Abstract: The purpose of this paper is to present the design of a Hybrid network security system based on distributed multi-agent approach. This system combines NIDPS, HIDPS, KIDPS and firewall using a semantic distribution through detection method and different security levels. The aim is to minimize the analysis loading so that we can achieve the following goals: to improve the response time, to decrease the number of false positives and false negatives, to ensure interoperability between detection system and prevention system, to reduce blocking harmless traffic and to clean traces of detection and prevention.

Keywords: Security, IDPS, Real Time, Multi-Agent, Distributed System

1. INTRODUCTION

Among the major computer security features, we find the firewall systems. However, they do not protect the confidentiality of data. For this purpose, we require the use of cryptographic algorithms to ensure the confidentiality of exchanges and customers. Moreover, a service based on the IP address to identify its customers can easily be a victim of IP spoofing or ARP spoofing. As a solution to these cases, we opt for IDPS (Intrusion Detection and Prevention Systems). IDPS has also the ability to take measures and to detect incidents according to a security policy. However, setting an IDPS is very delicate since a false alarm may cost time and money.

This research topic focuses on securing computer networks by using an architecture based on distributed multi-agents system.

2. THE STATE OF THE ART

In this section we are going to focus on introducing the concept of IDPS, agent and distributed system. These 3 points will be the core of our conception development.

2.1 IDPS

2.1.1 Definition

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. [3]

An intrusion detection system (IDS) is a software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. [3]

Thus, an IDPS focuses on the following:

- Identification of the possible incidents.
- Information of access of these incidents.
- The blocking of these incidents.
- Establishment of the log of these incidents to the security administrators. [3]

Moreover, the IDPS can be exploited for other purposes, such as:

- Identifying security policy problems.
- Documenting the existing threat to an organization.
- Deterring individuals from violating security policies. [3]

There are two types of IDPS: NIDPS (N: Network) that analyzes network traffic and HIDPS (H: Host) that can detect intrusions that have not been detected on the network and have achieved the target machine. We can also find KIDPS (K: Kernel) which are a special case of HIDPS; they are interfaced closely with the host operating system in order to inspect all system calls made on the machine that must be monitored and their calls arguments before executing them.

The IDPSs became a necessary complement to the security infrastructure of each company. They record information about observed events, notify security administrators of the most important of these events, and produce reports. They can also evolve in the security environment.

2.1.2 Common Detection methods

- Signature Based Detection

Y.SENHAJI

Architecture System Team - LISER
Hassan II University- AinChock ENSEM
Casablanc, Morocco
A signature is a model which corresponds to a known threat. This method is based on the comparison of the units of activities (Package, Log Entry) to the list of these signatures by using the operators of comparison. However, this method represents two disadvantages:

1. It can’t make the link between the request and its response.
2. It does not remember the attacks.

   - Anomaly Based Detection

This method is based on the comparison between the events and the definition of the events considered normal to detect deviations. In this case, the IDPS has a “Profile” which represents the normal behavior. Examples: lists of the users, hosts, connections networks… It is a method basing itself on statistical calculations (Ex: number of Emails sent by a user, number of tests of erroneous logins). So the second disadvantage of the first method is avoided, because the system is fixed with a training phase for generating the profile. This profile could still be regenerated after another measurement of the system. However, if during the generation or regeneration of the profile, the system includes harmful activities of which the rate of change is very small, these turn up to be an integral part of the profile.

   - Stateful Protocol Analysis

This step of analysis is based on the comparison between the protocols and their profiles. In addition, it exploits the combination of the request and its answer to be able to evaluate the state which constituted a weak point of the first method (Signature based detection).

2.1.3 Examples of IDPS

We can find on the market several examples of IDPS. For example:

   (a) SNORT

Snort can be configured to run in three modes:

   - Sniffer mode, which simply reads the packets off of the network and displays them in a continuous stream on the console (screen). [1]
   - Packet Logger mode, which logs the packets to disk. [1]
   - Network Intrusion Detection System (NIDS) mode: the most complex and configurable configuration, which allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees. [1]

(b) SURICATA

Suricata is a multi-threaded intrusion detection and prevention engine. To describe Suricata as multi-threaded means that it can run on a machine with multiple CPUs, using threads to spread the workload between the CPUs and process many packets and streams at one time.[2]

(c) PRELUDE

Prelude is a Hybrid IDS framework. In this context, hybrid means that Host agent data is combined with network information to form a comprehensive view of the network. [4]

2.2 Distributed System

A Distributed system can be distributed based on an existing conceptual distance between its components.

This distance can be:

   - Spatial: distribution by different processes assigned to solve a problem related to space.
   - Semantic: distribution by the specificity of knowledge and a particular know-how.
   - Structural: representations are heterogeneous and reasoning mechanisms are different.
   - Functional: according to its function and its role within the system.

2.3 Multi-Agent System [5]

A multi-agent system is a set of agents interacting with each other. An agent is a physical or abstract entity with a base of knowledge. It can be:

   - Reactive: having a capacity stimulus-response.
   - Cognitive: based on knowledge and the partial representation of the environment.
   - Intentional: having goals, intentions and explicit beliefs.
   - Rational: justifying its decisions and illustrating its achievements according to the rules and the applied methods.
   - Adaptive: having the capacity to enrich its knowledge data-base according to the environmental constraints.
   - Communicative: with a full communication and reasoning capability related to a field of activity.

3. PROBLEMATIC

To detect and eliminate attacks (security threats), a system must have tools to monitor data in transit. Thus, to analyze all network traffic seems to be the most ideal solution. However, this is an unthinkable alternative because the quantity of data will be very huge and analysis time will be very long (real-time problem). Moreover, even with focusing only on a part of the traffic,
load analyzis remains important and this analyzis must be in a real-time to serve the rapid availability of information. So the question is how to design a security system that can reduce this loading?

But at this stage, the stakes are always critical. The risk that the security system generates a false alarm (false positive) or it does not detect a real intrusion (false negative) still exists. So how can we reduce this probability?

Once the detection phase has been processed, we begin to wonder whether or not our system is protected. Detection only generates alerts and saves LOG files that administrators sometimes do not have time to analyze. Thus, adding prevention to detection could be a good alternative. The Security system becomes active with new features allowing not only the detection of threats but also, for example, blocking them. However, and in order to make the evolution possible, the security system should be designed so as to enable interoperability between its various components. Thus, the outputs of a process will match with the inputs of the other and vice versa and that in a systematic manner and without any external intervention or additional modules.

Now, assuming that the issue of interoperability has been set, it is well known that the detection is not 100% reliable. Therefore, we may block harmless traffic. In addition, and in the case of an IP spoofing, if the IP address of an important node in the network is spoofed (router, web service ...) then the service connected to this node will no longer be available. On the other hand, blocking traffic is a sign of the presence of a prevention system. The traces of the prevention system can be used against it to go in an unseen future attack.

In summary, we may retain the following points:

1. Reducing the load of analyzis
2. Reducing the number of false positives and false negatives
3. Resolving the interoperability problem between the detection system and the prevention system
4. Reducing the cases of blocking harmless traffic
5. Eliminating the traces of the prevention system.

It should be noted that we are conducting an academic research, so the solution must be a free and open source. Thus, and in order to respond to these points above, we will try to present in the following section a proposition of an architecture based on the concepts introduced in the state of the art section: Distributed system based on multi agents.

4. PROPOSED ARCHITECTURE

4.1 Introduction

Prior to deployment of the security solution, we assume that users are aware of the importance of security and its challenges and that all systems and applications are constantly updated (security patches).

Suppose we have a network with the following elements:

- A LAN (local area network): consists of several workstations.
- A DMZ (demilitarized zone): Consisting of machines on the internal network that need to be accessible from the outside (mail server, FTP server, web server ...)
- A Web Client: consists of Outside Network

4.2 Spatial Distribution

To secure the network while focusing on the concept of load reduction (Item No. 1 of the problematic) and increased response time, we will opt for the previously defined concept namely "the notion of distributed system". The security system will be deployed and distributed spatially in the network.

It will be composed of several distributed IDPS sensors. And for a more reduction of the data loading on these sensors, they must be accompanied by pre-filtering firewalls which analyze the data stream before capture. Moreover, and for a complementary security solution we will combine between NIDPS and HIDPS.

HIDPS will be deployed on the machines in the DMZ and on important servers. We can also add KIDPS (K: Kernel) for sensitive machines. KIDPS slows down the execution of programs, but they are the only ones which can effectively detect attempts Buffer Over flow.

Thus, we can see that we have performed a distributed load analyzis between: NIDPS, HIDPS, KIDPS and Firewall.

Note that in the hardware point of view, our system must be equipped with Hubs and not with Simple Switches. Hubs allow NIDPS to sniff the traffic (traffic capture). A switch switches packets directly to the recipient except in the case of switches with a SPAN port (Switch Port
Analyzer). Moreover, we suppose having unencrypted streams. For example, with SSL stream we must use SSL Proxy equipment.

Thus, our security architecture will be as follow:

![Figure 2: Network Security Platform Architecture](image)

**Sensors description:**

- **Ks:** KIDPS sensor for sensitive servers
- **Hs:** HIDPS sensor for important servers
- **N1:** NIDPS sensor analyzing traffic between the internal network and the Internet
- **N2:** NIDPS sensor analyzing traffic between the internal network or DMZ and Internet (before the firewall for its protection)
- **N3:** NIDPS sensor analyzing traffic between the elements of the DMZ and Internet
- **Hi:** sensor for HIDPS servers in the DMZ

### 4.3 Semantic Distribution

At this stage we have spatially distributed security architecture in specific locations of our network. It consists of IDPS (N, H and K) and a firewall. Now we may proceed to a second distribution: a Semantic distribution based on IDPS method detection. This distinction aims to specialise the IDPS while specialization causes reducing of the number of false positives and false negatives (Item No. 2 of the problematic).

Thus, each IDPS will be divided into 3 sub IDPS (we will call IDPSS):

- **IDPS-SPA:** Based on the "Stateful Protocol Analyzis" as a method of detection
- **IDPS-ABD:** Based on "Anomaly Based Detection" as a method of detection
- **IDPS-SBD:** Based on "Signature Based Detection" as a method of detection.

Thus, each IDPSS will focus on a detection method to analyze the system.

![Figure 3: IDPS Semantic distribution](image)

Moreover, we split each IDPSS into elementary agents:

![Figure 4: IDPSS Agents Structure](image)

<table>
<thead>
<tr>
<th><strong>Table 1: Capture Agent Structure</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Components</strong></td>
</tr>
<tr>
<td>Data Filter</td>
</tr>
<tr>
<td>Data Decoder</td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>Update</td>
</tr>
</tbody>
</table>

**Capture Agent:** catches and decodes information flow.

**Security Agent:** Manages and collects Security rules.
Table 2: Security Agent Structure

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Defines the security level (this will be detailed later)</td>
</tr>
<tr>
<td>Rules</td>
<td>Stores safety rules</td>
</tr>
<tr>
<td>Update</td>
<td>Updates interface and data structure</td>
</tr>
</tbody>
</table>

Analyzer Agent: Analyzes the captured data with the data from the security agent.

Table 3: Analyzer Agent Structure

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyzer</td>
<td>Manages analysis</td>
</tr>
<tr>
<td>Communicator</td>
<td>Manages the communication with other agents (capture, security and prevention), logs and other IDPSS</td>
</tr>
<tr>
<td>Historic</td>
<td>Manages the history of detection</td>
</tr>
<tr>
<td>Update</td>
<td>Updates interface and data structure</td>
</tr>
</tbody>
</table>

Prevention Agent: Decides action to be implemented during the detection of threats.

Table 4: Prevention Agent Structure

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Lists preventive actions</td>
</tr>
<tr>
<td>White list</td>
<td>Sets the white list (this will be detailed later)</td>
</tr>
<tr>
<td>Standard</td>
<td>Sets the list of standards (this point will be detailed later)</td>
</tr>
<tr>
<td>Update</td>
<td>Updates interface and data structure</td>
</tr>
</tbody>
</table>

4.3.1 Security Level

As you know, even by performing such a spatial and semantic distribution, it is unthinkable to analyze all traffic. It is therefore necessary to prioritize risk systems. The risk systems are those which provide services accessible via the Internet (HTTP, FTP ...). Thus, we introduce the concept of the security level of the security agent.

The analysis is performed initially using data from the security agent “level 1” and the result of the analysis is moved to the prevention agent for action. However, a further analysis is performed in the background (less critical) and whose treatment does not block traffic because its result is not directed to the prevention agent but to LOG files and for updating the history of the analyzer. Thus, an agent can evolve from level 2 to level 1 agent in terms of its history.

4.3.2 The Communicator Component

Among the important points that have been incorporated into the structure of analyzer agent is the Communicator field. This allows to:

- With Capture agent: Transfer data for analysis
- With Security agent: recover security rules and update the security level
- With Prevention agent: transfer the results of the analysis to action. This role solves the major problem being raised in the issue of interoperability between IDS and IPS (Item No. 3 of the problematic)
- With Administration Log: Update in detail all the events.
- With other IDPSS: Identify threats detected and avoid duplication.

4.3.3 White List

As was mentioned previously in the problematic (Item No. 4), it is important to minimize the cases of blocking of harmless traffic by the preventive agent. So we introduce the notion of white list as a field in the prevention agent structure. This will include the list of network addresses that should under no circumstances have been blocked by the prevention agent. So even usurping one of these addresses, a hacker will not cause blocking important services of our network.

4.3.4 The Standard Component

Detecting the presence of an IDPS by an attacker is dangerous. It will try to gather as much information about it to be able to override it discreetly. When the prevention agent responds to an attack (logs off, blocks a port ...) it must use the standard of the operating system in which it operates so that it will not let any traces (packet header) which will help its identification (point No. 5 of the problematic).

Moreover, NIDS sets interface capture in a promiscuous mode. Thus, by sending an "ECHO REQUEST" with a
non-existent MAC address, if the machine responds, then the attacker can detect the presence of the IDS. However, we can use the IEEE standards to detect spoofed MAC addresses. A hardware manufacturer wishing to produce network cards needs to obtain a three-byte organizationally unique identifier from the Institute of Electrical and Electronics Engineers to be used as a prefix for the MAC addresses of their products. This identity string permits a manufacturer to maintain their own allocation procedure for MAC addresses, ensuring they are globally unique. At this time, the IEEE has allocated 6,278 prefixes to various organizations (IEEE, 2002). The IEEE makes the list of prefix allocations and the assigned company information available to the public, largely for users to match a piece of equipment with a MAC address to its manufacturer. We can use this list to evaluate all MAC address sources on the network to determine if the prefix is allocated by the IEEE. MAC addresses that appear on the network using a prefix as yet unallocated by the IEEE can be flagged as anomalous activity. [6]

5. ILLUSTRATION PLATFORM

5.2 Generalities

To illustrate the proposed architecture, we have developed C++ software which is based on the WINPCAP tool. WinPcap is an OPEN SOURCE library that allows the capture and the analyzes of packet in Win 32 platforms. The choice of the operating system of test, in this case, is not important because the capture is made on rough packages directly from the network interface. The WinPcap library does not use, to reach the network, the primitives of the operating system such as for example the sockets. The figure (Figure 6) below illustrates the position of the Capture Agent which sniffs the packages in the core between the network interface and TCP/IP interface.

5.3 Interface of the illustrating Platform

The Illustrating software is a graphical application which arises as shown in the figure below (see Figure 7).

By clicking on the button “initialize” the application detects all network interfaces of the workstation (wired or wireless) and loads them in a drop-down list. By choosing an interface in this list, its information such as name, type, IP address etc., are displayed in a text box in the upper part of the application window (see Figure 7). Clicking the Capture button starts the process of listening to the network and preparing the data for other agents. Figure 8 shows an example of capture on a Wireless interface.

6. CONCLUSION AND FURTHER WORK

In this paper, we proposed hybrid security architecture based on a distributed multi-agent approach. The latter is a combination of NIDPS, HIDPS, KIDPS and firewall according to spatial and semantic distributions based on detection method and security levels. The aim is to reduce the analyzes loading to improve response time, to reduce the number of false positives and false negatives, to ensure interoperability between the detection system and the prevention one, to reduce the
number of harmless blocking traffic and to clean traces of detection and prevention. As further work we can detail more the securing of IDPS and the securing of their LOG files by developing a network management controlled by its own firewall. We can also study an additional distribution by protocol “structural distribution” (level 4: Web streaming, FTP streaming, SQL query ...). Moreover, we can think through optimizing safety rules (ex: if the network is running on Windows systems, the rules for UNIX systems are not needed). And at last, we can develop in detail the interface of illustration.

References


AUTHOR

Youssef SENHAJI received his Degree in engineering from the ENSAM, Meknès, Morocco. In 2009, he joined the System Architecture Team of the ENSEM School, Casablanca, Morocco. His current main research interests IDPS in a Distributed Multi-agents Systems.