A study of intrusion detection and prevention system for network security

Rutuja V.Kotkar[1], Mayuri B.Dandwate[2]

[1,2] *Asst. Professor, MCA Dept, PIRENS Institute of computer technology, Loni, Ahmednagar.MH.*

---***---

ABSTRACT: *This paper deals with the problem of computer security, that aims to develop a strong and freelance security design. This design consists of many probes spatially distributed to many locations within the network (sensitive servers, DMZ, workstations, etc.). These probes square measure NIDPS, HIDPS, KIDPS and Arduino Yun Board. These same probes were semantically distributed in keeping with 3 threat detection ways. At the tip of this paper, we have a tendency to developed a hybrid system consisting of a code IDPS portrayed by an enquiry developed below Visual C ++ associated associate embedded resolution developed below Python in an Arduino YUN board. we stock out a series of computer attacks on our detection system to assess its reaction time.*

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

1. INTRODUCTION

IDPS square measure necessary network security system. In this paper we'll present a mixture of 2 IDPS configuration. the primary configuration could be a code resolution developed with Visual C++.

The second configuration could be a hardware proposal embedded in associate Arduino Yun board. On these systems, we'll create many laptop attacks to examine their reactions.

But before we start, we'll introduce the concepts: detection technique and distributed system so we'll gift the Arduino Yun Board.

2. COMMON DETECTION WAYS

Intrusion detection is that the method of watching the events occurring in a very computing system or network and analyzing them for signs of attainable incidents, that square measure violations or at hand threats of violation of computer security policies, acceptable use policies, or normal security practices.

Among the detection ways employed by IDPS, we found:

- Signature primarily based Detection: this technique relies on the comparison of the units of activities (Package, Log Entry) to a listing of models by victimisation the operators of comparison. A model corresponds to a proverbial threat.
- Anomaly primarily based Detection: it's a technique basing itself on applied mathematics calculations and it's a “Profile” which represents the traditional behavior. thus this technique consists of constructing comparison between the events and also the definition of the events thought of traditional to notice deviations.
- Stateful Protocol Analysis: This technique compares the protocols and their profiles. additionally, it exploits the mix of the request and its answer to be ready to measure the state.

3. DISTRIBUTED SYSTEM

A Distributed system will be distributed supported associate existing abstract distance between its elements.

This distance will be:

- Spatial: distribution by completely different processes appointed to resolve a tangle associated with house.
- Semantic: distribution by the specificity of information and a selected ability.
- Structural: representations square measure heterogeneous and reasoning mechanisms square measure completely different.
- Semantic: in keeping with its perform and its role at intervals the system.

4. ARDUINO YUN BOARD

The Arduino Yun is associate electronic board that uses the Atmel processor ATmega32U4. Besides of that, it's a further processor: Atheros AR9331, that flip the UNIX system distribution OpenWrt Linino.

Fig 1: Arduino Yun Board

5. PLANNED DESIGN

5.1 Introduction

Prior to readying of the protection resolution, we have a tendency to assume that users square measure alert to the importance of security and its challenges which all systems and applications square measure perpetually updated (security patches).

Suppose we've a network with the subsequent elements:

- A space network|LAN|computer network} (local area network): consists of many digital computer
- A zone (demilitarized zone): Consisting of machines on the interior network that require to be accessible from the surface (mail server, FTP server, web server ...)
- A internet Client: consists of outdoor Network

5.2 SPACIAL DISTRIBUTION

To secure the network whereas that specialize in the conception of load reduction and raised reaction time, the protection system are deployed and distributed spatially within the network. it'll be composed of many distributed code IDPS (hereinafter referred IDPS) and hardware embedded Arduino IDPS sensors (hereinafter referred ARD). And for a additional reduction of the info loading on these sensors, they need to be in the course of pre-filtering firewalls that analyze the info stream before capture. Moreover, and for a complementary security resolution we'll mix between NIDPS and HIDPS. HIDPS are deployed on the machines within the zone and on necessary servers. We are able to conjointly add KIDPS (K: Kernel) for sensitive machines. Below the list of probes that we'll use:

- Ks: KIDPS sensing element for sensitive servers
- Hs: HIDPS sensing element for necessary servers
- N1: NIDPS sensing element analyzing traffic between the interior network and also the net
- N2: NIDPS sensing element analyzing traffic between the interior network or zone and net (before the firewall for its protection)
- N3: NIDPS sensing element analyzing traffic between the weather of the zone and net
- Hi: sensing element for HIDPS servers within the zone
- ARD : Network Arduino sensing element

5.3 LINGUISTICS DISTRIBUTION

In this step, we have a tendency to proceed to a second distribution, a linguistics one supported IDPS technique detection. This distinction aims to specialize the IDPS.

Thus, every IDPS and ARD are divided into 3 parts:

- IDPS-SPA: supported the "Stateful Protocol Analysis" as a technique of detection
- IDPS-ABD: supported "Anomaly primarily based Detection" as a technique of detection
- IDPS-SBD: supported "Signature primarily based Detection" as a technique of detection.
- ARD-SPA: supported the "Stateful Protocol Analysis" as a technique of detection
- ARD-ABD: supported "Anomaly primarily based Detection" as a technique of detection
- ARD-SBD: supported "Signature primarily based Detection" as a technique of detection.

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

To achieve our simulation on our system, we've developed three Systems:

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

The second could be a Python script embedded {in a|during a|in associate exceedingly|in a very} Yun Arduino board and doing the role of an IDS by exploiting RAW socket.

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

Thus, we'll at first attack a system protected by the binomial HIDPSS and ARD and second 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 primary step, we'll combine associate HIDPSS associated an ARD as below:

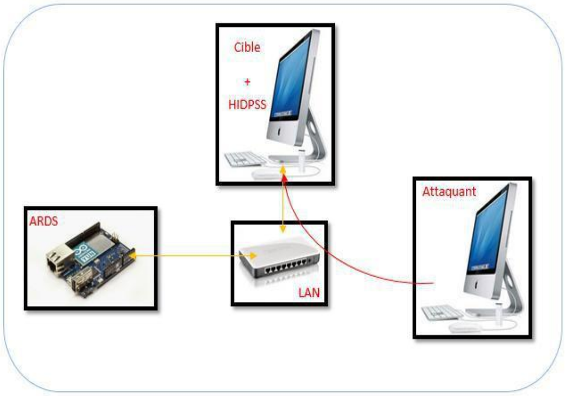
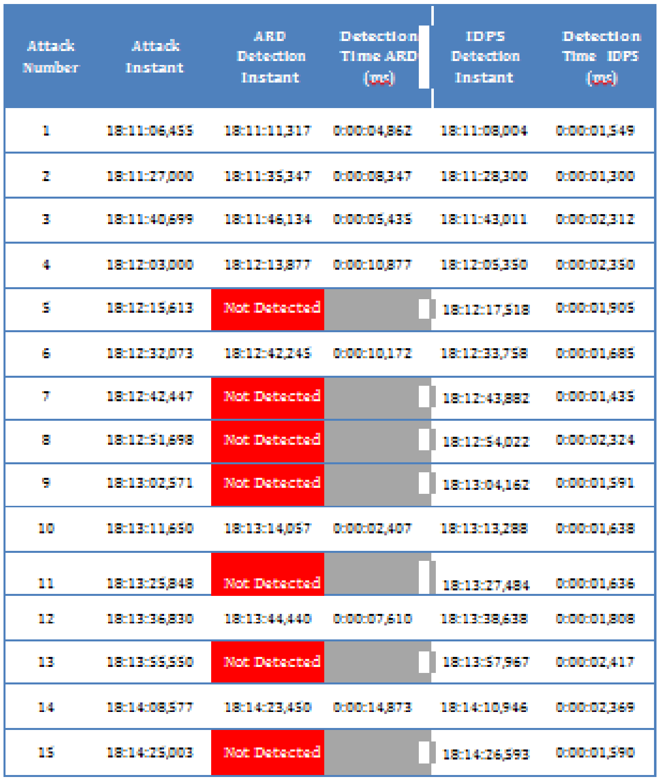


Fig 2: Case HIDPS/ARDS

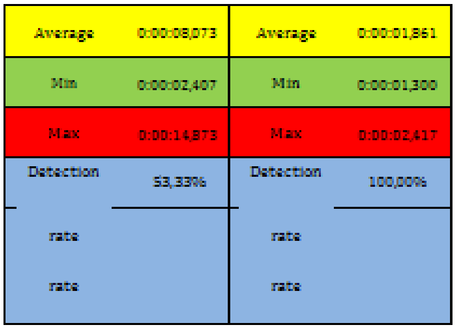
6.1.2 Evaluation of the detection time

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 rate	53,33%	Detection rate rate	100,00%

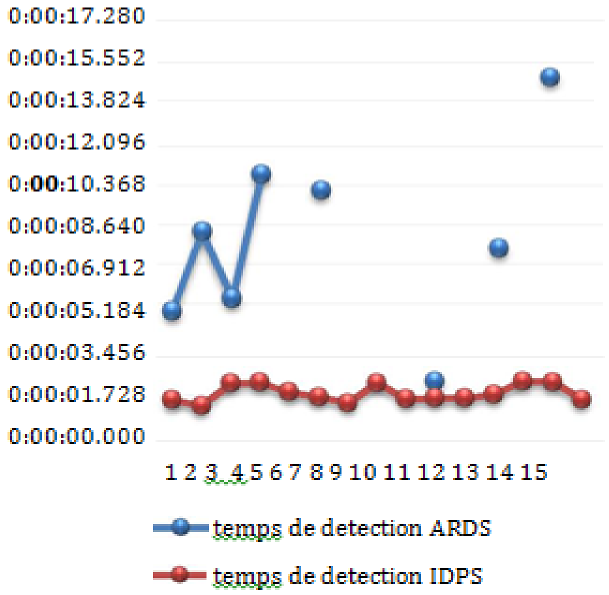


Fig 3: Evolution of the detection time of an attack – HIDPS/ARD

Of course, this detection time could vary relying on:

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

But, yet, we have a tendency to note that:

- The threat detection rate HIDPS is 100% at the time the ARD is simply fifty three.3%
- The detection time of the HIDPSS is considerably higher than that of ARD

Thus, we have a tendency to discover that associate embedded system isn't altogether cases the quickest system. however it depends of security functions.

6.2 NIDPS/ARD System

6.2.1 Diagram of the simulation

In this case we have a tendency to combine associate NIDPSS with associate ARD as below:

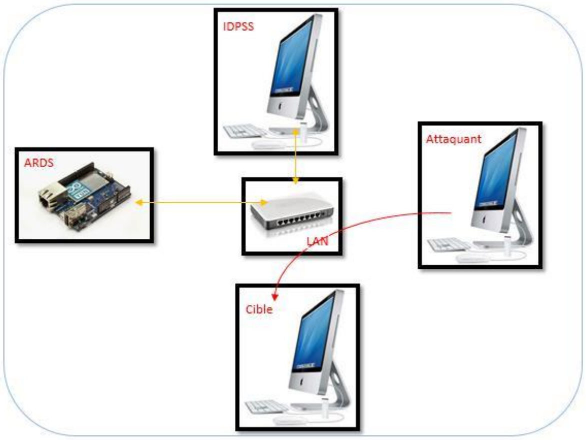


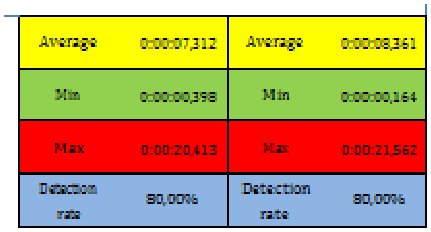
Fig 4: Case NIDPS/ARD

6.2.2 Evaluation of the detection time

We carry out a series of attacks on our detection system to assess its reaction time to associate attack. so we have a tendency to get the results below.

Table 2. outline of various 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 associate attack –NIDPSS/ARD-S

Of course, this detection time could vary in keeping with a similar conditions mentioned within the previous section.

But, yet, we have a tendency to note that:

- The detection rates of ARD and NIDPS don't seem to be 100%
- The ARD detection time is on the average quicker than the NIDPSS

Software system prevails within the case of the direct protection of a number. yet, the package offers opportunities for additional advanced interference. These results support the importance of our probes combination and distribution within the style of our security design. A distribution that covers numerous eventualities and ensures altogether cases the most effective reaction time.

As any work, we are able to study the chance to form with Arduino Boards a Proxy system to boost the interference of the embedded system.

Thus, we are able to notice that in contrast to the previous case, the embedded system has higher performance.

7. CONCLUSION AND ANY WORK

In this paper, we have a tendency to planned hybrid security design supported a distributed approach of

NIDPS, HIDPS, KIDPS and Arduino Board in keeping with spacial and linguistics distributions supported detection technique.

We noted that the embedded system has, within the case of associate analysis of the network, the quickest reaction 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] Boriana Ditcheva, Lisa Fowler. “Signature-based Intrusion Detection”. University of North Carolina at Chapel Hill. 2005

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

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

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

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