International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 9 – No.5, August 2015 – www.ijais.org

Network Security: Hybrid IDPS

Youssef Senhaji
Architecture System Team
Hassan II University of Casablanca ENSEM
Casablanca, Morocco

Hicham Medromi
Architecture System Team
Hassan II University of Casablanca ENSEM
Casablanca, Morocco

ABSTRACT
This paper deals with the issue of computer security, which aims to develop a robust and independent security architecture. This architecture consists of several probes spatially distributed to several locations in the network (sensitive servers, DMZ, workstations, etc.). These probes are NIDPS, HIDPS, KIDPS and Arduino Yun Board. These same probes were semantically distributed according to three threat detection methods. At the end of this paper, we developed a hybrid system consisting of a software IDPS represented by a probe developed under Visual C ++ and an embedded solution developed under Python in an Arduino YUN board. We carry out a series of computer attacks on our detection system to assess its response time.

General Terms
Network Security, IDPS, Real Time, Embedded System, Distributed System, Arduino.

Keywords
Network Security, IDPS, Real Time, Embedded System, Distributed System, Arduino.

1. INTRODUCTION
IDPS are important computer network security system.

In this paper we will present a combination of two IDPS configuration. The first configuration is a software solution developed with Visual C++.

The second configuration is a hardware proposal embedded in an Arduino Yun board.

On these systems, we will make several computer attacks to see their reactions.

But before we begin, we'll introduce the concepts: detection method and distributed system and then we'll present the Arduino Yun Board.

2. COMMON DETECTION METHODS
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. [1]

Among the detection methods used by IDPS, we find:

- Signature Based Detection: this method is based on the comparison of the units of activities (Package, Log Entry) to a list of models by using the operators of comparison. A model corresponds to a known threat.
- Anomaly Based Detection: It is a method basing itself on statistical calculations and it has a "Profile" which represents the normal behavior. So this method consists of making comparison between the events and the definition of the events considered normal to detect deviations.
- Stateful Protocol Analysis: This method compares the protocols and their profiles. In addition, it exploits the combination of the request and its answer to be able to evaluate the state.

3. 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.
- Semantic: according to its function and its role within the system.

4. ARDUINO YUN BOARD
The Arduino Yun is an electronic board that uses the Atmel processor ATmega32U4. Besides of that, it has an additional processor: Atheros AR9331, that turn the Linux distribution OpenWrt Linino.

Fig 1: Arduino Yun Board

5. PROPOSED ARCHITECTURE
5.1 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

5.2 Spatial Distribution

To secure the network while focusing on the concept of load reduction and increased response time, the security system will be deployed and distributed spatially in the network. It will be composed of several distributed software IDPS (hereinafter referred IDPS) and hardware embedded Arduino IDPS sensors (hereinafter referred ARD). 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. Below the list of probes that we will use:

- 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
- ARD : Network Arduino sensor

5.3 Semantic Distribution

In this step, we proceed to a second distribution, a semantic one based on IDPS method detection. This distinction aims to specialize the IDPS.

Thus, each IDPS and ARD will be divided into three parts:

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

6. TEST RESULTS FOR THE HYBRID SYSTEM: IDPS/ ARD

To achieve our simulation on our system, we have developed 3 Systems:

The first is an application developed with C ++ making the role of an IDPS exploiting the PCAP library.

The second is a Python script embedded in a Yun Arduino board and doing the role of an IDS by exploiting RAW socket.

The third system is an application that generates targeted intrusion attacks.

Thus, we will initially attack a system protected by the binomial HIDPSS and ARD and secondly the case of a system protected by the binomial NIDPSS and ARD.

6.1 HIDPS/ARD System

6.1.1 Diagram of the simulation

As a first step, we will pair an HIDPSS and an ARD as below:

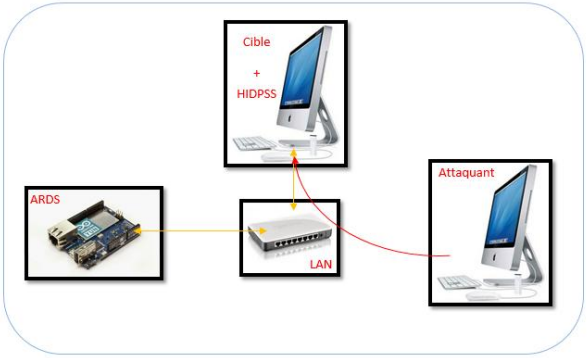

Fig 2: Case HIDPS/ARDS

6.1.2 Evaluation of the detection time

We carry out a series of attacks on our detection system to assess its response time to an attack. Thus we get the results below.

Table 1. Summary of different detection time - HIDPS / ARD

Attack Number	Attack Instant	ARD Detection Instant	Detection Time ARD (ms)	IDPS Detection Instant	Detection Time IDPS (ms)
1	18:11:06,455	18:11:11,317	0:00:04,862	18:11:08,004	0:00:01,549
2	18:11:27,000	18:11:35,347	0:00:08,347	18:11:28,300	0:00:01,300
3	18:11:40,699	18:11:46,134	0:00:05,435	18:11:43,011	0:00:02,312
4	18:12:03,000	18:12:13,877	0:00:10,877	18:12:05,350	0:00:02,350
5	18:12:15,613	Not Detected		18:12:17,518	0:00:01,905
6	18:12:32,073	18:12:42,245	0:00:10,172	18:12:33,758	0:00:01,685
7	18:12:42,447	Not Detected		18:12:43,882	0:00:01,435
8	18:12:51,698	Not Detected		18:12:54,022	0:00:02,324
9	18:13:02,571	Not Detected		18:13:04,162	0:00:01,591
10	18:13:11,650	18:13:14,057	0:00:02,407	18:13:13,288	0:00:01,638
11	18:13:25,848	Not Detected		18:13:27,484	0:00:01,636
12	18:13:36,830	18:13:44,440	0:00:07,610	18:13:38,638	0:00:01,808
13	18:13:55,550	Not Detected		18:13:57,967	0:00:02,417
14	18:14:08,577	18:14:23,450	0:00:14,873	18:14:10,946	0:00:02,369
15	18:14:25,003	Not Detected		18:14:26,593	0:00:01,590

Average	0:00:08,073	Average	0:00:01,861
Min	0:00:02,407	Min	0:00:01,300
Max	0:00:14,873	Max	0:00:02,417
Detection rate	53,33%	Detection rate	100,00%

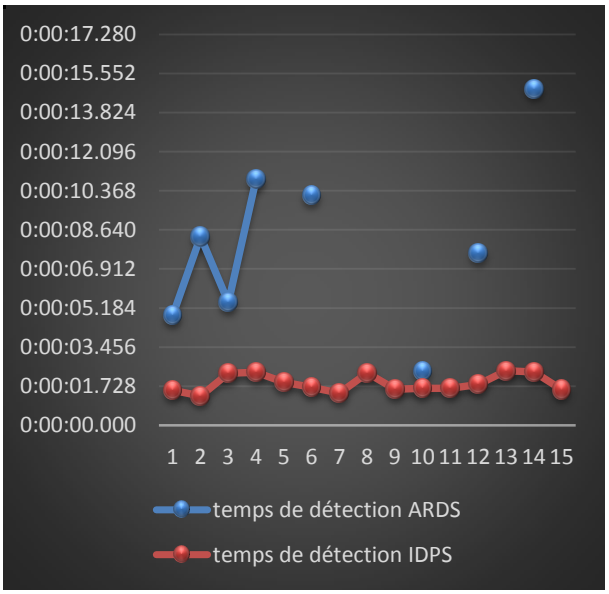

Fig 3: Evolution of the detection time of an attack – HIDPS/ARD

Of course, this detection time may vary depending on:

- The physical characteristics of our simulation system workstations, network cards, Switch ...
- Network saturation at the time of the attack
- The number of attacks
- The duration between attacks
- The number and nature of security rules
- etc.

But, nevertheless, we note that:

- The threat detection rate HIDPS is 100% at the time the ARD is only 53.3%
- The detection time of the HIDPSS is significantly better than that of ARD

Thus, we discover that an embedded system is not in all cases the fastest system. But it depends of security purposes.

6.2 NIDPS/ARD System

6.2.1 Diagram of the simulation

In this case we pair an NIDPSS with an ARD as below:

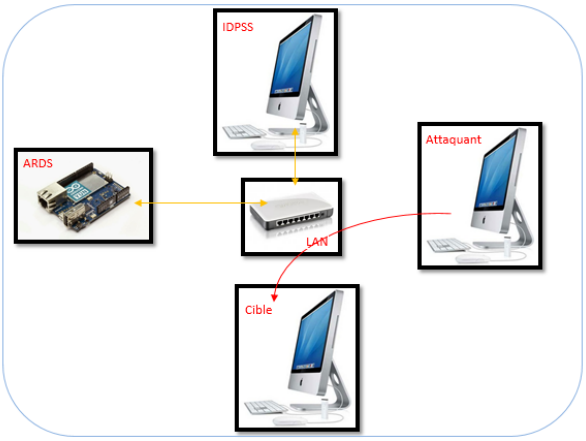

Fig 4: Case NIDPS/ARD

6.2.2 Evaluation of the detection time

We carry out a series of attacks on our detection system to assess its response time to an attack. Thus we get the results below.

Table 2. Summary of different detection time – NIDPS / ARD

Attack Number	Attack Instant	ARD Detection Instant	Detection Time ARD (ms)	IDPS Detection Instant	Detection Time IDPS (ms)
1	18:19:42,000	18:19:52,126	0:00:10,126	18:19:52,983	0:00:10,983
2	18:20:04,884	18:20:25,297	0:00:20,413	18:20:26,446	0:00:21,562
3	18:20:36,928	Not Detected		Not Detected	
4	18:21:04,352	18:21:08,428	0:00:04,076	18:21:10,346	0:00:05,994
5	18:21:23,728	Not Detected		Not Detected	
6	18:21:41,809	18:21:46,160	0:00:04,351	18:21:47,053	0:00:05,244
7	18:22:04,226	18:22:04,624	0:00:00,398	18:22:04,390	0:00:00,164
8	18:22:22,634	Not Detected		Not Detected	
9	18:22:41,651	18:22:46,293	0:00:04,642	18:22:48,025	0:00:06,374
10	18:23:02,000	18:23:03,215	0:00:01,215	18:23:02,221	0:00:00,221
11	18:23:21,000	18:23:33,712	0:00:12,712	18:23:34,670	0:00:13,670
12	18:23:35,674	18:23:49,072	0:00:13,398	18:23:50,895	0:00:15,221
13	18:23:56,002	18:24:03,415	0:00:07,413	18:24:05,091	0:00:09,089
14	18:24:11,773	18:24:15,752	0:00:03,979	18:24:17,276	0:00:05,503
15	18:24:25,143	18:24:30,162	0:00:05,019	18:24:31,456	0:00:06,313
		Average	0:00:07,312	Average	0:00:08,361
		Min	0:00:00,398	Min	0:00:00,164
		Max	0:00:20,413	Max	0:00:21,562
		Detection rate	80,00%	Detection rate	80,00%

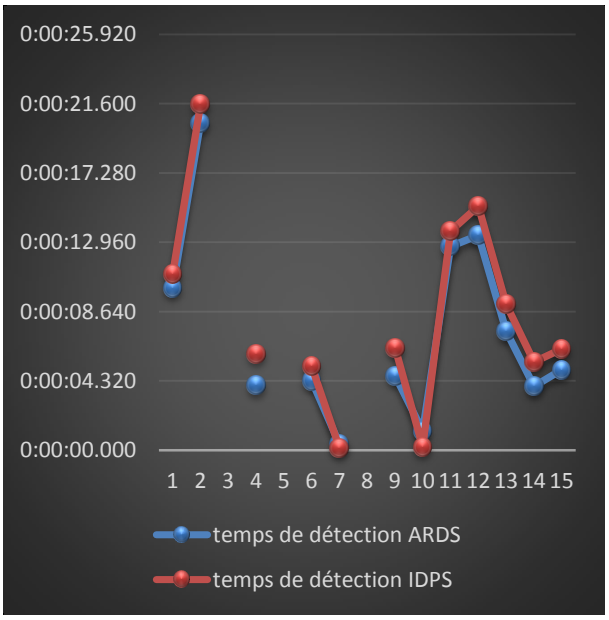

Fig 5: Evolution of the detection time of an attack – NIDPSS/ARD-S

Of course, this detection time may vary according to the same conditions mentioned in the previous section.

But, nevertheless, we note that:

- The detection rates of ARD and NIDPS are not 100%
- The ARD detection time is on average faster than the NIDPSS

Thus, we can notice that unlike the previous case, the embedded system has better performance.

7. CONCLUSION AND FURTHER WORK

In this paper, we proposed hybrid security architecture based on a distributed approach of NIDPS, HIDPS, KIDPS and Arduino Board according to spatial and semantic distributions based on detection method.

We noted that the embedded system has, in the case of an analysis of the network, the fastest response time, when the software system prevails in the case of the direct protection of a host. Nevertheless, the software system offers opportunities for more advanced prevention. These results support the importance of our probes combination and distribution in the design of our security architecture. A distribution that covers various scenarios and ensures in all cases the best response time.

As further work, we can study the possibility to create with Arduino Boards a Proxy system to improve the prevention of the embedded system.

8. REFERENCES

[1] Open Information Security Foundation. « Getting Started With Suricata ». OISF, 2011

[2] Karen Scarfone, Peter Mell. “Guide to Intrusion Detection and Prevention Systems IDPS”. NIST. US Departement of Commerce. 2007

[3] Daniel Guinier. “Sécurité et qualité des systèmes d'information - Approche systémique”. Masson. 1992

[4] Boriana Ditcheva, Lisa Fowler. “Signature-based Intrusion Detection”. University of North Carolina at Chapel Hill. 2005

[5] Martin Roesch, Chris Green, Sourcefire, Inc. “SNORT User’s Manual 2.9.0”. The Snort Project. 2010

[6] WINPCAP documentation. Copyright (c) 2002-2005 Politecnico di Torino Dsfg

[7] Rachid Guerraoui, Lu´ıs Rodrigues, “Introduction to reliable distributed programming’’, Springer-Verlag, August 24, 2005.

[8] Web Site: Arduino - http://www.arduino.cc/.

[9] Y.SENHAJI, “Network Security: Distributed Agents Approach’’, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 01, Issue 02, July-August 2012

[10] Y.SENHAJI, H.MEDROMI, “Network Security: ARDUINO Yun Based IDS’’, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 4, Issue 4, July - August 2015