Building IDS Log Analysis System on Novel Grid Computing Architecture

Wei-Yu Chen, Wen-Chieh Kuo, Yao-Tsung Wang
National Center for High-Performance Computing, Taiwan
{ waue ,rock ,jazz}@nchc.org.tw

Abstract

With the increasing of viruses and malicious attacks, the volume of alerts generated by Intrusion Detection System (IDS) becomes very large. Using conventional methods to analyze a lot of data would drag on system performance. In this paper, we propose an IDS log analysis system, named ICAS, to provide a summarized alarm reports. This system is based on the new grid computing platform (Hadoop) and operated by the designed Map / Reduce algorithms. In our experiments, ICAS can achieve at least 89% of the integration rate and provide good performance with large data sets. Hence, ICAS can perform an analysis system with high performance and good summarized ability.

Keywords: Hadoop, IDS, alert correlation, Map Reduce, cloud computing.

1. Introduction

Mostly, the researches of intrusion detection area are involved in the false positive and false negative of Intrusion Detection System (IDS). However, how to find out the malicious intrusion precisely is not the only problem. From the view of an administrator, a lucid and readable log appearance is exigent and essential. Nowadays, most of the IDS log management system store alerts by handling domain into database and show these results by query method. However, there are several problems in this method. Firstly, large amount of data would cause database less efficient. Secondly, it is easy to ignore the crucial information in large amount of alerts. Moreover, if the database were crash, all of the alerts would be missing.

By using parallel and distributed computing, there are several benefits for resolving above problems, such as analyzing huge data sets, high performance for reading access and fault tolerance. In this paper, we propose an IDS-Log Cloud Analysis System (ICAS), to analyze the IDS logs and provide a summarized alarm reports. In order to let this system operate on a new grid computing platform named as Hadoop, which is also well known as Cloud Computing platform, we design a Map / Reduce algorithms to adapt it.

The rest part of the paper is organized as follows: Section 2 reviews related works and distinguished the new approach from previous solutions. Section 3 describes the concepts of alerts integration. Section 4 introduced the overall system architecture. The integration of the alert merging process into Cloud Computing is presented in Section 5. The experimental performance results are reported in Section 6. Finally, we summarize the contributions and comments on further research.

2. Background

2.1 Intrusion Detection System

An IDS analyzes information about the activities produced from networks and seeks for malicious behavior. Detection methods are used by intrusion detection systems in two different ways, according to two different criterions: anomaly detection [4], [10], [11], [12] and misuse detection. In anomaly detection systems, a “normal profile” should be built by historical data about a system’s activity and then use this profile to identify patterns of activity. On the contrary, misuse detection systems [16], [18] are based on specific attack signatures that are matched against the stream of audit data seeking for that malicious attack is occurring. Most of IDS are based on this misuse detection system, such as Snort [19]. Snort is the most popular IDS especially in open source network intrusion detection systems. It can absolutely promote intrusion prevention system. Snort utilizes a rule-driven language, containing the benefits of signatures and anomaly. Snort has become the standard for the industry and many experiments on academic paper are based on it [2], [25], [27].

2.2 Alert Correlation

Alert correlation is an analysis process that takes the alerts generated by IDS and creates reports under its surveillance network. A number of the proposed approaches include a multiphase analysis of the alert stream. For example, the model proposed by Andersson [1] and Valdes et al. [20], [21] presents a correlation process by collecting low-level events using the data of attack threads and using a similarity metric to fuse alerts into merged alerts. This approach depends on a knowledge pool that contains the description of security-relevant characteristics and priorities of these alerts and a format of passive alert verification. [17]

2.3 Cloud Computing

The term "Cloud Computing" means the usage of computer technology (Computing) based on Internet (Cloud). The computing capabilities are provided as a service without knowledge or expertise support. Cloud Computing is the next natural step in the evolution of on demand information technology services and products. Cloud Computing became famous in October 2007 when IBM and Google announced collaboration [14]. This was followed by IBM's announcement of the "Blue Cloud" effort [22]. Until now, Google is one of the leaders in this technology and has built Internet consumer services like search, social networking, Web e-mail and online commerce that use Cloud Computing. The companies such as Yahoo [26] and Amazon [6] also provide great Cloud Computing applications, too.

3. Alert Integration Procedure

Although some correlation approaches have been suggested in section 2.2, there is no consensus on what this process is or how it should be implemented and evaluated. Fortunately, Fredrik [23] et al. proposed a comprehensive approach to integrate alerts. Their experiment results verify this approach with outstanding reduction rate. Condensing this concept, we extract some essence steps of merging alerts and implement it into our analysis system.

As shown in Figure 1, the merging process checks whether raw-alert and meta-alerts could be merged. Initially, raw alert whose key is K_1 with value v_1 and v_2 is the first alert, so it goes into meta-alert directly. Next, the second alert whose key is K_2 approaches. Because each of their keys is not identical, merging process sends it into meta-alerts. After that, merging process combines the third alert with the meta-alert whose key is K_1 and appends the meta-alerts to new valueV_3.

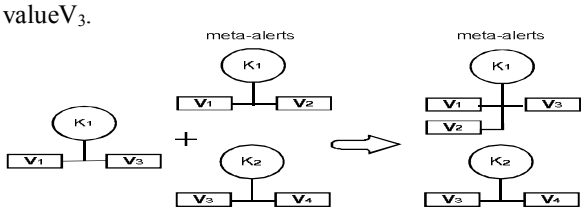
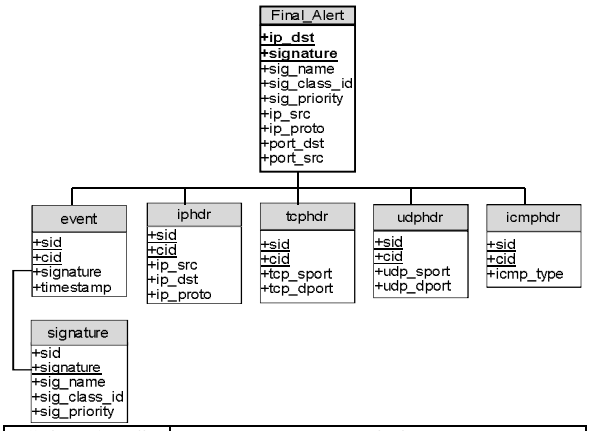


Figure 1. Alert merging process.



"Final_Alert" Field	Description
Ip_dst	Destination IP addresses. (Primary key).
signature	Signature is a unique ID used to identify attack method in Snort rules. (Primary key).
Sig_name	Signature name corresponded to Snort ID.
Sig_class_id	Classification ID of this Snort ID
Sig_priority	Priority of Snort ID
Ip_src	Source IP address.
Ip_proto	TCP/IP protocol.
Src_ports	Attack lunched ports
Dst_ports	Victim ports

Figure 2. Database table and its description.

This predecessor's research supplies a classic idea. At the aspect of implementation, we use Snort with MySQL [15] database and imitate this approach to design an integration process. The Snort alert data stored in MySQL is separated into "event" (associated with "signature" table), "iphdr", "tcphdr", "udphdr" and "icmphdr" tables. In these five tables, both of "sid" and "cid" are composite primary keys used to identify an alert, but both of them would be unessential for merged alert. The other significant fields are designed in "Final_Alert" table shown as Figure 2. By extracting data from original tables, the integration process merges all data overall and inject result into the "Final_Alert" table.

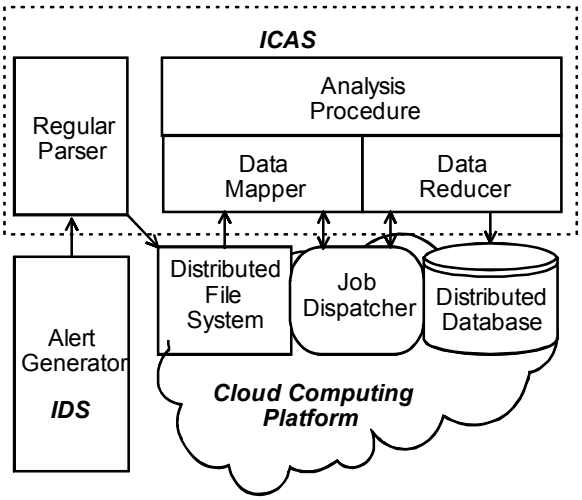


Figure 3. Architecture of ICAS.

4 System Architecture

The IDS Cloud Analysis System (ICAS) is an alert integration system building on the infrastructure of Cloud Computing. Figure 3 shows the overall architecture of this paper proposed. There are three parts in this architecture. The individual components are described as following.

4.1 Intrusion Detection System

Alert Generator should be software designed and network IDS to detect unwanted attempts. At the present time, Snort is the current supported IDS.

4.2 Cloud Computing Platform

Cloud Computing Platform is based on two Apache's free Java software projects: Hadoop [8] and HBase [9]. Hadoop is inspired by Google's MapReduce [5] and Google File System [7] to develop a framework, which including MapReduce and Hadoop Distributed File System (HDFS) supports data intensive distributed applications running on large clusters of commodity computers. HBase is a column-oriented distributed database modeled after Google's BigTable [3]. This Cloud Computing platform is able to work with thousands of nodes and petabytes of data.

Nodes, or naming data nodes, supply blocks of data over the network using a block protocol specific to Hadoop. They can communicate to each other to rebalance data, to control data flow, and to maintain the replication of data.

Distributed File System (HDFS) is a single file system that can be distributed across several nodes connected by network. In contrast to shared disk file systems where all nodes have uniform direct access to the entire storage.

Job Dispatcher (MapReduce) consists of one Job Tracker and several Task Trackers. The Job Tracker controls client applications and pushes work out to available Task Tracker nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware filesystem, the Job Tracker knows which node the data lives on, and which other machines are nearby.

Distributed Database (HBase) provides Bigtable-like capabilities on top of Hadoop. Its goal is the hosting of very large tables including billions of rows with millions of columns.

4.3 IDS-log Cloud Analysis System

Regular Parser normalizes raw IDS log to form a regular form. Each alert in IDS log file contains many statements to specify an accident but ICAS just extracts several important fields described in Figure 2.

Analysis Procedure consists of Data Mapper and Data Reducer. Based on Hadoop architecture, Data Mapper and Data Reducer are adapted for the MapReduce infrastructure.

Data Mapper is applied to parallel every item in the input dataset. This produces a list of (key, value) pairs for each call. After that, the Cloud Computing framework gathers all pairs with identity key from all lists. After that, all pairs are grouped together and separated into several group for each one of the different generated keys.

Data Reducer is applied in parallel to merge data from Data Mapper. After collect results into database.

5. Integrating IDS into Cloud Computing

Figure 4 shows the procedure of ICAS. LOG is a log file produced by alert generator. Regular Parser, Analysis Procedure, Data Mapper, Data Reducer are procedure processes of ICAS. Database is distributed database in cloud architecture. Meta file is a file transferred into distributed file system. All of meta-data are intermediate product of Job Dispatcher between Data Mapper and Data Reducer.

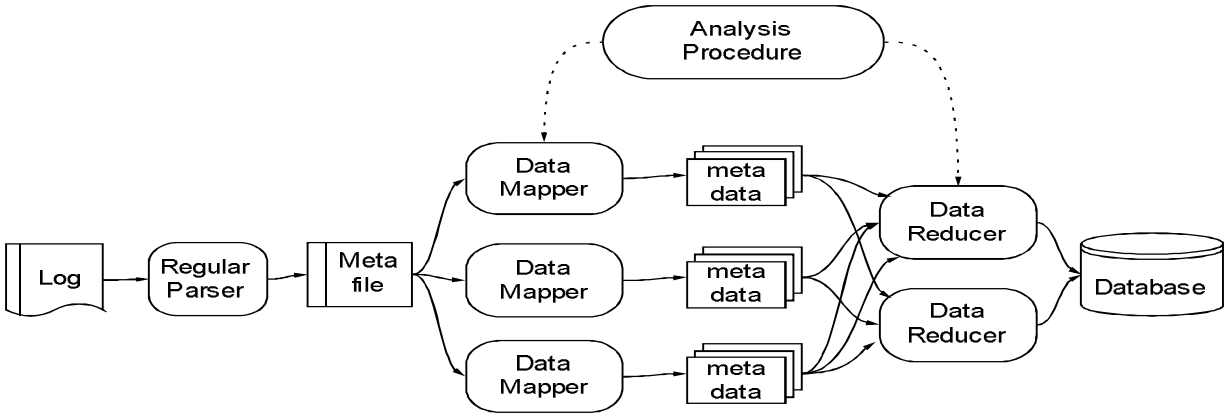


Figure 4. Procedure of ICAS.

At the beginning of procedure, alert generator collects malicious packets and stores information into a log file. However, the log format is not trim. The first component, Regular Parser, extracts the essential information and discards useless data then parses as a regular form. Next step, system would transfers the metafile to distributed file system which splits metafile spread every node. Job Dispatcher lunches Data Mapper and assigns jobs to every node.

The major work of Data Reducer is to reduce the redundancy and merge information. The reduce rule is: if any of two alerts that destination IP and signature are respectively the same, the two would integrate as one and other attributes should be merged. For example, there are six alerts in metadata shown as table 2, and table 3 is the result. I1 and I2 are the same approach because that attacker does the method twice at different time. After analysis job worked, I1 and I2 reduced as R1. The different attribute, I3's b and I4's c, should be merged as R2's {b, c} because they have the same attributes, destination IP and signature. The example is that there are two hosts using the same method to attack one target. Any of the major attributes, signature and destination IP, are different, reduce job would not merge. For example, I5 and I6 are respectively on behalf of R3 and R4. This process is specified in Figure 5. Finally, ICAS stores the results into distributed database. In order to suit for the functionality described in section 3, we set ip_dst as row-key and signature as column-family to simulate "Final_Alert". The other field and its values are gathered in column-qualifier.

Table 1. Initial metadata.

ID	Ip_dst	signature	others
I1	Ip_1	A	{a},{b},…
I2	Ip_1	A	{a},{b},…
I3	Ip_2	A	{a},{b},…
I4	Ip_2	A	{a},{c},…
I5	Ip_3	A	{a},{b},…
I6	Ip_3	B	{a},{b},…

Table 2. Result after reduce.

ID	ip_dst	signature	others
R1	Ip_1	A	{a},{b},…
R2	Ip_2	A	{a},{b, c},…
R3	Ip_3	A	{a},{b},…
R4	Ip_3	B	{a},{b},…

Analysis Pseudo Algorithm:
01: INPUT: meta-data produced by Regular Parser ***Log***
02: generate new structure set S={s_0, s_1,… s_n}
03: map:
04: for each line l_n in ***LOG***, do:
05: parse l_n into structure s_n of {***ip_dst, signature, ip_src, sig_name, sig_class_id, priority, ip_proto, src_port, dst_port***}
06: end for;
07: end map;
08: reduce:
09: loop:
10: select s_a s_b where s_a{ip_dst, signature} is equal to s_b{ip_dst, signature}
11: if s_a is equal to s_b then:
12: delete s_a ;
13: else: merge all s_a's fields to s_b ;
14: end if;
15: until go through whole S
16: end reduce;
17: store S into Database;

Figure 5. Pseudo algorithm.

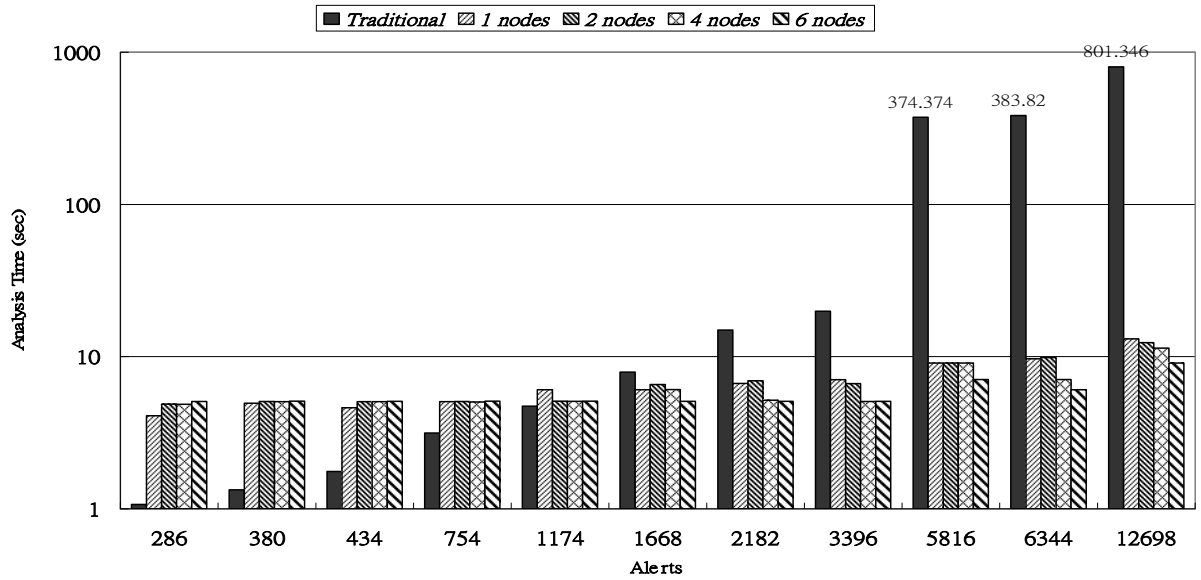


Figure 6. Graph of analysis processing time.

Table 3. Experiment results.

Origianl Alerts	Analysis Time (sec)					Results	Reduction Rate
	Tradional	1 nodes	2 nodes	4 nodes	6 nodes		
286	1.068	4.087	4.869	4.864	5.077	30	89.51%
380	1.333	4.94	5.069	5.067	5.097	11	97.11%
434	1.76	4.61	5.066	5.068	5.09	9	97.93%
754	3.145	5.066	5.079	5.038	5.096	16	97.88%
1174	4.73	6.066	5.093	5.089	5.097	33	97.19%
1668	7.909	6.07	6.56	6.071	5.082	16	99.04%
2182	14.949	6.671	6.95	5.166	5.088	16	99.27%
3396	19.901	7.053	6.654	5.076	5.091	68	98.00%
5816	374.374	9.081	9.076	9.07	7.076	66	98.87%
6344	383.82	9.68	9.872	7.069	6.069	72	98.87%
12698	801.346	13.096	12.367	11.367	9.083	36	99.72%

6. Experimental Result

As described in section 3, the evaluation result of that method is shown as “Traditional” in Figure 6. The other information is the performance of ICAS computing on different numbers of nodes, logged as 1, 2, 4 and 6 nodes. The traditional method and ICAS use similar integration algorithm and result table format “Final_Alert” in Figure 2. The main distinction between the two methods is single and distributed architecture. The other notable difference is that traditional method lets Snort generate alerts into MySQL then gets data from database and the ICAS gets data from Hadoop file system by parsing raw Snort alert files.

Figure 6 plots the analysis processing time that is produced by the traditional method and the ICAS using 11 data sets. These data sets, named as its amount of alerts, are generated and released by MIT Lincoln Laboratory [13] and professor Wu’s library of U.C.Davis [24]. Each experimental machine is equipped with: Intel Core 2 Quad 2.4GHz CPU, 2 Gigabytes DDR2 667 memories, 7200 RPM SATA-1 hard disk and 1Gigabits network bandwidth.

The result in Figure 6 points out that the traditional method could complete its works rapidly while alert number is fewer than 1174, but this method would spend a long period to digest more than five thousand alerts. On the contrary, ICAS has an average and short processing time between 4 to 13 seconds. It is worth mentioning that equipping more nodes owes better analysis capability on general condition. However, if alerts were fewer than 1 thousand, the ICAS with more nodes would spend more time to handle. The reason is that larger number of nodes should spend more time to communicate with each other. All detailed experiment data including reduction rate is shown in Table 3. It proves several benefits of this system such as more than 89 percent of the reduction rate and less than 14 seconds of the processing time in our experiment.

7. Conclusions and Further Research

This paper explains the architecture and software design of ICAS, an IDS log analysis system based on Cloud Computing architecture. This paper supplies an idea about Cloud Computing technique in security area.

By viewing the experimental result, the ICAS is proved with high reduction rate and computing ability. Actually, the significant benefits to build IDS analysis system on Cloud platform are its scalability and reliability. Many aspects of this research need to be improved and expanded in the future. Several avenues of this work remain open.

Supporting More IDS Type. At present time, the Snort is the only IDS supported by ICAS, but we would extend the ability of Regular Parser to deal with more IDS log type.

Easily Readable Final Report. The final report is still a simple format, it needs to be integrated more element, such as attack verifications, suggestion approaches … and so on.

Enhancing and Optimizing System. The algorithm of ICAS should be optimized and improved to get more efficient performance.

Application on Broader Area. The cloud architecture is provided with amazing computing ability, we should design more functions to fit its properties.

References

[1] D. Andersson, M. Fong, and A. Valdes, "Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis," Third Ann. IEEE Information Assurance Workshop, Jun. 2002.

[2] M. Attig and J. Lockwood, "A Framework for Rule Processing in Reconfigurable Network Systems", 13th Annual IEEE Symposium, Apr. 2005.

[3] F. Chang, J. Dean and S. Ghemawat, "Bigtable: A Distributed Storage System for Structured Data", OSDI 2006, Dec 2006

[4] D. E. Denning, "An Intrusion Detection Model," IEEE Transaction on Software Engine, Feb. 1987.

[5] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," OSDI 2004, Dec. 2004.

[6] EC2, "Amazon Elastic Compute Cloud" http://www.amazon.com/gp/browse.html?node=201590011, 2008

[7] S. Ghemawat, H. Gobioff and S. Leung, "The Google File System" SOSP 2003, Dec. 2003.

[8] D. Borthakur, "The Hadoop Distributed File System" http://lucene.apache.org/hadoop, 2008.

[9] Hbase. "Hbase: Bigtable-like structured storage for hadoop hdfs." http://wiki.apache.org/lucene-hadoop/Hbase, 2008

[10] H.S. Javitz and A. Valdes, "The NIDES Statistical Component Description and Justification," Technical Report, SRI Int, Mar. 1994.

[11] C. Ko, M. Ruschitzka, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach," IEEE Symp. Security and Privacy, May. 1997.

[12] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," CCS '03, Oct. 2003.

[13] MIT Lincoln Laboratory, Lincoln Lab Data Sets, http:// www.ll.mit.edu/IST/ideval/data/data_index.html, 2000.

[14] S. Lohr, "Google and I.B.M. Join in 'Cloud Computing' Research," Oct. 2007.

[15] MySQL, The open source database, http://www.MySQL.com/, 2008

[16] P.G. Neumann and P.A. Porras, "Experience with EMERALD to Date," First USENIX Workshop Intrusion Detection and Network Monitoring, Apr. 1999.

[17] P. Porras, M. Fong, and A. Valdes, "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation," the Recent Advances in Intrusion Detection, Oct. 2002.

[18] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Seventh USENIX Security Symp., Jan. 1998

[19] M. Roesch, "Snort—Lightweight Intrusion Detection for Networks,"Proc. USENIX LISA '99 Conf., Nov. 1999

[20] A. Valdes and K. Skinner, "An Approach to Sensor Correlation,"Proc. Recent Advances in Intrusion Detection, Oct. 2000.

[21] A. Valdes and K. Skinner, "Probabilistic Alert Correlation," Recent Advances in Intrusion Detection, Oct. 2001.

[22] M. A. Vouk, "Cloud Computing – Issues, Research and Implementations," Proceedings of the ITI 2008, Jun. 2008.

[23] Fredrik Valeur, Giovanni Vigna and Richard A. Kemmerer, "A Comprehensive Approach to Intrusion", IEEE Transactions on Dependable and Secure Computing, Jul. 2004

[24] Felix Wu laboratory, "TCPdump Data Sets," http://www.cs.ucdavis.edu/%7Ewu/tcpdump/, 2005

[25] Y. S. Wu, B. Foo, Y. Mei, and S. Bagchi. "Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS," Computer Security Applications Conference, Dec. 2003.

[26] Yahoo, "Hadoop and Distributed Computing at Yahoo!" http://developer.yahoo.com/blogs/hadoop/, 2008

[27] A. T. Zhou, J. Blustein, and N. Zincir-Heywood. "Improving Intrusion Detection Systems through Heuristic Evaluation;" Electrical and Computer Engineering, May. 2004.