Architecture and Mechanisms for Implementing an FPGA-based Stateful Intrusion Detection System

Jin-Tae Oh†, Byoung-Koo Kim†, Seung-Yong Yoon†, Jong-Soo Jang†, and Yong-Hee Jeon††,

† Applied Security Group, Information Security Research Division, ETRI, Daejeon, Korea
†† Catholic University of Daegu, Gyeongsan, Gyeongbuk, Korea

Summary
This paper proposes Gigabit IDS to detect and respond against various attacks on high-speed links. Our proposed system has hardware-based stateful intrusion detection architecture that can provide the high-performance detection mechanism. It is possible through the pattern matching and heuristic analysis functions that are processed in FPGA Logic. In this paper, we propose architecture designed to perform intrusion detection on high-speed links with reduced false positive rates. We then present the efficient detection mechanisms for the FPGA-based Reconfiguring Hardware. It is revealed that the prototype has a consistent performance with varying traffic level.

Key words: *IDS(Intrusion Detection System), network security, SPI(Stateful Packet Inspection), Pattern Matching*

1. Introduction

Networks have grown in both size and importance over the last a few decades. As the networks are growing and expanded, the problem of unauthorized access and tampering with data is also increasing. In order to encounter with increased threats, many Intrusion Detection Systems(IDSs) have been developed to serve as the last line of defense in the overall protection scheme of computer and network systems.

Basically, IDS is classified into host-based IDS and network-based IDS[1]. Audit sources discriminate the type of IDSs based on the input information they analyze. Host-based IDS analyzes host audit source, and detects intrusion on a single host[2]. With the widespread use of the Internet, IDSs have become focused on network attacks. Therefore, most IDSs employed network-based IDS(NIDS). Network-based IDS uses the network as the source of security-relevant information. Consequently, network-based IDS moves security concerns from the hosts and their operating systems to the network and its protocols.

The IDS can also be classified into two major approaches based on the detection method they operate; misuse intrusion detection and anomaly intrusion detection[3]. The first, misuse intrusion detection is based on the detection of intrusions that follow well-defined patterns of attack exploiting known vulnerabilities. Therefore, this approach is based on the pattern of known misuse or abnormal behavior. This approach is very efficient, but it is hard to detect new intrusion patterns. This approach is also possible to draw false negative detection. Most of existing NIDSs, such as Snort[4], NFR[5], and NetSTAT[6], only employ the misuse detection approach for reducing the degradation of performance to the minimum.

The anomaly detection is based on the detection of anomalous behavior or the abnormal use of the computer resource. This approach is based on the database of normal behavior. Therefore, it costs a great deal, but this approach is capable of detecting unknown intrusions. This approach is possible to draw false positive detection, but hard to set a threshold value. Whichever approaches we adopt, they all have its own advantages and disadvantages, respectively. However, anomaly intrusion detection approach may not easily be adopted in real-time intrusion detection, since it tends to be computationally intensive because of several maintained metrics that are updated after each system activity. Therefore, most IDSs have employed misuse detection approach due to the performance and availability consideration.

Misuse detection might be implemented by one of the following primary techniques: Expert System, State Transition Analysis, Model-based Approach and others [7]. But, these techniques do not present the definite mechanism, and sometimes contain complex and ambiguous concepts. Also, these approaches are not suitable as a speedy detection mechanism in high-speed network environments. Therefore, most IDSs focus on more speedy and exact pattern matching algorithm and detection mechanism against Denial of Service(DoS) attacks and Port Scan attacks.

This paper proposes Gigabit IDS to detect and respond against attacks on the high-speed network. It is possible through the function that is processed in FPGA(Field Programmable Gate Array) Logic. The proposed system has hardware architecture that can provide efficient ways to detect and respond against various attack behaviors in

Manuscript received May 5, 2007
Manuscript revised May 20, 2007

high-speed and high volume large-scale networks. In order to reduce false positive alerts, we adopt Stateful Packet Inspection(SPI) in our system.

The remainder of the paper is structured as follows. The next section presents related works on IDS and SPI. Then, section 3 presents the architecture of our Gigabit IDS, and describes major FPGA components. Section 4 presents the efficient detection mechanisms for the FPGA-based Reconfiguring Hardware including SPI mechanism. Section 5 describes a prototype that we have implemented, and some experimental results of the system are given. Finally, we summarize this paper and conclude in section 6.

2. Related Works

Most of NIDSs based on misuse detection approach has concentrated on catching and analyzing only the audit source collected on Fast Ethernet links. With the advancement of network technology, Gigabit Ethernet has become the actual standard for large network installations. As the result, the existing NIDSs have problems in performance as ever, such as bottleneck, overhead in collecting and analyzing data in a specific component. Accordingly the effort of performing NIDS on high-speed links has been the focus of much debate in the intrusion detection community. In efforts to provide intrusion detection capability with enhanced performance, several NIDSs, such as RealSecure[8], ManHunt[9], and CISCO IDS[10], which have a target to run on high-speed links were developed. Nonetheless these NIDSs is still not practical because of technical difficulties in keeping pace with the increasing network speed, and real-world performance also will likely be degraded. Therefore, there is an emerging need for security analysis techniques that can keep up with the increased network throughput[11].

Besides the performance problem with NIDS, another major problem is the high false positive alert rate. In order to reduce these false positive alerts, a lot of methods and techniques are proposed. SPI is one of these solutions. SPI was originally developed for Firewall[12], but it became a very important factor in NIDS. Stateless NIDSs generate tremendous false positive alerts while stick or snot attempts to attack[13]. Most existing NIDSs have SPI module which supports statefulness but they don't satisfy high-performance in gigabit Internet environment. It is very difficult to manage a lot of session state information with limited hardware resource and to satisfy performance of high-speed Internet.

To guarantee both performance and functionality with respect to stateful intrusion detection, we designed and implemented SPI-based intrusion detection module in a FPGA to help alleviating a bottleneck in network intrusion detection systems. Stateful intrusion detection is performed by SPI-based intrusion detection module. The performance of stateful intrusion detection system mainly depends on the performance of processing session table[14] and pattern matching[15]. In this paper, therefore, we describe session state management and pattern matching methods in detail.

3. Intrusion Detection Architecture

3.1 System Architecture

The architecture of our system, named "Next Generation Security System"(shortly NGSS) is illustrated in Figure 1. We designed it to detect intrusions on high-speed links at real-time basis. The NGSS has two main systems: Security Gateway System(SGS) and Security Management System(SMS). Our SPI-based intrusion detection module is implemented in Security Gateway System(SGS). The SGS is a security node system that may be positioned at an ingress point in protected networks. Security policies from Security Management System(SMS) are applied and executed in the SGS.

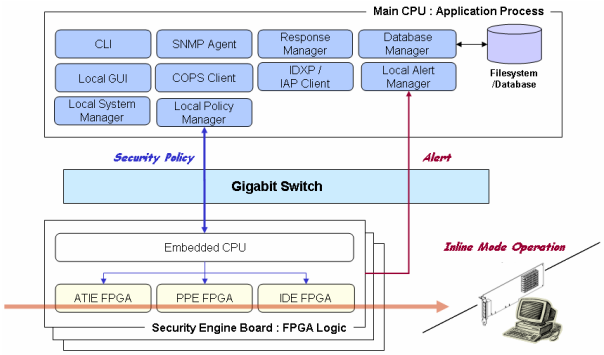

Fig. 1 Security Gateway System(SGS)

The SGS is aimed at real-time network intrusion detection based on misuse detection approach. As shown in Fig.1, SGS consists of three parts; Application Process for communication channel with SMS and system management functions, FPGA Logic for wire-speed packet forwarding, packet preprocessing, high-performance stateful intrusion detection and others, Gigabit Switch for communication between FPGA Logic and Main CPU. We can divide FPGA Logic into several sub FPGA Logics. Most of all, the summary of the internal FPGA Logics for detection operation is given in the following;

- ATIE(Anomaly Traffic Inspection Engine) FPGA : this FPGA Logic performs the wire-speed forwarding, and generates the alert message according to the detection result from IDE FPGA. This Logic also

performs the response function according to the response strategy such as rate limiting, packet filtering, and so on.

- PPE(Pre-Processing Engine) FPGA: this FPGA Logic performs the preprocessing function such as protocol normalization, IP de-fragmentation, TCP reassembly and session management for SPI-based intrusion detection, which is prior steps for intrusion detection .
- IDE(Intrusion Detection Engine) FPGA: this FPGA Logic performs the intrusion detection function such as pattern matching and traffic volume-based packet measuring. Most of all, this Logic includes the effective detection mechanisms for high-performance intrusion detection. Briefly, it performs the detection operation without the packet loss.

Further details on these FPGA chips are given in the following section.

3.2 Security Engine Board

Embedded CPU on the Security Engine Board manages the ruleset that is required for intrusion detection. Through the interoperability of these components, SGS analyzes data packets as they travel across the network for signs of external or internal attacks. Namely, the major functionality of SGS is to perform the real-time traffic analysis and stateful intrusion detection on high-speed links. Therefore, we focus on effective detection strategies applied to FPGA Logic.

The Security Engine Board of our system is composed of three FPGA Chips and one Embedded CPU. As shown in the figure 2, one is ATIE(Anomaly Traffic Inspection Engine) FPGA Chip for wire-speed packet forwarding and blocking, another is PPE(Pre-Processing Engine) FPGA Chip for packet preprocessing, and the other is IDE(Intrusion Detection Engine) FPGA Chip for high-performance intrusion detection.

First, ATIE FPGA Chip uses the Xillinx FF-1517 FPGA Chip. It is connected to the PM3386 for incoming packet processing and PM3387 for alert message sending. It uses two external TCAM and two external SRAM for incoming packet scheduling and management. As aforementioned, the main function of ATIE FPGA Chip is the wire-speed packet forwarding and response coordinating such as alert response and packet filtering. Incoming Packets from PM3386 is sent to PPE FPGA Chip, and if it is determined as attack according to the analysis result from other FPGA Chips, alert message is sent to the Main CPU.

Second, PPE FPGA Chip uses the Xillinx FF-1152 FPGA Chip. And, it uses two external TCAM and four external SRAM for operating the session management, IP de-fragmentation and TCP reassembly.

The main function of PPE FPGA Chip is the preprocessing as a previous step for intrusion detection. The preprocessing function supports the SPI(State-ful Packet Inspection) based intrusion detection and IDS evasion attack detection.

Also, it checks out the protocol validation about service protocol such as SMTP, HTTP. If the incoming packet is invalid against service protocol, alert information is sent to ATIE FPGA Chip.

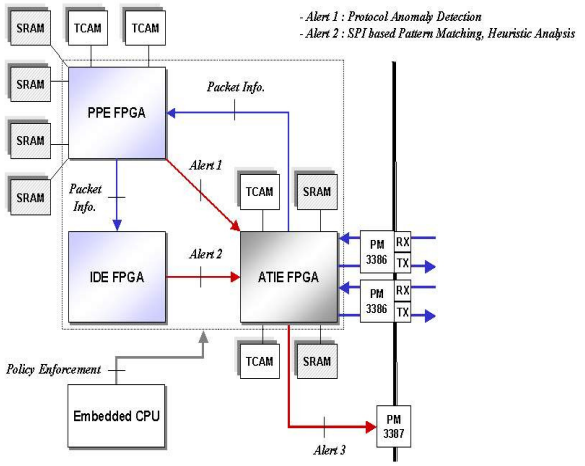

Fig. 2 The FPGA Chip Arrangement of Security Engine Board

Finally, IDE FPGA Chip uses the Xillix FF-1517 FPGA Chip. It uses three mechanisms for high-performance intrusion detection: Flexible Header Combination Lookup Algorithm for packet header pattern matching, Linked Word based Store-less Running Search Algorithm for packet payload pattern matching, and Traffic Volume based Analysis Algorithm for DoS and Port-scan attack detection. Through these mechanisms, it supports the high-performance intrusion detection function without packet loss.

4. Intrusion Detection Mechanisms

In our system, stateful intrusion detection is performed by SPI-based intrusion detection module on PPE and IDE FPGA in Fig.1. The performance of stateful intrusion detection system mainly depends on the performance of processing session table and intrusion detection strategy. In this section, therefore, we describe session state management and pattern matching mechanism in detail.

4.1 SPI mechanism

Figure 3 shows the SPI-based intrusion detection module in the SGS. Legitimate TCP sessions are established through 3-way handshakes and terminated through 4-way

handshakes. State manager has session state table and tracks these session states. If input packet doesn't exist in the session entries, the packet will be dropped or forwarded to IDE with additional state information according to security policies.

At first, for the packet from IP De-fragmentation sub-module, necessary information fields are extracted through packet parser. Packet filter transfers all packets that are passed by security filtering policies to State Manager. These filtering policies can be applied to the specific protocols or ports. The State Manager manages session state table and current session status. TCP Reassembly sub-module reassembles the TCP segments in the right order. SPI Information Generator generates and transmits useful SPI information for detecting attack and abnormal packet to Intrusion Detection Engine. The SPI information includes session establishment, flow direction, and others. For example, session establishment indicates whether the packet is a part of an established TCP Session or not. Flow direction indicates whether the packet is for sending from client to server or not. Finally, Intrusion Detection Engine performs effective pattern matching with SPI information transmitted from SPI Information Generator.

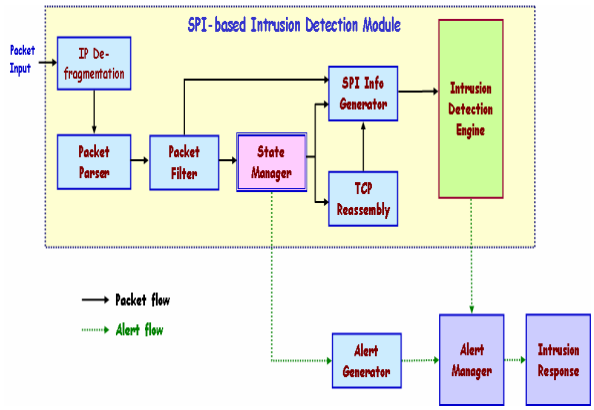

Fig. 3 SPI-based Intrusion Detection Module

4.2 State Management Mechanism

The terms "session", "connection" and "flow" are used interchangeably in this paper. State information is separately managed in embryonic connection table and is established by the State Manager. State table is composed of 4-level linked lists, which are IDLE, SYN, SYN/ACK, and EST Links. Embryonic connection table (SYN, SYN/ACK Link) includes sessions that TCP 3-way handshaking does not finish, while the established connection table(EST Link) manages completed sessions. It is necessary to manage two tables separately, because embryonic connection table needs to have shorter timeout value than that of established connection table. This scheme helps to prevent against denial-of-service(DoS) attack such as SYN flooding. In addition, if TCP flags are SYN or SYN/ACK, only session table lookup is performed in embryonic connection table. This considerably helps to reduce the session table lookup time.

The SPI devices and computers have vulnerabilities to SYN flooding attack in nature[16]. So SGS must have a mechanism against such an attack. Whenever the session entry is accessed, the entry moves to the position of tail in its link. Because the resource of session table is limited, the possibility that state table is full always exists. Although the state table is full, new attack must be detected constantly. Therefore, if the state table is full, the entry that timeout occurred will be replaced by a new one(in order of link level: SYN, SYN/ACK, and EST Link). And then, if the entry that timeout occurred doesn't exist, the new session entry will be allocated instead of existing one according to the LRU(Least Recently Used) algorithm on EST Link. Because the most idle entry is the first entry(the position of head) on EST link, this entry is chosen and replaced as new one. Existing session state table architectures of SPI devices store all session information in single entry, which causes high time cost of session table processing. But our session table has a new architecture in which a session entry is divided into two parts. That is, session state table consists of Session Index Table and State Information Table. Figure 4 shows the relationship of LRU Logic, Session Index Table and State Information Table.

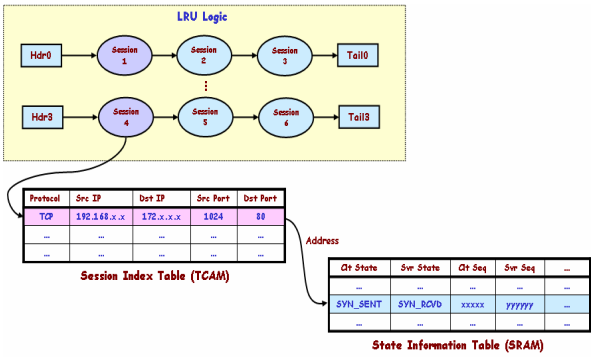

Fig. 4 The Architecture of State Table

Session index table uses one CYNSE70256 TCAM(9Mbits) for high speed session entry searching and state information table uses one Cypress SRAM(2Mbytes) to store state information. Each session entry in session index table matches to state information in state information table one to one. Linked lists for LRU Logic are implemented using internal Block SelectRAM in Xilinx FPGA Chip. TCAM can be configured to contain tables of different widths. Because TCAM capacity is 9Mbits, we can configure up to 128K entries in 72-bit configuration or up to 64K entries in 144-bit configuration.

One session entry in session index table is composed of 5-tuple that is 104bits in length. Therefore, we chose 144-bits configuration and could manage up to 64K entries. Stored state information per session entry on SRAM is 256bits in length. State manager can detect attack and abnormal packet by tracking sequence number ant ttl in state table. If the sequence number of current packet is smaller than last received acknowledge number, alert should be generated. Also, when the payload size of packet is out of window size, alert should be generated. Packets in the same session should generally have about the same number of routers to traverse on their way between the two hosts. If the number of hops changes too drastically, it might be a sign of someone trying to evade detection. So if their difference is over the specific threshold, alert should be generated.

4.3 Detection Mechanisms

The detection mechanism of our system is mainly performed on IDE FPGA Chip. For effective high-performance intrusion detection, our system has three detection mechanisms. One is the header lookup mechanism for flexible header combination lookup, another is the payload matching mechanism for packet payload matching, and the other is heuristic analysis mechanism for DoS and Port-scan attack detection.

4.3.1 Header Lookup Mechanism

Header lookup mechanism is performed by flexible header combination lookup algorithm. This algorithm compares pre-defined header related rule-sets with header information of incoming packets. If the incoming packet is matched with existing header patterns, 256 bits match result is sent to payload matching logic and traffic volume based heuristic analysis logic. As shown in Figure 5, this mechanism uses three memory maps: TCAM Lookup Map for each header field matching, Rule Combination Check Map for multiple header field matching, and Sequence Check Map for don't care filed matching.

First, TCAM Lookup Map is composed of three internal TCAM: 8bits lookup map for 8bits header fields such as ICMP type, TCP flags, and so forth, 16bits lookup map for 16bits header fields such as service port value, IP identification, and so forth, and 32bits lookup map for 32bits header fields such as IP address field. These maps have each 128entries, 64entries. In other words, our system supposes that header values of all rule-sets are within the entry number of each map. The match result of these lookup maps is used by Rule Combination Check Map and Sequence Check Map.

Second, Rule Combination Check Map is composed of 256*256 block select RAM memories. Match address from TCAM Lookup Map is used for 256bits result of this memory map. This 256bits result presents the rule-set matching result about current matching field. If match result of ICMP type field is "{255{2'b0}, 2'b1}", the first rule-set is to be matched. Therefore, it is possible to support the multiple matching. In other words, our system supposes that the combination of all rule-sets is within the 256 entry numbers.

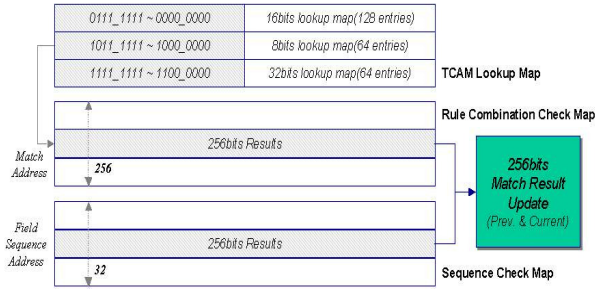

Fig. 5 Header Lookup Mechanism

Finally, Sequence Check Map is composed of 32*256 block select RAM memories, and includes don't care information about current matching field. If don't care information of ICMP type field is "{255{2'b0}, 2'b1}", the first rule-set is always to be matched. In other words, our system supposes that the kinds of packet header fields are within the 32 entry numbers. The result of this map is combined with result of Rule Combination Check Map.

Basically, the above operations are performed recursively about all packet fields of incoming packet. Finally, updated 256bits match result is referred by logics for packet payload matching and traffic volume based analysis.

4.3.2 Payload Matching Mechanism

Payload matching mechanism is performed by linked word-based store-less running search algorithm. This algorithm compares pre-defined packet payload related rule-sets with packet payload information of incoming packets. If the incoming packet is matched with existing payload patterns, alert message is generated according to the 256bits header lookup result. For this operation, this algorithm uses the pattern reconstruction technique. As shown in Figure 6, reconstruction pattern length has boundary of size5 or 7 because of the limit of block memory in FPGA Chip. The first 5bytes of "/bin/echo" pattern are equal to the first 5bytes of "/bin/kill" pattern and "/bin/chmod" pattern. Therefore, "/bin/" string of these patterns is stored in the same memory space. Like this, other patterns are reconstructed. Through pattern reconstruction like this, our system can have about 2,000~3,000 rule-sets in the limited memory storage on FPGA Chip.

After above pattern reconstruction, linked word based store-less running search is performed as payload matching mechanism. This mechanism uses the spectrum dispersion technique as shown in Figure 7. The spectrum dispersion technique is method to calculate unique hash values about reconstructed patterns. For example, 5bytes "/etc/" pattern has the 9bits hash value "011010011" by sum about shifted values of each character. These hash values are used as the rule memory address for each pattern.

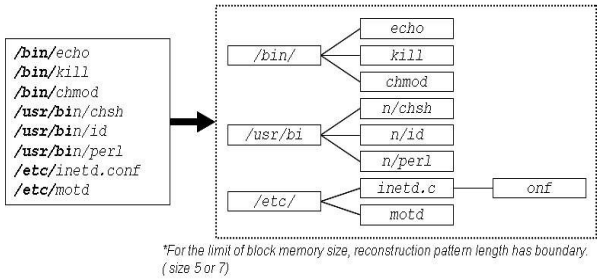

Fig. 6 Pattern Reconstruction for Payload Pattern Matching

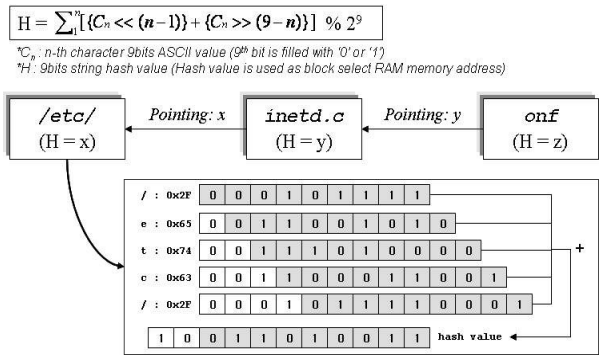

Fig. 7 Linked Word based Store-less Running Search

The memory construction is performed by Embedded CPU when the system is being booted in advance. After system booting, IDE FPGA Logic performs the hash value calculation on the incoming packet to the unit of byte. If the payload in incoming packet is matched with the pattern in memory pointed by the calculated hash value, then it is checked out whether the related reconstructed patterns is matched or not. If all reconstructed patterns are matched with incoming packet, alert message is generated according to the header lookup results. Through these operations, our system performs the pattern matching operation without lowing of performance and packet loss.

4.3.3 Traffic Volume-based Heuristic Analysis Mechanism

Traffic volume-based analysis mechanism is performed by traffic volume-based heuristic analysis algorithm. Similar to the pattern matching mechanism, this algorithm also compares pre-defined rule-sets with packet information of incoming packets. But, this mechanism is based on the traffic volume. In other words, this mechanism generates alert message by traffic volume within time threshold. As shown in Figure 8, if the incoming packet is matched with existing rule-set, then count value of the rule-set is increased, and count threshold and time threshold is checked out. If the count threshold exceeds by the incoming packet within the time threshold, alert message is generated. Through these operations, our system is capable of detecting the DoS and Port-scan attacks such as TCP Sync Flooding attack, UDP Bomb, SYN/ACK/XMAS Port-scans, and others.

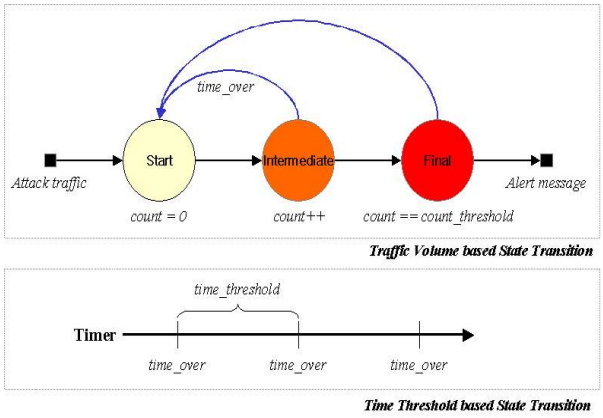

Fig. 8 Traffic Volume-based Heuristic Analysis

5. Implementation and Experiments

5.1 Implementation

We have developed our SGS prototype based on the NGSS architecture, as shown in Figure 9. The prototype we have developed is programmed in a combination of Java and C, Verilog programming language. Most of all, SGS is implemented in programming languages that is best suited for the task it has to perform. Basically, application processes of SGS are implemented in C programming language. FPGA Logic of SGS is implemented on a Xilinx Vertex-II Pro XC2VP70 FPGA(7M Gate)[17] using Verilog HDL(Hardware Description Language) that is best suited for high-speed packet processing. The simulation of all functions were conducted by the ModelSim PE 6.1 simulator[18]. And all logics have been synthesized by Symplify Pro 8.1 tool[19].

In our prototype, FPGA Logic performs many functionalities, such as wire-speed forwarding, 5-tuple(Protocol, source/destination IP Address, source/destination Port) based packet blocking, and

intrusion detection related functions. Most of all, it performs the SPI-based intrusion detection function through the interoperability of PPE FPGA and IDE FPGA functions. Also, it is capable of detecting the IDS evasion attempting by performing the detection operation through the IP de-fragmentation and TCP reassembly.

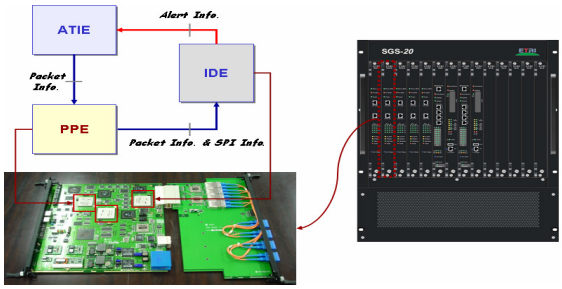

Fig. 9 SGS Prototype

Especially, the prototype we have developed focus on FPGA logic for real-time traffic analysis and stateful intrusion detection on high-speed links. Also, we employed inline mode capable of effective response by using four Gigabit Ethernet links as shown in Fig. 9. The minimum clock period for data from input to output is 8ns which corresponds to a throughput of 2Gbps. That is, our system is capable of processing until the maximum throughput of full-duplex 2Gbps on incoming packets in FPGA Logic.

5.2 Experimental Results

For performance evaluation of our prototype system, we applied Snort ruleset to SGS. And we used IXIA Traffic Generator[20], Gigabit Switch, IDS Informer Attack Tool[21] and Nessus Vulnerability Scanner[22] for experiments.

At first, we generate and transmit packet to the test bed using IXIA Traffic Generator. Then we simulate attacks by Nessus and IDS Informer. While background traffic generated by IXIA is increasing gradually, we observed the rate of alert generation. The rule-set used in the system includes 200 rules. Fig. 10(a) shows that increasing traffic rate does not have an effect on detection rate of SGS. This result illustrates that the detection performance of our system is not degraded along with the increased traffic level.

The second experiment was to run SGS with a constant traffic rate of 100Mbps and an increasing number of signatures. The experiment starts with only the 200 signatures that are needed to achieve maximum detection for the given attacks. Fig. 10(b) shows the results of this experiment. Also, increasing number of signatures does not have an effect on detection rate of SGS. The experimental scenario used in these experiments are previously used in [11] for Snort sensor. Compared with the Snort sensor, our prototype system shows a consistent performance in traffic level and has nothing to do with increasing number of signatures used.

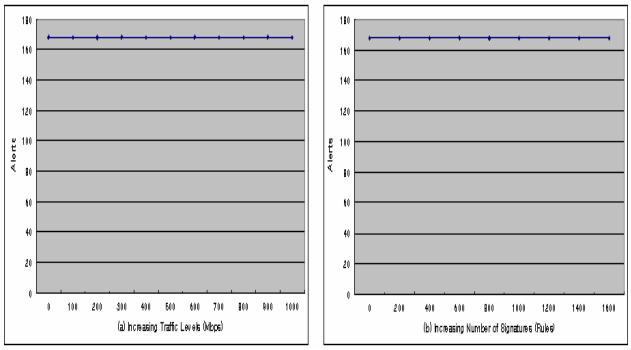

Fig. 10 Results for Detection Rate

6. Conclusions

The most eminent problems in the existing IDS system consist of two-fold: performance and false alerts. In order to perform the real-time traffic analysis and intrusion detection on high-speed links, we proposed architecture for implementing the detection mechanism in FPGA-based reconfiguring hardware. In this paper, we presented the system architecture for the prototype of our system developed for the traffic analysis carried in a Gigabit link.

The other major problem and limiting factor with NIDS is the high false alert rate. In order to reduce the false rate, we proposed SPI-based IDS module. Even though SPI-based intrusion detection module in network security system is implemented for more accurate intrusion detection, if not satisfied with performance, they may not be used. Therefore both problems are correlated.

The performance of stateful intrusion detection system mainly depends on the performance of processing session table and pattern matching. To guarantee both high-performance and functionality with respect to stateful intrusion detection, we designed and implemented SPI-based intrusion detection module in a FPGA to help alleviating a bottleneck in network intrusion detection systems. Some experiment results are given for the prototype. Our proposed system showed a consistent performance with varying traffic level.

References

[1] H. Debar, M. Dacier and A. Wespi, "Research Report Towards a Taxonomy of Intrusion Detection Systems", Technical Report RZ 3030, IBM Research Division, Zurich Research Laboratory, June, 1998.

[2] D. Anderson, T. Frivold and A. Valdes, "Next-generation intrusion detection expert system(NIDES)", Technical Report SRI-CLS-95-07, May, 1995.
[3] .S. Kumar and E. Spafford, "A pattern matching model for misuse intrusion detection", In Proceedings of the 17th National Computer Security Conference, pp. 11-21, Oct., 1994.
[4] M. Roesch. "Snort-Lightweight Intrusion Detection for Networks". In Proceedings of the USENIX LISA '99 Conference, November, 1999.
[5] Marcus Ranum, "Burglar Alarms for Detecting Intrusions", NFR Inc., 1999.
[6] Thomas Ptacek and Timothy Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Secure Networks Inc., 1998.
[7] S. Kumar, "Classification and Detection of Computer Intrusions", Ph D, Purdue University, 1995.
[8] Tarek Abbes, Alakesh Haloi, and Michael Rusinowitch, High Performance Intrusion Detection using Traffic Classification, AISTA 2004 in Cooperation with the IEEE Computer Society Proceedings, Nov. 15-18,. 2004,
[9] Sarang Dharmapurikar, Praveen Krishnamurthy, T.S. Sproll and J.W. Lockwood, Deep packet inspection using parallel bloom filters, IEEE Micro, Volume 24, Issue 1, Pages:52-61, Jan.-Feb.2004
[10] Z.K. Baker and V.K. Prasanna, Time and area efficient pattern matching on FPGAs, In proceeding of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays, pages 223-232, ACM Press, 2004
[11] C. Kruegel, F. Valeur, G. Vigna, and R. Kemmerer, Stateful Intrusion Detection for High-Speed Networks, in Proceedings of the IEEE Symposium on Research on Security and Privacy, Oakland, CA, IEEE Press, May 2002.
[12] http://www.checkpoint.com, Firewall-1 Product.
[13] Brian Caswell, Jay Beale, James C. Foster, Jeremy Faircloth, Snort 2.0 Intrusion Detection(Syngress Publishing, February 2003)
[14] Xin Li, Zheng-Zhou Ji, and Ming-Zeng Hu, Stateful Inspection Firewall Session Table Processing, Proc. of the International Conference on Information Technology: Coding and Computing(ITCC'05), Volume 2, pp. 615-620, April 2005.
[15] Sergei et al., SNORTRAN: An Optimizing Compiler for Snort Rules, Fidelis Security Systems, Inc., 2002.
[16] Hyogon Kim, Jin-ho Kim, Inhye Kang, and Saewoong Bahk, Preventing Session Table Explosion in Packet Inspection Computers, IEEE Transaction on Computers, Vol. 54, No. 2, February 2005.
[17] http://www.xilinx.com
[18] http://www.model.com
[19] http://www.synplicity.com
[20] http://www.ixiacom.com
[21] http://www.bladesoftware.net
[22] http://www.nessus.org

Jin-Tae Oh received the B.S and M.S degrees in Electronics Engineering from Kyungpook National University in 1990 and 1992, respectively. He worked at ETRI (Electronics and Telecommunications Research Institute) from 1992 to 1998. During 1998-1999, he stayed in MinMax Tech , USA, as a Research staff. He served as a Director in Engedi Networks, USA, during 1999-2001. He was both Co-founder and CTO Vice President in Winnow Tech. USA during 2001-2003. From 2003, he works with the Security Gateway Team, ETRI, Daejeon, Korea.

Byoung-Koo Kim received the B.S. and M.S. degrees in Information and Communication Engineering from Sungkyunkwan University in 1999 and 2001, respectively. Since 2001, he has stayed in Security Gateway System Team, Electronics and Telecommunications Research Institute(ETRI) of Korea to study Network Security related Topics.

Seung-Yong Yoon received the B.S. and M.S. degrees in Computer Engineering from Chungnam National University in 1999 and 2001, respectively. Since 2001, he has stayed in Security Gateway System Team, Electronics and Telecommunications Research Institute(ETRI) of Korea to study Network Security related Topics.

Jong-Soo Jang received the B.S and M.S degrees in Electronics Engineering from Kyungpook National University in 1984 and 1986, respectively. He received his Ph. D degree in Computer Engineering from Chungbuk National University in 2000. Since 1989, he has been working with ETRI, Daejeon, Korea and now is the Director of Applied Security Group.

Yong-Hee Jeon received the B.S degree in Electrical Engineering from Korea University in 1978 and the M.S and Ph. D degrees in Computer Engineering from North Carolina State University at Raleigh, NC, USA, in 1989 and 1992, respectively. From 1978 to 1985, he worked at Samsung and KOPEC(Korea Power Engineering Co.). Before joining the faculty at CUD in 1994, he worked at ETRI(Electronics and Telecommunications Research Institute) from 1992 to 1994. Currently, he is a Professor at the School of Computer and Information Communications Engineering in Catholic University of Daegu(CUD), Gyeongsan, Korea.