Usability of ARM ANN anomaly detector in local networks

David Malanik, and Radek Vala

Abstract— the paper deals with techniques that allow implementation of the ARM platform based network anomaly detector in the computer network. The anomaly detection is based on the ANN. The ARM device is represented by the one board solution with Ethernet port. The main focus of this paper is inside the benchmarking of the ARM platform. The priority output is inside the decision of the usability in real computer environment. The second goal flows from realized test and might provide recommendation for the implementation. The ARM is basically limited power devices and it has limited usage in the high power operation. But for many application it has sufficient power.

Keywords—ARM, computer network, sniffer, anomaly detection, security, ANN.

I. INTRODUCTION

THE question of security inside computer networks represents one of the most important questions of this time. The network are huge and placed everywhere. The population is addicted to fully operated networks with high availability. Many companies solve the problem of stability and security of their networks.

The major problem for the computer networks is DoS (Denial of Service) attack. This type of the attack is followed by the other security break occasionally. DoS attacks represented the highest percentage of computer crime cost in 2013 [1]. Most of DoS and DDoS (Distributed Denial of Service) are launched by the botnets [2], [3]. This fact increasing the necessity of the early detection for any network incidents. The fast detection of the unwanted traffic inside the network is critical. The possible solution is in implementation of the IDS/IPS (Intruder Detection System/ Intruder Prevention System).

The IDS/IPS solution is represented by the specific HW network parts with his own operation and detection system commonly. The other solution is based on the server with specific software (for example SNORT[1], Surikata[2]). This paper deals with the implementation of the IDS/IPS system on ARM platform. The main advantages of the ARM solution is the cost and power consumption of the solution. The main limitation is flowing from the limited performance of this solution. The paper performs use tests for the ARM computer based on the Banana Pi.

.

II. TESTING ENVIRONMENT

The usability test is focused do the three main part of the ARM board. Firstly, there is an isolated network for the testing purposes. The schema of the network is on Fig. 1.

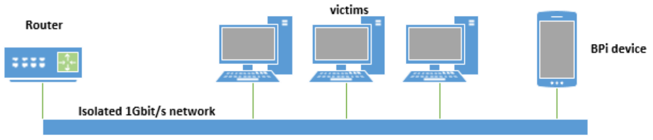

Fig. 1 Network schema

The device for the IDS/IPS role is the Raspberry Pi derivate called the Banana Pi. The purpose of the changing the RPi to BPi depends on the computation power increasing in BPi and existence of the dedicated 1GBit Ethernet port. The HW specification of the device is in Table I.

Table I BPi HW specification

SoC[3]	ARM Cortex-A7 dual-core, 1GHz, Mali400MP2 GPU
System Memory	1GB DDR3 DRAM
Storage	SD card slot, Extensible with SATA connection (2.5" SATA HDD with 5V)
Video output	HDMI, Composite, Extensible with on-board LVDS connector
Connectivity	Gigabit Ethernet
USB	2* USB 2.0 ports, 1* OTG micro USB port, 1* micro USB for power supply
Dimensions	92 mm X 60 mm
Weight	48 g

The work was performed with financial support of research project NPU I No. MSMT-7778/2014 by the Ministry of Education of the Czech Republic and also by the European Regional Development Fund under the Project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

D. Malanik is with the the Faculty of Applied Informatics, Tomas Bata University in Zlín, Nad Stráněmi 4511, 760 05 Zlín, Czech Republic (e-mail: dmalanik@fai.utb.cz).

R. Vala is with the the Faculty of Applied Informatics, Tomas Bata University in Zlín, Nad Stráněmi 4511, 760 05 Zlín, Czech Republic (e-mail: vala@fai.utb.cz).

[1] https://www.snort.org/
[2] http://suricata-ids.org/
[3] SoC = System on a chip

The performance test was focused to the three main component of the BPi device:

- CPU performance, IPS systems contains ANNs (Artificial Neural Network) components.
- HDD performance; it is necessary for the capturing live data from the network environment.
- Ethernet device speed. The communication over system must be transparent for users (without marginal speed degradation).

A. CPU performance test specification

The CPU test was designed as the real procedure contains the ANN training routines. The procedure is written in C++ language and compile directly on the BPi device. The measuring value is the time that require 732 iteration of Back Propagation training process.

Each test had 100 repeat iteration. The fragment of the testing procedure is shown below.

```
for a in $( seq -f "%03g" 1 100 )
do
    echo -n Sequence $a/100:-- >> log.txt
    date >> log.txt
    ./ann
done
```

B. HDD performance test specification

The next part of test is focused to real HDD speed of the BPi storage. The test was realized by the Linux command *dd*; specifically by the various option of this command shown below [4], [8].

```
dd if=/dev/zero of=/root/test.dd bs=4K count=1000
dd if=/dev/zero of=/root/test.dd bs=64K count=1000
dd if=/dev/zero of=/root/test.dd bs=4K count=100000
dd if=/dev/zero of=/root/test.dd bs=64K count=10000        dd if=/dev/zero
of=/root/test.dd bs=256K count=1000
dd if=/dev/zero of=/root/test.dd bs=1M count=1000
```

C. Ethernet device test specification

The network is isolated from other application. The first rand of tests become from LAN testing realized by the *iperf* Linux tool [5]. Test contains 50. The appropriate commands are shown below.

```
for a in $( seq -f "%03g" 1 50 )
do
    echo -n Sequence $a/50:-- >> log.txt
    date >> log.txt
    iperf -c 192.168.215.50 -t 300 -d >> log.txt
done
```

D. Comparison with common SW IDS implementation

The last part of analyses shows the comparison with real implementation on the testing server. The server has following specification:

CPU: Intel ® i7-3770 (4 physical cores, 8 logical)
RAM: 8GB
HDD: 250GB SATA-II
LAN: 1000BASE-T
OS: Debian 7.8 64-bit
IDS/IPS: SNORT 2.9.7.2

III. CPU PERFORMANCE TEST

This part contains reports from realized test with ANN training process. The device (BPi) was tested with Back Propagation training process. The training process had 732 iteration and it was repeated 100 times.

The Table II shows the duration of each iteration.

Table II CPU performance test

Iteration		100
Duration/s	MIN.	116
	MAX.	120
	AVG.	118.1
Operations/s	MIN.	6.10
	MAX.	6.31
	AVG.	6.19

The shortest execution time was **116s for 732** iteration of the training cycle. The **average execution time was 118.1s**. The system provides **avg. 6 operations per second**. The fluctuation of execution time is shown on Fig. 2.

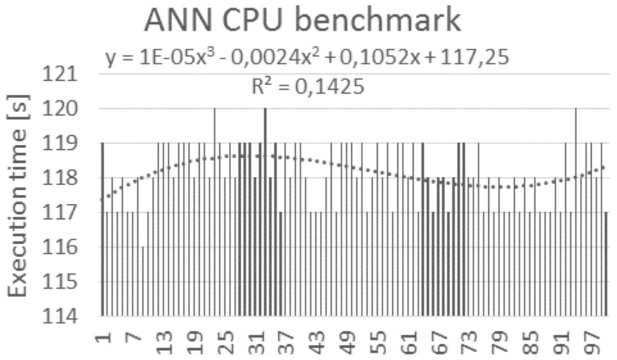

Fig. 2 CPU performance test

The Fig. 2 also contains the approximation of the execution time process. The regression is not so accurate, because the ARM device caching some values to the RAM and the training process is not be stable process with constant difficulty. The 4 second difference between minimal and maximal value represents 3.4% variability of the execution time. This difference is the relatively good result [7].

IV. HDD PERFORMANCE TEST

These results reported the HDD performance speed of the BPi device. The speed depends on the SDHC card that was used as the storage of the solution. The SDHC card was Class 10 Kingston with guaranteed minimal write speed 10MB/s. The other card might bring higher power. The test was realized by the Linux command *dd*.

The Table III shown the maximal, minimal and average value of disc bandwidth examined by the testing procedure. The test set contains 4 KB, 64 KB, 256 KB and 1 MB blocks.

Table III HDD benchmarking

	MB/s					
	4KB high	4KB	64KB high	64KB	256KB	1M
MIN.	3.2	9.5	82.6	9.9	8.8	9.8
MAX.	111	12.1	87.2	12.0	14.0	11.4
AVG.	71.4	11.1	85.5	11.3	12.3	10.7

The high versions represent test result with 10 000 (other 1 000) block. The device might perform increasing of power by caching values inside the RAM with small block size and small number of blocks. The test report with 4KB and 64KB block size does not reflect real values with small number of blocks. The benchmark fluctuates from **3.2 to 111 MB/s with RAM support operations.**

The relevant data is with bigger number of block or with higher block size. The real disk bandwidth was between **10.7 and 12.3 MB/s** (reported from the average bandwidth measurement).

The next figures represents the fluctuation of HDD speed in particular tests with variable block size and number of written blocks.

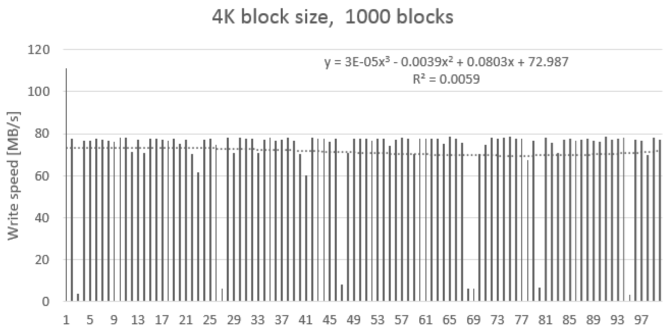

Fig. 3 HDD 4KB, 1000x

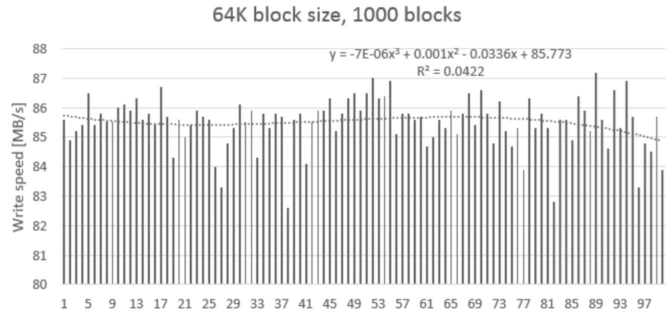

Fig. 4 HDD 64KB, 1000x

Figures Fig. 3 and Fig. 4 show the invalid test result. The BPi wrote values to the RAM in this scenario and write process to the physical HDD was delayed. This reports does not reflect the speed of the storage, but for the usability test also play important role. This report shows, how it operate with small data values. If the sniffer of the IDS provide small packet for analyses, the device might save this packet to the memory and operates with it very quickly. The operation speed is important for the early detection of each attack.

Relevant write test results are shown below (Fig. 5 - Fig. 8).

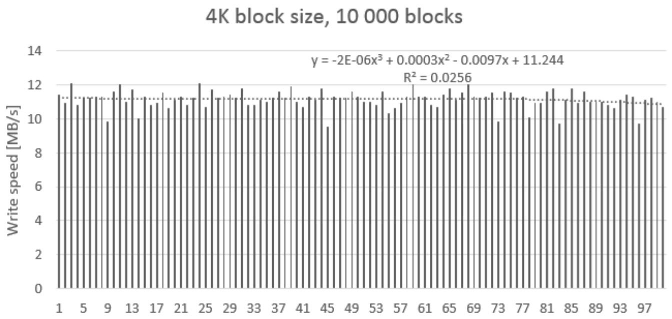

Fig. 5 HDD 4KB, 10 000x

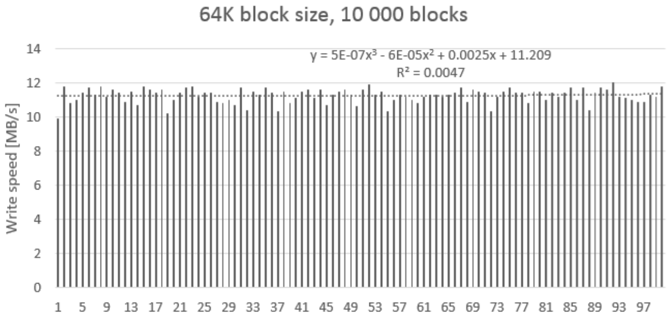

Fig. 6 HDD 64KB, 10 000x

The upper figures reflect to the real speed of the storage. The write speed fluctuate around 10MB/s. This speed was minimal guaranteed speed of write operation on the used SDHC card.

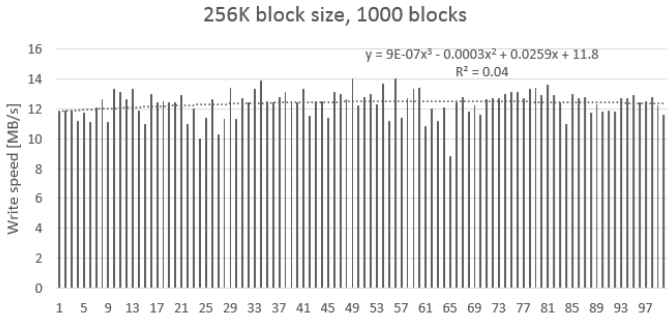

Fig. 7 HDD 256KB, 1000x

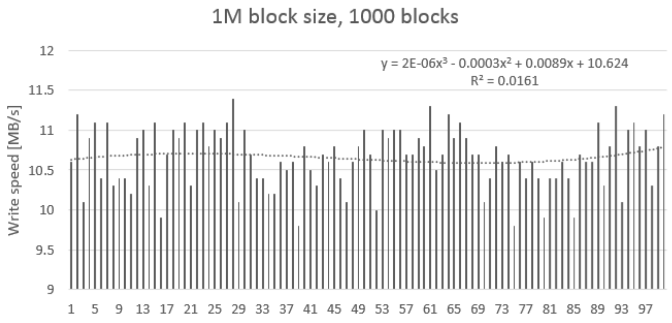

Fig. 8 HDD 1MB, 1000x

Tests shown on Table III and Fig. 5 - Fig. 8 represents the real speed of used storage in the BPi device.

V. Ethernet device test

The network was isolated from other application. The first rand of tests become from LAN testing realized by the *iperf* Linux tool [5]. Test contains 50 measurements [6]. All tests was realized with isolated network shown on Fig. 1. Tests was realized between BPi devices and clean server with i7 CPU, gigabit Ethernet and Kali Linux OS.

Table IV IPERF network bandwidth test

Iteration		50
Download/ Mbit/s	MIN.	578
	MAX.	647
	AVG.	609.5
Upload/ Mbit/s	MIN.	166
	MAX.	243
	AVG.	205.6

Table IV shows examined values of network bandwidth on both direction. The download speed fluctuated between **578 and 647 Mbit/s.** This value is close to the real bandwidth on the gigabit Ethernet. The upload speed is lower significantly. The upload fluctuated from **166 to 243 Mbit/s.** This might produce an important limitation for the device. But the upload speed is not critical parameter in IDS/IPS application. The critical parameter is the download speed because the device must be able to read data from the network very fast.

The download speed produces one important issue; if there is a huge quantity of captured data, the limitation will flows from the small write speed of the device storage. Tests described in previous chapter examined that maximal write speed of the storage was approx. 10 MB/s (80 Mbit/s), which is significantly lower than the download speed of Ethernet interface.

The download speed histogram from 50 measurement is shown on Fig. 9.

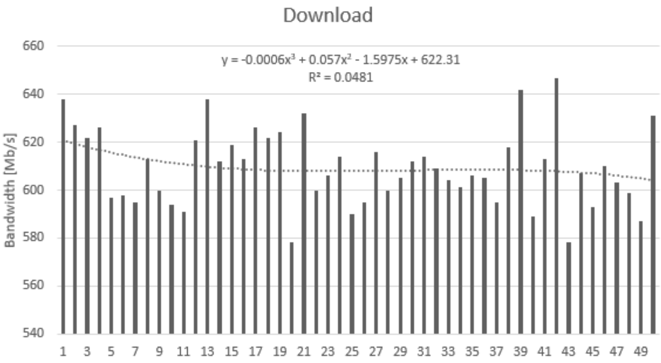

Fig. 9 Download speed histogram

The download speed on the interface fluctuate few. The download speed is stable particularly. These is not any significant variances of the speed.

The next figure reports the speed variation of the upload speed on the interface.

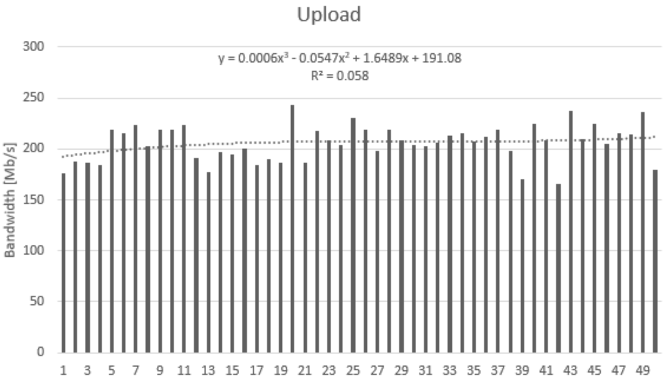

Fig. 10 Upload speed histogram

Also the upload speed on the interface fluctuate few. The speed is stable particularly. These is not any significant variances of the speed.

VI. Comparison with common SW IDS implementation

This part describe the comparison the implementation based on the BPi device and other solution based on the real server with Intel i7 CPU, 8GB RAM, 7200rpm HDD and 1Gbit/s Ethernet port. The HW specification is totally different but the size, price and power consumption is marginally higher! The next table compare the basic parameters.

The Comp. parameter represents the percentage difference between BPi and i7 (BPi represented 100%).

Table V Parameters comparison i7 vs. BPi

Parameter	Server i7	BPI ARM	Comp.
performance	very high	lower	---
size	1U rack	92x60x25 mm	---
price[4]	2200 EUR	73 EUR	3 014%
Power c.	340W	2.5W	13 600%

The next part contains the comparison of the CPU/SOC speed.

The server (i7) was tested with Back Propagation training process. The training process had 732 iteration and it was repeated 100 times.

The Table VI shows the duration of each iteration.

Table VI CPU performance test i7

Iteration		100
Duration/s	MIN.	6
	MAX.	7
	AVG.	6.08
Operations/s	MIN.	104.6
	MAX.	122.0
	AVG.	120.4

[4] Price in EUR converted from CZK (15.04.2015)

The shortest execution time wass **6s for 732 iteration** of the training cycle. The **average execution time was 6.08s**. The system provides **avg. 120 operation per second**. The fluctuation of execution time is shown on Fig. 11.

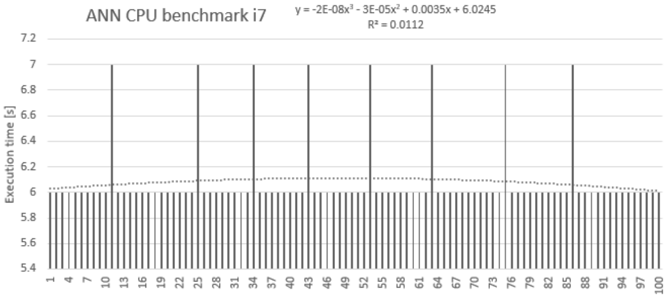

Fig. 11 ANN CPU benchmark i7

The comparison of CPU performance is shown in table below (BPi represented 100%).

Table VII ANN CPU benchmark i7 vs. BPi

Iteration				100
		i7	**BPi**	**i7 vs. BPi**
Duration/s	MIN.	6	116	5.1%
	MAX.	7	120	5.8%
	AVG.	6.08	118.1	5.1%
Operations/s	MIN.	104.6	6.10	1 715%
	MAX.	122.0	6.31	1 933%
	AVG.	120.4	6.19	1 945%

The result reflect the marginal power difference between server solution with i7 and BPi with ARM SoC. The i7 solution is approx. 20 times faster in ANN training process than the BPi SoC. This result was predictable, because the i7 is more power CPU than any ARM SoC in the World.

But with the other comparison is the angle of the view different. The solution with i7 was **20 times faster**, but the **price is 30 times higher** and the **power consumption is 136 times higher**. In these factors, the solution based on the BPi is economical for continuous performance.

The graphical comparison is shown on Fig. 12.

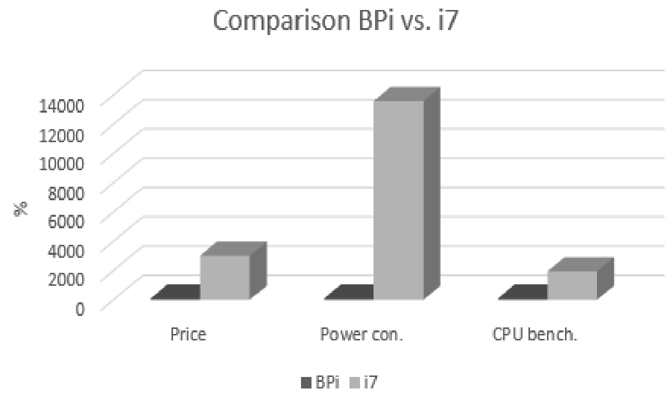

Fig. 12 Comparison BPi vs. i7

VII. Limits of usability

The solution based on the BPi platform has the specific limits for usage; these limits flowing from the realized tests. The firs limitation is inside the speed of the retraining of the neural network for the IDS/IPS functions. The speed is significantly lower than in solution with Intel i7 CPU. But the necessity of the high re-training speed is only in solution that has huge traffic and high variability of traffic signature. For many implementation this does not represents a problem.

The second issue is the slow storage speed. The network bandwidth test revealed that the speed of interface is too close to the real 1 Gbit/s interface. But the other test of the storage write speed shows the limitation of the write operation with storage on BPi. The limitation is approx. 10 MB/s. This result degrade the usability of the BPi solution only for the Fast Ethernet Networks[5].

The usability on the 1 Gbit/s network is discussable; the test with small quantum of data[6] shows, that this small block is operated with speed approx. 80MB/s. If there is not a continuous traffic close to 1 Gbit/s, the solution will work normally.

VIII. Future work

The future work will be focused to the next tests with BPi platform. The testing procedure will be realized with various SDHC card with different speed class. The next option is the use the embedded SATA controller in the BPi and connect laptop HDD as the storage; that might provide the increasing of the storage write speed, but it is possible, that the solution provide next limitation of usage flowing from the limited power of integrated SATA controller. And not in the end, the testing procedure might be repeating on the new version of the Raspberry Pi[7]. The option of the relatively cheap solution for the IDS/IPS systems based on the widely supported platform as the BPi or RPi is very interesting.

IX. Conclusion

The previous chapters describe the entry testing procedures and results of the usability of ARM based IDS/IPS systems with ANN features. The paper is focused as the entry study for the further research that currently running.

The result show the usability limits for the common version of ARM development board. But shows the comparison with “Big” solution based onto server with i7 CPU architecture.

The BPi solution is usable in many scenarios, but the implementation must calculate with its limits. The solution is much cheaper and have marginally lower power consumption. The price and power consumption might produce the new angle of view for these devices. It is possible to implement not only one IDS/IPS solution based on the powerful server. It is possible to implement small IDS/IPS device inside each

[5] 100 Mbit/s Ethernet networks
[6] Less than 66MB in one block
[7] But this SoC has still only 100Mbit/s Ethernet interface

network segment and implement similar scenario that used honeynet and honeypots. The correlation of detection thread with each devices might be helpful for the advanced network attacks.

As it described in the introduction of this paper, we need the solution that provide fast identification for the networks attack such as DoS and DDoS. This detection need implementation of the IDS/IPS systems inside the network. Let’s try distribute the ISD/IPS to many small cell everywhere inside the network.

REFERENCES

[1] Ponemon. "2013 cost of cyber crime study reports," 18.12., 2015; http://www.hpenterprisesecurity.com/register/thank-you/2013-fourth-annual-cost-of-cyber-crime-study-global.

[2] M. S. Kang, S. B. Lee, V. D. Gligor, and Ieee, “The Crossfire Attack,” *2013 Ieee Symposium on Security and Privacy (Sp)*, pp. 127-141, 2013.

[3] Chen, and Chia-Mei, "Detecting botnet by anomalous traffic," Lin and Hsiao-Chung, eds., *Journal of Information Security and Applications*, 2014.

[4] E. Nemeth, *UNIX and Linux system administration handbook* 4ed., Upper Saddle River, NJ: Prentice Hall, 2011.

[5] M. Palmer, *Guide to UNIX Using Linux*, 4 ed., pp. 697, Australia: Thomson/Course Technology, 2008.

[6] B. R. Chang, H.-F. Tsai, and C.-M. Chen, “Empirical Analysis of Server Consolidation and Desktop Virtualization in Cloud Computing,” *Mathematical Problems in Engineering*, 2013.

[7] M. Sharif, K. Singh, J. Giffin, and W. Lee, “Understanding precision in host based intrusion detection formal analysis and practical models,” *Recent Advances in Intrusion Detection, Proceedings*, vol. 4637, pp. 21-41, 2007.

[8] B. Djordjevic, V. Timcenko, “Ext4 File System performance Analysis in Linux Environment,” in 11th WSEAS International Conference on APPLIED INFORMATICS AND COMMUNICATIONS (AIC '11), Florence, Italy, 2011.