EAI Endorsed Transactions
on Scalable Information Systems

Research Article ICST.ORG

High-performance Architecture of Network Intrusion Prevention Systems

Zhao Yueai[1, *] , Hou Pengcheng[1] , Wang Ling[2] and Han Suqing[1]
[1]Department of Computer Science, Taiyuan Normal University, Taiyuan, P. R. China
[2] Department of Computer Science, Engineering College of Shanxi University

Abstract

Software-based Network Intrusion Prevention Systems have difficulty in handling high speed links. Network processor (NP) is an emerging field of programmable processors that are optimized to implement network data. In this paper, a novel Network Intrusion Prevention scheme is designed based on a heterogeneous multi-core processing architecture where its NP devices complement genera purpose multi-core processors to improve the performance of packet processing. We use Netronome's network processor to process network traffic at the data link (Ethernet), network (IP), and transport/control layers. A set of network-based anomaly Intrusion Detection sensors is used in processing network traffic. Experimental results show our enhancements can reduce the processing load of the Intrusion Detection sensors. The load balancing by the protocol is better then other previous work.

Keywords: Network Intrusion Prevention, Network Processor, heterogeneous multi-core processing architecture, anomaly detection.

Received on 21 March 2014, accepted on 24 April 2014, published on 27 May 2014

doi: 10.4108/ sis.1.3.e3

1. Introduction

The rapid development of mobile Internet, cloud computing, intelligent terminal is becoming a new engine of economic development. It has greatly changed the mode of people's work and life, as the logic derivative product of virtualization, networking and cloud computing. Big data not only means the mass data, more complex, but also include more sensitive data, these data will be attractive to potential attackers. Disclosures sensitive data become a significant goal of network intruders, increase risk of personal privacy leakage.

Network intrusion prevention systems (NIPS) are lagging behind routers and firewalls in the technology curve. The complexity stems mainly from the need to analyze not just packet header but also content and higher-level protocols. NIPS needed to be updated with new detection components and heuristics. Anomaly detection can detected unknown attacks, but has high false positive rate. Over the past several years, many techniques are employed in Network Intrusion Detection Systems (NIDS), such as data mining, artificial immune system, support vector machine and so on. But, software-based network intrusion prevention systems have difficulty in handling high speed links. Network processor is an emerging field of programmable processors that are optimized to implement data. In this paper, a novel network intrusion prevention scheme based a heterogeneous multi-core processing architecture where its NP devices complement genera purpose multi-core processors. We use Netronome's Network Processor to process network traffic at the data link (Ethernet), network (IP), and transport/control layers (TCP, UDP, ICMP). A set of anomaly Intrusion Detection sensors is used to processing network traffic in Intel Xeon E5620 processor with four core eight hyper-threading. Experimental results shows that our enhancements can reduce the processing load of the sensors, and load balancing by the protocol is better then the five-tuple consisting of the source IP address, destination IP address, protocol type, TCP/UDP source port, and TCP/UDP destination port.

The rest of this paper is organized as follows. In Section 2 we discuss work related to anomaly detection. In Section 3 we describe NP and the heterogeneous multi-core processing architecture and implementation of our system. We examine the performance benefits of using NP-based load balancing in Section 4. Finally, we summarize and comment on future research directions in Section 5.

*Corresponding author: tysyzya@sina.com

2. Existing Work

There are two main types of Intrusion Detection/ Prevention Systems: signature-based (SBS) and anomaly-based (ABS). SBS systems (e.g. Snort) rely on pattern recognition techniques where they maintain the database of signatures of previously known attacks and compare them with analyzed data. An alarm is raised when the signatures are matched. On the other hand ,ABS systems build a statistical model describing the normal network traffic, and any abnormal behavior that deviates from the model is identified. In contrast to signature-based systems, anomaly-based systems have the advantage that they can detect zero-day attacks, since novel attacks can be detected as soon as they take place [1]. Many distinct techniques are used based on type of processing related to behavioral model. They are: Statistical based, Operational or threshold metric model, Markov Process or Marker Model, Statistical Moments or mean and standard deviation model, Univariate Model, Time series Model, Cognition based, Finite State Machine Model, Description script Model, Adept System Model, Machine Learning based, Bayesian Model, Genetic Algorithm model, Neural Network Model, Fuzzy Logic Model, Outlier Detection Model, Computer Immunology based, User Intention based [1][2][3]. We describe the prominent and the most recent of Network Intrusion Detection System.

Heberlein first proposed Network Intrusion Detection System NSM [4], After that many anomaly detection methods were developed, such as Bayesian network method, Markov chain method, mining method, neural network method and statistical analysis method [5,6,7,8,9]. eBayes anomaly detection component of EMERALD is based on the Bayesian network model [5]. EMERALD was the winners of the DARPA IDS evaluation in 1999, by learning to obtain various characteristics of the conditional probability of an attack class. It classified traffic on the network to determine whether it belongs to some kind of attack. Although it was the winner of DARPA evaluation, the detection rate only reached 50%.

SPADE[6] is a plug-ins based on statistical anomaly detection of Snort, used to automatically detect the port scanning attacks, is also the first to use the idea of anomaly detection to detect port scanning attack, the model used a simple frequency-based method to calculate the value of a packet exception, if the given data packet appear less, abnormal value is higher, when the abnormal data packet exceeds the given threshold, the packet is forwarded to a correlation engine. However, due to the SPADE take all the packets that have not encountered as attacks, therefore, SRADE high false alarm rate.

Matthew.v.Mahoney proposed a series of anomaly detection methods such as PHAD, ALAD and LERAD[7] , These all use time-based models, in which the probability of an event depends instead on the time since it last occurred. For each attribute, they collect a set of allowed values (anything observed at least once in training), and flag novel values as anomalous. Specifically, they assign a score of tn/r to a novel valued attribute, where t is the time since the attribute was last anomalous (during either training or testing), n is the number of raining observations, and r is the size of the set of allowed values. PHAD, ALAD, and LERAD differ in the attributes that they monitor. PHAD (Packet Header Anomaly Detector) has 34 attributes, corresponding to the Ethernet, IP, TCP, UDP, and ICMP packet header fields. ALAD (Application Layer Anomaly Detector) models incoming server TCP requests: source and destination addresses and ports, opening and closing TCP flags, and the list of commands (the first word on each line) in the application payload. Depending on the attribute, it builds separate models for each target host, port number (service), or host/port combination. LERAD (LEarning Rules for Anomaly Detection) also models TCP connections, but samples the training data to suggest large subsets to model.

3. Heterogeneous Multi-core Processing Architecture Based Network Processor

3.1 Network Processor

Network processor is multi-core processor, augmented with networking-specific instruction, hardware-assists, and memories. This has its own dedicated circuit structure and designed for network packet processing, at the same time it is a programmable chip. Network processor not only can be programmed to optimize packet processing, but also can take over many of the original master CPU complete control and management functions. Currently there are many companies producing network processor on the market, such as the dedicated packet processing network processors are IBM, Netronome, MMC, Xstrema companies. In this paper, we provide an overview of the architecture of the Netronome’s network processors with the NFE-i8000 as an example, there are 16 micro-processing engines, accelerated network processing is required to reach the 10~40 Gbps data rate that many network applications demand.

Research about network processor is focused on the distribution of micro-engine, code optimization and detection algorithm, they aims to take full advantage of multi-core processors by parallelism, and detection systems are mostly used Snort to run on the general-purpose processing platform. Charitakis created a tool that can translate Snort rules into microcode S2I for packet header detection [9]. Hyeyoung proposed the concepts of DIDE and CIDE modular detection [10]. Konstantinos Xinidis describes the design and implementation of a high-performance Network Intrusion Prevention System that combines the use of software-based Network Intrusion Prevention sensors and an IXP1200 network processor board [11].

Most of the studies on NP use Snort as sensors, but snort is rule-based pattern matching techniques, with the kind of attack increased, rules must be constantly updated. With the increasing number of rules and increased the detection computational load, it is difficult to meet the demand for high-speed networks. So we use anomaly detection algorithm to detect new attacks.

3.2 Design an intrusion prevention system on the heterogeneous multi-core processing architecture

In this paper, a novel network intrusion prevention scheme is designed, based on a heterogeneous multi-core processing architecture where its NP devices complement genera purpose multi-core processors such as Intel’s Xeon family.

The primary metric of interest in the design of NIPS is throughput. To take advantage of the high-speed capabilities of network processors’ flow processing and multi-core processors common data processing. The common multi-core x86/IA processor handles application layer services, and professional network processor handles the data link (Ethernet), network (IP), and transport/control layers business, through PCIE accelerator tight coupling of these two processors, can be achieved to a linear, safety, virtualization and unified platform. Figure 1 shows the architecture and data paths.

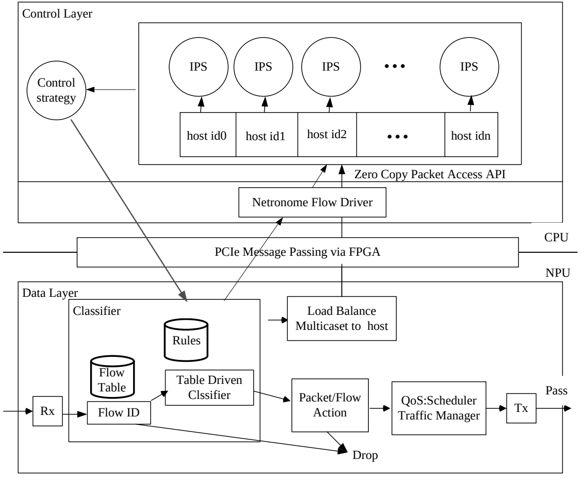

Figure 1 Architecture and Data paths

Designs are composed of two layers: data layer and control layer. The control layer is X86/IA common processing architecture as a general-purpose multi-core processors application and control CPU. It is responsible for the management and control systems. On the other hand, according to test results, It controls the packet and flows of data layers. Intrusion protection/detection through the load balancer group supports multi-processor parallel analysis of data packets, mainly to detect the anomaly behavior. According to results submitted to the policy control module, the policy control module then translate the action to the data flow processing operations in the packet classifier filters, while updating the control policy table.

The data layer is composed of the network processors, as a coprocessor to achieve precise and high-performance handing in network level. It achievers high-speed data streams effective control and handling through a multi-engine (ME) parallel processing, Which mainly contains the following modules: Packet receiving module Rx, packet classification filter module Classifier, Qos module and data packet transmission module Tx. Packet classification filter determines the flow associated with a packet, and then configures the control strategy based on the detection process, the control strategy for different data streams delineate different levels of defense, which take different processing operations, such as direct forwarding (pass). directly discarded (drop). to send data to the control surfaces on the Intrusion Detection group (via_host) or directly pass via_host (Tap-to-host). Qos module for network data flows congestion control, packet scheduling queue processing and optimization, In order to ensure the function of intrusion protection system, it can improve the processing performance.

Then depicts the main data paths in this architecture, Traffic enters the system at the Rx block. Then the Flow Identification subsystem (Classifier) determines the flow associated with a packet. When the first packet in a flow is received, the subsystem creates a new flow table entry. For subsequent packets in the flow, the subsystem retrieves the existing flow table entry. Should the policy to be applied to the flow be known, the system applies the policy by handling the traffic within the Netronome Flow Engine(NFE), by directing the traffic to the host CPU(via_host) or a combination thereof (Tap-to-host).

Traffic can be directed to host CPU applications via the Netronome Packet Access (zero copy) API. This API minimizes kernel mode to user mode transitions and data copying, thereby improving performance. A modified libpcap implementation provided by Netronome Flow Manager (NFM) can be used to run applications that require libpcap for accessing packets. Any of the before mentioned types of partner application can modify the policy applied to a specific flow (or set of flows), or modify (i.e. add, change, or delete) rules or the policy associated with rules. These operations are depicted using star symbols in the diagram. Traffic exits via the scheduler/traffic manager and the Tx block [12].

4 Packet Header Anomaly Detection(PHAD)

PHAD was an anomaly detection algorithm that learns the normal ranges of values for each packet header field at the data link (Ethernet), network (IP), and transport/control layers (TCP, UDP,ICMP)[7]. PHAD uses the rate of anomalies during training to estimate the probability of an anomaly while in detection mode. If a packet field is observed n times with r distinct values, there must have been r "anomalies" during the training period. If this rate continues, the probability that the next observation will be anomalous is approximated by r/n.

In this model, if an event last occurred t seconds ago, then the probability that it will occur in the next one second is approximated by 1/t. Often, when an event occurs for the first time, it is because of some change of state in the network .Thus each packet header field containing an anomalous value is assigned a score inversely proportional to the probability,

$$Scorecfield = {}^{tn}\!/_{r}$$

Finally, we add up the scores to score the packet.

5. Implementation and Evaluation

5.1 Implementation

In this Section, we examine the performance of our architecture. We use Netronome's network processor NFE-i8000 to process network traffic at Layers 4-7 include Packet capture and load balancing. Server is a 2.4 GHz Intel Xeon E5620 processor with four core eight hyper-threading disabled. The host operating system is Linux (kernel version 2.4.20, Red-Hat 9.0).The sensor software is a modified PHAD.

5.2 Evaluation data

Our experiments were performed on the DARPA 1999 evaluation data; Three weeks of training data were provided for the 1999 DARPA Intrusion Detection off-line evaluation. The first and third weeks of the training data do not contain any attacks. This data was provided to facilitate the training of anomaly detection systems. The second week of the training data contains a select subset of attacks from the 1998 evaluation in addition to several new attacks. The forth and fifth weeks of data are the test data used in the 1999 Evaluation from 1999/6/19 to 1999/10/1. There are 201 instances of about 56 types of attacks distributed throughout these two weeks [13].

5.3 Result

We use the third week of data as training data, followed load balancing with the 5-tuple configuration to 8 CPU(host id0 -host id1), the fourth and fifth weeks 9 days data as the test dataset, the results are shown in Table 1.

Table 1 list the experimental results at the false alarm rate is 101, 8 general purpose processors are running the PHAD. The experimental results show load balancing the packet by 5-tuple can only guarantee the same flow but not the same attack information to the Intrusion Detection sensors in the control layer, so the total number of detected attacks is more than real attacks. Then we change the load balancing strategies by protocol processing model.

In the next experiments, we made a comparative experiment on the third week data with splitting and without splitting training dataset by protocol. while the same test dataset of the fourth and fifth week total of 9 days of data, which are divided according to the protocol type TCP, UDP, ICMP and others, the experimental results are show in the table 2, we can see that in this system load balancing the packet by protocol is better results than the original way down by Mahoney VM, especially the training dataset is also load balancing by protocol type.

Table 1 5-tuple configuration load balancing

Attack	Detection rate(false alarm below 10 per day)								
type	0	1	2	3	4	5	6	7	Total
Probe	11/37	10/37	14/37	12/37	7/37	10/37	9/37	15/37	88/37
Dos	15/65	13/65	9/65	11/65	14/65	11/65	11/65	17/65	101/65
R2L	4/56	3/56	3/56	6/56	4/56	6/56	1/56	3/56	30/56
U2R	1/37	1/37	2/37	2/37	1/37	0/37	1/37	1/37	9/37
Data	1/16	1/16	1/16	1/16	1/16	0/16	0/16	0/16	5/16
New	5/62	4/62	6/62	6/62	4/62	5/62	1/62	4/62	35/62
All	31/201	27/201	28/201	31/201	26/201	27/201	22/201	36/201	228/201

Table 2 the protocol load balancing

Attack	Detection rate(false alarm below 10 per day)										PHAD
type	TCP		UDP		ICMP		others		Total		
Probe	11/37	12/37	6/37	2/37	1/37	9/37	0/37	0/37	18/37	23/37	18/37
Dos	5/65	5/65	9/65	7/65	9/65	9/65	0/65	0/65	23/65	21/65	21/65
R2L	2/56	1/56	0/56	0/56	0/56	2/56	0/56	0/56	2/56	3/56	2/56
U2R	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37
Data	0/16	0/16	0/16	0/16	0/16	0/16	0/16	0/16	0/16	0/16	0/16
New	9/62	8/62	1/62	0/62	0/62	2/62	0/62	0/62	10/62	10/62	9/62
All	18/201	18/201	15/201	9/201	10/201	20/201	0/201	0/201	43/201	47/201	41/201

6 Conclusions

In this paper, we have presented a high-performance network intrusion prevention system. It is based a heterogeneous multi-core processing architecture where its NP devices complement genera purpose multi-core processors. A customized load-balancing component built using the Network Processor, and a number of sensors implemented on multi-core processors. To achieve macro security situational awareness and micro threats found on network, we must to analysis big data, Such as APT (Advanced Persistent Threat) attack with high-end testing. Therefore, there are several directions that we are currently pursuing. First, we are design parallel anomaly detection algorithm on multi-core platform to process large data sets. Second, we are re-examining the structure of the sensor, consider the possibility of using a more fine-grained protocol processing model.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China (No. 61273294), a grant from the Key Technologies of Shanxi Province of China (No.20110321024-02)& Shanxi Science and Technology Infrastructure Program (No.2012091003-0104).

References

[1] V.Jyothsna, V.V. Rama Prasad, K.Munivara Prasad. A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications, Vol. 28, No. 7, pages: 26-35, August, 2011.

[2] H. Wang, Y. Zhang, J. Cao, Effective collaboration with information sharing in virtual universities, IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 6, pages: 840-853, June, 2009.

[3] Xiaoxun Sun, Hua Wang, Jiuyong Li, and Yanchun Zhang. 2012. Satisfying Privacy Requirements Before Data Anonymization. Comput. Vol.55, No.4, pages: 422-437, April, 2012.

[4] L.T. Heberlein, G.V. Dias, K.N. Levitt, B. Mukherjee with J. Wood, D. Wolber, A Network Security Monitor.Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy. Oakland, CA.pages: 296-304, 7-9 May 1990.

[5] A.Valdes, K.Skinner, Adaptive model-based monitoring for cyber attack detection, Proceeding of the 3rd international workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France, 2000

[6] S.Staniford, J.A.Hoagland, J.M.McAlerney. Practical automated detection of stealthy portscans. Journal of Computer Security, 10: pages: 105-136, 2002.

[7] V.M.Mahoney. A machine learning approach to detecting attacks by identifying anomalies in network traffic. Melbourne: Florida Institute of Technology, 2003.

[8] Min Li, Xiaoxun Sun, Hua Wang, Yanchun Zhang, and Ji Zhang. 2011. Privacy-aware access control with trust management in web service. World Wide Web 14, 4 (July 2011), pages:407-430.

[9] G.Vasiliadis, M.Polychronakis et al. MIDeA: A Multi-Parallel Intrusion Detection Architecture. In Proceedings of the 18th ACM/SIGSAC Conference on Computer and Communications Security (CCS). October 2011, Chicago, IL, USA.

[10] C.Hyeyoung , Daeyoung Kim, Juhong Kim, Yoonmee Doh, et al. Network Processor Based Network Intrusion Detection System.ICOIN 2004, pages: 973-982.

[11] K.Xinidis, K. G. Anagnostakis and E. P. Markatos. Design and Implementation of a High-Performance Network Intrusion Prevention System. In Proceedings of 20th International Information Security Conference.Chiba,Japan, 2005, Vol. 181, pages: 1571-5736.

[12] Netronome,Netronome Flow Manager Programmer's Reference Manual Version 2.2, 2009.

[13] MIT Lincoln Laboratory, 1999 DARPA Intrusion Detection Evaluation Data Set http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1999data.html.