
4. AI based system provide more accurate recognition of events
5. Pre-filtering mechanisms could prevent false alarms
6. Distributed system can easily analyse high traffic volume
7. Agents mobility facilitates possibility to pass events information to the generally not accessible (sub)nets
8. Smart sensors located on the user's host machines can diagnose conflicting services, hardware malfunctions and prevent user's misuse actions.

Feature	Regular system	Agent based system
Ability to process high traffic volume	Limited	Capable (distributed computing)
Unknown events recognition	Not possible	Possible (AI based)
Data can be compromised	High possibility	Almost impossible (high security, data encryption)
False positive alarms	Often spotted	Seldom (pre-filtering mechanisms)
Info about threats distributed to other systems	Only in a limited area (locally)	Across the network thru agents / sensors (globally)
Modular structure*	No	Modular structure (Replaceable IN/OUT module)
Combined mechanisms WMI / SNMP	No	YES

Tab. 1. Agent based event monitoring system vs. regular event monitoring system.

5. Further work

Currently we have implemented and tested several smart sensors that can interact with higher agents layer, are able to perform pre-filtering on a gathered events and work in Windows / UNIX environments.

We strive to create a complete distributed event monitoring solution that will work in any network environment without human interaction – communicate thru encrypted channel, and facilitate unknown event recognition.

Acknowledgement

Work reported in this paper was sponsored by a EU grant DESEREC IST-2004-026600, (years: 2006-2008) "Dependability and Security by Enhanced Reconfigurability".

References

[1] Bace, R.G: Intrusion Detection. Technical Publishing (ISBN 1-57870-185-6)

[2] Idris, N.B., Shanmugam, B.: Artificial Intelligence Techniques Applied to Intrusion Detection. IEEE Indicon 2005 Conference, India, Chennai (2005)

[3] Garcia, R.C. Copeland, J.A.: Soft Computing Tools to Detect and Characterize Anomalous Network Behaviour, IEEE World Congress (2000) 475-478

[4] J. E. Dickerson, J. Juslin, J. A. Dickerson and O. Koukousoula, " Fuzzy Intrusion Detection", in the proceedings of North American Fuzzy Information Processing Society 2001 (NAFIPS 2001) , Vancouver, Canada, July 25th, 2001

[5] Staniford-Chen, S., Tung, B., and Schnackenberg, D. "The Common Intrusion Detection Framework (CIDF)". Information Survivability Workshop, Orlando FL, October 1998.

[6] Chenxi Wang, Knight J.C. 2000: Towards survivable intrusion detection. Third Information Survivability Workshop -- ISW-2000, October 24-26, 2000 (www.cert.org/research/isw/isw2000/papers/38.pdf)

[7] Naqvi, S. Riguidel, M. "Security and trust assurances for smart environments". In: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 7-10 Nov. 2005, pp. 8

[8] IV Framework EU sponsored project: DESEREC "Dependability and Security by Enhanced Reconfigurability" http://www.deserec.eu

[9] Goubard, J.E., THALES Communication "WP3 WP4 illustrative concepts", DESEREC Project Meeting, April, 5-6, 2006, Madrid (project internal materials)