Detection of DDoS and IDS Evasion Attacks in a High-Speed Networks Environment

Jin-Tae Oh† , Sang-Kil Park†, 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

BcN(Broadband convergence Networks) is being deployed in order to support a variety of network applications such as E-Commerce, DMB(Digital Multimedia Broadcasting), Home Network, VoIP(Voice over IP), and other services. As network bandwidth is growing rapidly and services are converged, the opportunity and severity of network intrusions are growing as well. This paper presents a novel Intrusion Detection System (IDS) architecture named 'Security Gateway System (SGS)' designed to perform intrusion detection and prevention on high-speed network links. Among several other features in the system, we focus on the detection of DDoS(Distributed Denial of Service) and IDS evasion attacks. We implemented both the mechanisms for handling the bandwidth consuming attack and the detection engine against IDS evasion attack in FPGA(Field Programmable Gate Array). We present some experimental results in a gigabit test bed. The results show that the real-time detection against both attacks is possible with 2 gigabits throughput in each security board.

Key words: *IDS(Intrusion Detection System), DoS(Denial of Service) attack, Bandwidth Control, IDS evasion attack*

1. Introduction

Internet gives us the benefit of remote access to the huge amount of information. With the explosive growth of network applications, the demand for bandwidth is also growing. In order to meet the demand for bandwidth and to follow the necessity of service convergence, BcN is currently being deployed with enhanced capability of QoS(Quality of Service) provisioning and security, and IPv6. In a high-speed networks environment such as BcN, it is more likely for the network resources to be exposed to various intrusion activities. The propagation speed of intrusion is also expected to be much faster than in the existing Internet.

The first issue handled in this paper is a DoS attack that consumes a huge amount of network bandwidth. Since the emergence of the first worm "Morris" in 1988, the Internet resources were frequently exposed to hacking activities such as Nimda, Code Red, and SQL Slammer worm. Therefore, the patterns of the most siginificant and powerful attacks in the Internet are of worm propagation and DoS attacks[1-4]. These attack types make a tremendous traffic on the Internet. For defending against the DDoS attacks, the network engineers have made many attempts to design the systems that help identify the sources of launching DDoS attacks and stop the malicious attacks. The systems are usually deployed at three administrative network domains: victim network, intermediate network and source network. In this paper, we present the Bandwidth Controller in our proposed IDS named 'Security Gateway System (SGS)' against the bandwidth consuming DDoS attack.

This Bandwidth controller is implemented in hardware chipset(FPGA) Virtex II Pro which is produced by Xilinx and acts as a policing function. We referenced the TBF(Token Bucket Filter) in Linux Kernel 2.4 and implemented this function in HDL(Hardware Description Language) Verilog[5,6]. This HDL code is synthesized in hardware chipset and performs as the Gigabit Traffic controller in real time. This policing function can throttle the traffic as the bandwidth controlling bps speed.

In an effort to encounter against various types of malicious traffic and/or intrusions, many Intrusion Detection System(IDS)s have been developed. Most IDS systems are of signature-based. The signature-based IDS devices rely almost entirely on string matching. Thus, breaking the string match of a poorly written signature is trivial. Current NIDS(Network-based IDS)s are barely capable of real-time traffic analysis and detecting IDS evasion techniques on Fast Ethernet links[7,8]. Gigabit Ethernet has become the actual standard for large network installations. Therefore, there is an emerging need for enhanced security analysis techniques that can keep up with the increased network throughput. In this paper, we present the SGS architecture that has a pattern matching approach with Detection Engine against IDS evasion attacks. We implemented our system through the FPGA (Field Programmable Gate Array) logic as detection mechanism that can be applied to Gigabit-Ethernet links. We first briefly introduce the whole architecture of our system designed to perform intrusion detection on high-speed links. And then, we present the efficient Detection

Manuscript received June 5, 2007
Manuscript revised June 20, 2007

Engine against IDS evasion techniques that is run by FPGA logic. Especially, we focus on detection mechanism against IDS evasion with Unicode[9,10].

The remainder of the paper is structured as follows. The next section presents background concepts of DoS attacks and IDS evasion techniques. Section 3 describes the SGS architecture. Section 4 presents our proposed detection mechanisms against DDoS attacks. In section 5, IDS evasion detection mechanisms that we have applied in the proposed system is described. In section 6, experimental results in a gigabit test bed are given. Finally, we present some conclusions.

2. Background

2.1 DDoS attacks

The most prevalent form of attack, denial of service(DoS) attack such as shown in Figure 1 is one of the most difficult attack patterns to encounter with. Even among the hacker community, the initiation of DoS attacks is regarded as trivial requiring so little effort to execute. Even so far, DoS attacks deserve special attention from security administrators because of their ease of implementation and potentially significant damages. DoS attacks are different from most other attack patterns because their purposes are not gaining unauthorized access to networks or illegally retrieving information.

These attacks prevent legitimate users from accessing network resources for normal use or management of network facilities. The DDoS attack disrupts not only the site under attack but also the local network of the compromised host. Depending on the number of infected Web sites in a network, the amount of load generated by these attempts could cause a local network disruption. This disruption varies from a slow network to an unusable network as pipes fill and devices fail from the unexpected load. Services running on any infected system are likely to be slowed, and their legitimate usages are possibly blocked[11].

The threat of DoS attacks can be reduced through the following methods:

-Anti-Spoof features: Proper configuration of these features on router and firewalls can reduce the risk.

-Anti-DoS features: Proper configuration of these features on router and firewalls can help limit the effectiveness of an attack.

-Traffic Rate Limiting: ICMP-based DDoS attacks are commonly limited by this method.

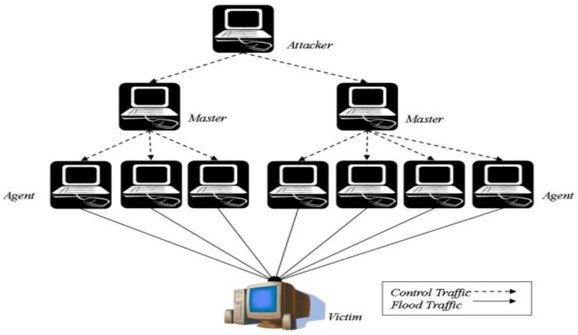

Fig. 1 An example of DDoS attacks

2.2 IDS evasion techniques

Although there are various categories of intrusion detection systems, evasion techniques have also become sophisticated. The basic idea behind evasion is to fool the IDS into seeing data different from what the target host will see, thus allowing the attacker to slip through undetected. Some IDS evasion techniques are described below.

2.2.1 Path obfuscation

An attacker can use a Web browser’s URL to enter a path statement in order to access a file on the Web server with the intention of causing damage, or to retrieve sensitive information. Normally, the path statement would be incorporated into the attack signature, and the attack could be recognized. However, the attacker could alter the URL’s path statement to appear differently to evade detection and cause harm. For an example, “/winnt/. /. /. /test” is the same as “/winnt/test,” but the signatures don’t match, thus an IDS trained to alert on “/winnt/test” will miss this attack.

2.2.2 Unicode Encoding

Hex encoding can be used to represent characters in URLs. This type of encoding has mostly been used in security community. The famous IIS exploit that uses this encoding method is an example of what a Unicode request looks like.

http://127.0.0.1/scripts/..%c0%af../winnt/system32/cmd.exe?+c+dir+c:\

Directory traversal exploits use strings like “.. /.. /.. / ”. Most IDSs have signatures to detect this, but attackers replace the “ / ” with the Unicode equivalent, “%c0%af,” and evade the IDS and thus traverse other directories. In fact, many variations of the same string could be created. Table 1 shows some examples of Unicode encoding.

Table 1: Unicode Exploits

#	URL
1	/msadc/..%255c../..%255c../..%255c../winnt/system32 /cmd.exe?/c+dir+c:\
2	/msadc/..%25%35%63../..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir+c:\
3	/msadc/..%255c..%255c..%255c..%255cwinnt/system 32/cmd.exe?/c+dir+c:\
4	/msadc/..%25%35%63..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
5	/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\

3. System architecture

3.1 overall architecture

Through the cooperation of components in the system, 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 intrusion detection on high-speed links. Therefore, we focus on effective detection strategies applied to FPGA logic and kernel logic. In our approach, each pattern matching component is automatically translated into structural VHDL. The components are then synthesized and mapped on to the FPGA with a vendor CAD tool set. Through the CAD tools, the design functionality is tested and the resource requirement and performance are estimated and obtained. Figure 2 illustrates a block diagram of SGS architecture and depicts overall security board composition. Detailed explanation for each block is missing due to the page limitation in this paper.

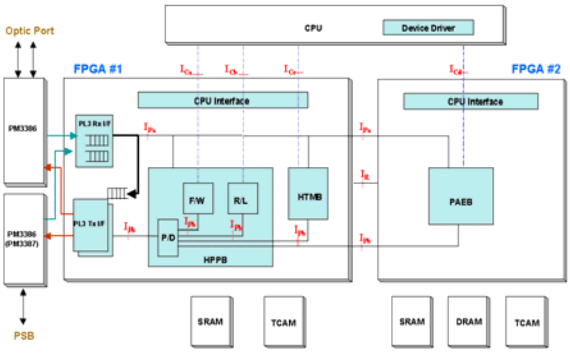

Fig. 2 A Block Diagram of SGS Architecture and Components

Hardware based and high performance SGS contains security features including functions of firewall, IDS, Rate-limiting, and traffic metering which are implemented on two FPGA (Xilinx Vertex II Pro) chips in each security board module[12]. Security board also has embedded CPU MPC860 with Linux OS. Total five security boards are implemented in SGS. Firewall, Rate-limiting, and traffic metering are implemented in FPGA #1, while intrusion detection function is implemented in FPGA #2. Each security board has two gigabit port interfaces.

3.2 security card architecture

Figure 3 shows a hardware based gigabit Security Card(SC) for network anomaly detection and response inputs packet through the gigabit Ethernet interface. SC is composed of mainly two function groups. One is packet Analyzer and another is packet Responder. This input packet is investigated by the Analyzer and Responder. This function is programmed in HDL(Hardware Description Language) code. Every input packet to SC is investigated and transferred in near real time. Packet is decided by the Responder to transmit or drop.

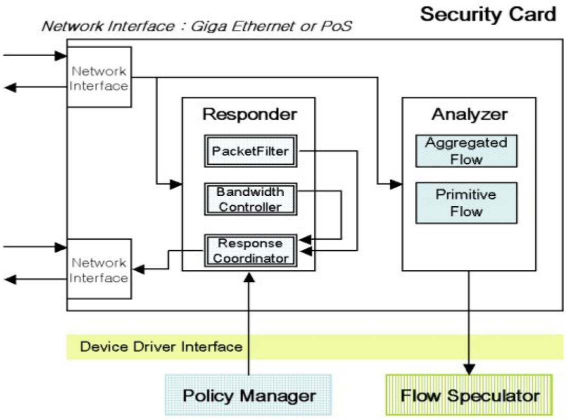

Fig. 3 Organization of Network Attack Response Card

In case of operating the gateway system which adopted the structure as shown in Figure 3, each ingress traffic is analyzed and aggregated flows are created by Analyzer located in Xilinx FPGA chipset. BPS(Bis Per Second) and PPS(Packet Per Second) information is updated by aggregated flow manager in real time. The created information is analyzed by traffic information analyzers and second-step processing information is sent to the response policy manager(RPM). The RPM creates 5-tuple based response policy such as packet-filtering and bandwidth-controlling. These created policy information is translated for the TCAM(Ternary Contents Addressable Memory) . The policy structures are shown in Table 2. Response Policies in TCAM can cover all IP packets including TCP, UDP, and ICMP packets. This response policy is written in TCAM by device driver. After response policy is applied(written) in TCAM, every ingress packets are compared by every entries of TCAM. If TCAM's search result is true, this packet is processed by the packet-filtering block or bandwidth- controlling block.

Table 2: Packet Filtering and Bandwidth-Controlling Response Policy

Field Name	Bit Stream	Description
Reserved	[143:129]	Reserved
Block	[128]	BandwidthCtl(1) /Firewall(0)
Valid	[127]	Rule Valid
Tcam_Port	[126]	Direction[0/1]
ICMP_CODE	[125:118]	ICMP Code
ICMP_TYPE	[117:110]	ICMP Type
TCP_flags	[109:104]	TCP Flag Info
TCP_Dport	[103:88]	TCP/UDP Destination Port
TCP_Sport	[87:72]	TCP/UDP Source Port
Protocol	[71:64]	IP Header
DST IP	[63:32]	Destination IP Address
SRC IP	[31:0]	Source IP Address

3.3 IDS chip architecture

Figure 4 shows the IDS chip architecture that contains a detection engine against IDS evasion attacks.

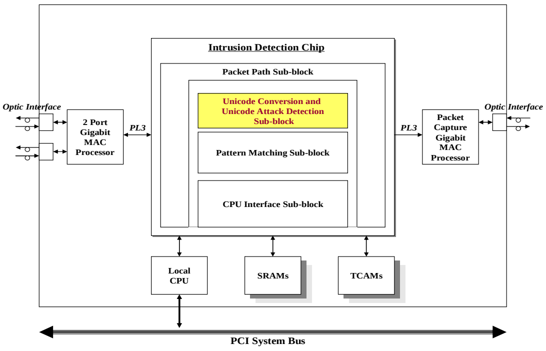
Intrusion Detection Chip
Packet Path Sub-block
Unicode Conversion and Unicode Attack Detection Sub-block
Pattern Matching Sub-block
CPU Interface Sub-block
Optic Interface
2 Port Gigabit MAC Processor
PL3
PL3
Packet Capture Gigabit MAC Processor
Optic Interface
Local CPU
SRAMs
TCAMs
PCI System Bus

Fig. 4 IDS chip architecture

Internal modules for detection operation consists of Unicode conversion and Unicode attack detection function, pattern matching function, and Rule management function.

- Unicode conversion and Unicode attack detection function is to decode and decide whether the packet is for attack or not against Unicode string.

- Rule-based Pattern Matching module matches the incoming packet for the packet header and contents of data. It performs an effective intrusion detection function based on the pre-defined intrusion rule-sets providing the capability of real-time analysis and immediate response.

- CPU Interface module communicates with the rule-related software module operating in embedded CPU and manages the rule-set that is required for intrusion detection.

Through the cooperation of these components, SGS analyzes data packets as they travel across the network for signs of external or internal attacks.

4. Detection mechanisms of DDoS attacks

In this section, we use the network hostile traffic detector to detect and respond against the attack by installing it in the gateway system. We present a reconfigurable policing and its methodology. A leaky bucket or token bucket is commonly used in traffic engineering. In ATM(Asynchronous Transfer Mode), the leaky bucket is commonly used as a packet shaping mechanism. In packet environment, token bucket is commonly used as a packet policing. In a packet network, the packet size is not equivalent. In Linux, Kernel adopt the TBF(Token Bucket Filter) in TC(traffic Controller) for the Diffserv[1,6,13-16].

4.1 Applying Double Token Bucket mechanism

Bandwidth-Controller is adopting a Double token-bucket mechanism. Using the Double token-bucket is like as setting two thresholds. The first threshold represents the first bucket size, and the second one is the sum of double-bucket sizes. We apply the double token-bucket to the stack data structures as shown in Figure 5-①. The sustainable bandwidth rate is created by the Policy Manager. These input bandwidth rates are translated and adopted in the memory structures as illustrated in Figure 5-② and Table 3.

4.2 Token Replenishing Process

In order to adopt the double-token bucket mechanism, we use two processes: TRP(Token Replenishing Processor) and TBMP(Token Bucket Management Processor). The flow diagram of TRP is shown in Figure 6. It is performed every 250us. It reads each 7 data set from memory and calculates current_token value and add the added_token value to the current_token. The current_token value can not overlimit the total_token values which is located in the rule index memory.

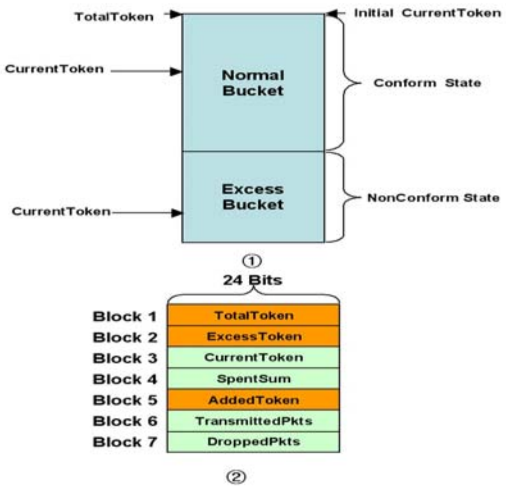

Fig. 5. Bandwidth-Controller using Double Token Bucket and variables

Table 3: Calculation of the Token Bucket Variables

- Normal Bucket = Sustainable bps rate/8
- Excess Bucket = Normal Bucket * 2
- ExcessToken = Normal Bucket
- TotalToken = Normal Bucket + Excess Bucket
- Initial Value of CurrentToken = TotalToken
- AddedToken = Normal Bucket * (1/4000)
- SpentToken = ExcessToken – CurrentToken (valid only Positive)
- SpentSum = SpentSum + SpentToken
- Initial value of SpentToken = 0
- Initial value of SpentSum = 0

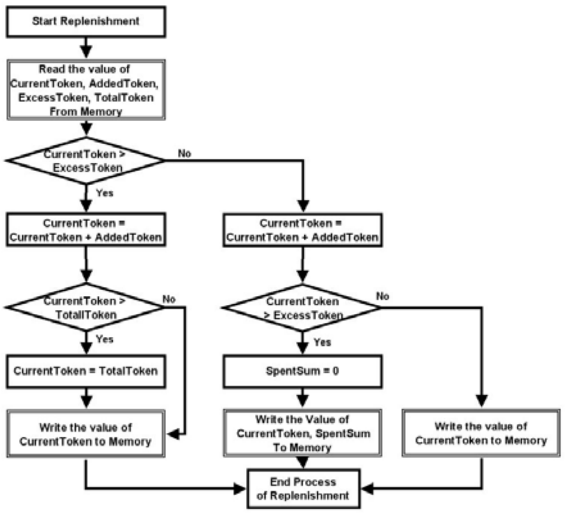
Start Replenishment
Read the value of CurrentToken, AddedToken, ExcessToken, TotalToken From Memory
CurrentToken > ExcessToken
No
Yes
CurrentToken = CurrentToken + AddedToken
CurrentToken = CurrentToken + AddedToken
CurrentToken > TotalToken
No
Yes
CurrentToken > ExcessToken
No
Yes
CurrentToken = TotalToken
SpentSum = 0
Write the value of CurrentToken to Memory
Write the Value of CurrentToken, SpentSum To Memory
Write the value of CurrentToken to Memory
End Process of Replenishment

Fig. 6 Flow Diagram of Token Replenishing Process

Bandwidth-controlling block is actuated by the input 125 ㎒ clock signal. 1clock time is 8 ns. We use two variable counters for checking 250 μs.

- 1us counter = 1clock * 125 = 8ns * 125 = 1000ns
- 250us counter = 1us counter * 250

Bandwidth-controller's main algorithm is applied in the TBMP(Token Bucket Management Processor) as shown in Figure 7. TBMP uses Xilinx Dual-Port Memory with TRP. TBMP can observe current_token value. If it's value belongs to under the excess-token value, the system can issue a notification information to administrator.

TBMP uses the spent_sum register for calculating excess usage from the 2nd bucket. Spent_token values are calculated when the current_token value belongs to under the excess_token variable. If the spent_sum or spent_token values reach over the excess_token variable, packet is dropped. It means that the packet flow is more than the throttling bandwidth speeds. TBMP and TRP processes are using same data area. If one process accesses the data, the data must be altered. For maintaining exact value of the memory, we adopt the memory access control block(memory access Interface). If one process(A) accesses the memory, other process is waiting until the process A is finished. This behaves as a thread in kernel.

5. Detection mechanisms of IDS evasion attacks

5.1 Block diagram of the architecture

Figure 8 shows a block diagram of the architecture to detect IDS evasion attacks. It has parallel data-paths of Unicode Attack Detection Unit in Reconfigurable Hardware. Packet data is passed to the units through a 16-bit bus. The header information of each packet is compared with the predefined header data. If the header information matches the rule, the payload is sent to Unicode pattern match units in which detection algorithms against IDS evasion techniques are included. However, unlike the software implementation, all the rule chains are matched in parallel to achieve a predictable high performance.

5.2 Path Obfuscation Detection Module

To detect Path Obfuscation technique, PODM(Path Obfuscation Detection Module) extracts URI field's value in HTTP request packet. And then makes decision whether URI field's value is valid or not. If URI field's value is out of value range defined in URI syntax in RFC 2396[17], PODM decides that this packet is invalid. Figure 9 represents the PODM algorithm.

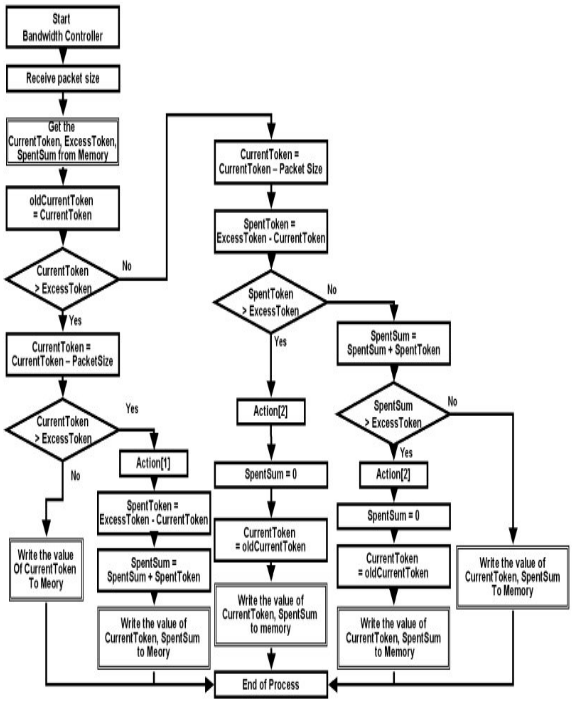
Start
Bandwidth Controller
Receive packet size
Get the CurrentToken, ExcessToken, SpentSum from Memory
oldCurrentToken = CurrentToken
CurrentToken > ExcessToken
No
Yes
CurrentToken = CurrentToken – PacketSize
CurrentToken > ExcessToken
Yes
No
Action[1]
SpentToken = ExcessToken - CurrentToken
SpentSum = SpentSum + SpentToken
Write the value of CurrentToken, SpentSum to Meory
Write the value Of CurrentToken To Meory
CurrentToken = CurrentToken – Packet Size
SpentToken = ExcessToken - CurrentToken
SpentToken > ExcessToken
No
Yes
Action[2]
SpentSum = 0
CurrentToken = oldCurrentToken
Write the value of CurrentToken, SpentSum to memory
SpentSum = SpentSum + SpentToken
SpentSum > ExcessToken
No
Yes
Action[2]
SpentSum = 0
CurrentToken = oldCurrentToken
Write the value of CurrentToken, SpentSum to Memory
Write the value of CurrentToken, SpentSum To Memory
End of Process

Fig. 7 Flow diagram of TBMP

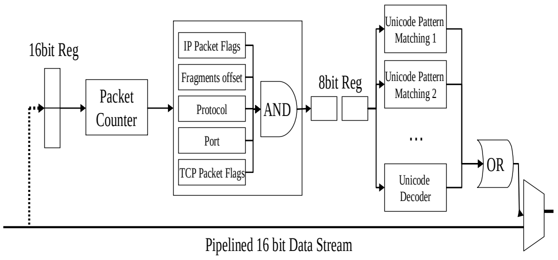

Fig. 8 A block digram of IDS evasion attack detection

```
OPDM (INPUT PKT_String) {
    Extract URI in HTTP Paylaod
        IF (Start_Delimiter){
            URL_Char_Pointer++;
        while (NOT END_Delimiter){
                URL_Char_pointer++;
              Check whether URI Char is
                    vaild;
              IF Invalid, Action Response
                    }
                }
            go to exit;
            }
```

Fig. 9 PODM algorithm

5.3 Unicode Decoder Module

UDM(Unicode Decoder Module) inspects the packet's payload for Escaped characters and converts them back to their ASCII Values. At first, UDM receive input packets and make address for searching SRAM. If %u****(* is hex value) is found in input packet, **** is used as address for SRAM searching key. SRAM consists of ASCII characters. UDM gets ASCII characters from SRAM using matching address. Figure 10 shows the process for Unicode conversion. Converted Packet is sent to Pattern Matching Block for Signature based intrusion Detection.

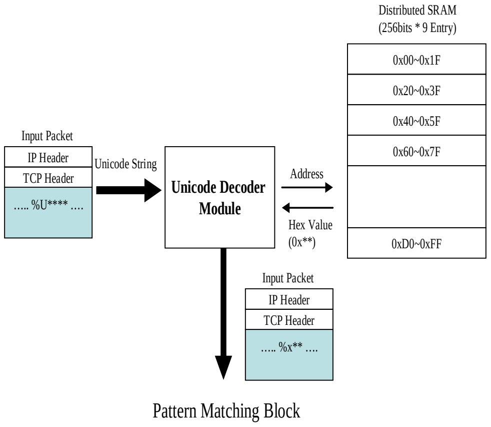

Fig. 10 The process for Unicode Conversion

6. Experimental Results

6.1 DDoS Attacks

The Security Card implemented using double-token bucket mechanism supports two bi-directional gigabit Ethernet interface. Therefore, the input traffic of 1st gigabit Ethernet interface(port a) is transferred to the output port of 2nd gigabit Ethernet interface(port b) and the input traffic of 2nd gigabit Ethernet interface(port a) is transferred to the output port of 1st gigabit Ethernet interface(port b). Figure 11 represents a test bed with IXIA equipment(Traffic Generator/Analyzer), Gigabit L2 Switch, and Security Card. The Security Card has 2 gigabit Ethernet interface and these interfaces are bi-directional. 40Mbps of normal traffic is generated from A Port Tx (1).

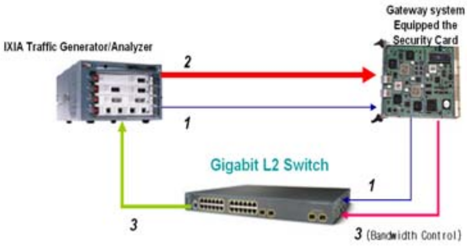
IXIA Traffic Generator/Analyzer
Gateway system Equipped the Security Card
2
1
Gigabit L2 Switch
1
3
3 (Bandwidth Control)

Fig. 11 A Testbed for DDoS attacks

The test scenario is as follows:

- The generation of abnormal traffic is increased from 40Mbps to 1Gbps in 8 steps from B port Tx(2)
- Enforce the rate-limiting rule to abnormal traffic(3): Rule creation and enforcement to the Security Card by Policy Manager and Policing each flow at 40Mbps in A port Rx
- Transmit normal traffic, only policing abnormal traffic

More concrete test environments are described in Table 4 as well as the results of test.

Table 4: Traffic Policing to normal traffic and abnormal traffic

Pkt Size	A Tx	B Rx	A Rx	B Tx
64 Bytes	40 Mbps	40M~1Gbps	20~ 320Mbps	40 Mbps
512Bytes	40 Mbps	40M~1Gbps	20~ 320Mbps	40 Mbps
1024Bytes	40 Mbps	40M~1Gbps	20~ 320Mbps	40 Mbps
1518bytes	40 Mbps	40M~1Gbps	20~ 320Mbps	40 Mbps

The efficiency and accuracy of Bandwidth-controlling block have been proven by the test results as shown in Table 4.

Bandwidth-Controlling block implemented in FPGA of security card can support the full 1Giga bps bi-directional traffic without loss. Bandwidth-controller controls the output traffic at 40M bps rate. Therefore, the sum of 8 flow traffics becomes 320Mbps. The output traffic rate is 320Mbps, and normal 40Mbps traffic is transferred without any loss as illustrated in Figure 12.

6.2 IDS Evasion Attacks

We also have used IXIA Traffic Generator/ Analyzer to generate and transmit packet to SGS in this experiment. Security board of SGS has two gigabit interface fiber ports and two ports are connected to IXIA Traffic Generator. One port is used to receive packets from IXIA and the other port is used to send packets to IXIA again after processing.

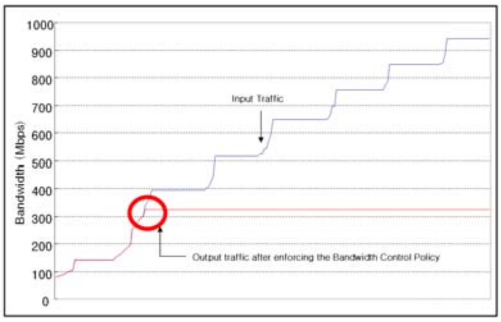

Fig. 12 Controlling the Abnormal Traffic by Bandwidth Control

For generating background traffic, we have used IXIA Traffic Generator. We sent IDS evasion packets with background traffic to SGS. Table 5 represents the experimental result. The results show that all types of IDS evasion attacks are decoded and/or detected up to 2 Gbps of background traffic.

Table 5. The Results of IDS Evasion Test under Load

Background Traffic Attack Type	0G bps	1G bps	1.5G bps	2G bps
URL Encoding	D/ DE	D/ DE	D/ DE	D/ DE
/./ Directory Insertion	D	D	D	D
Premature URL ending	D	D	D	D
Long URL	D	D	D	D
Fake Parameter	D	D	D	D
TAB Separation	D	D	D	D
Case Sensitivity	D	D	D	D
Windows \ delimiter	D	D	D	D
Session Splicing	D	D	D	D

(Note: D:Decoding, DE: Detection)

7. Conclusions

In this paper, we have presented two important problems in intrusion detection systems: DDoS and IDS evasion attacks. The first issue we’ve handled in this paper is the problem of DoS attacks. The most existing blocking solution only responds in way of passive detection technologies. More and more implementing active response is becoming important to protect the legitimate users. In this paper, we implemented a re-configurable bandwidth-controller using double token bucket mechanism. It can serve 2Giga bps Ethernet on bi-directional traffic. We proposed the packet classification mechanism running in real time. To support QoS in real-time packet-classification is very important in BcN.

We then presented a hardware based Detection Engine against IDS evasion with Unicode. One of important requirements of security system is of high performance. Even though many security functions are supported, if not satisfied with network speed, they may not be used. We have developed security boards which provide Intrusion Detection function that support total 2 Gigabit throughput (two gigabit ports). The key idea is to apply Unicode conversion and detection program to Intrusion detection module to detect IDS evasion techniques in wire speed.

References

[1] Paul Ferguson, Geoff Huston, Quality of Service – Delivering QoS on the Internet and in Corporate Networks, Wiley Computer Publishing, 1998.

[2] Pars Mutaf, "Defending against a Denial-of-Service Attack on TCP," In Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID'99), 1999.

[3] I. Garber, "Denial-of-Service Attacks Rip the Internet," IEEE Computer, pp.12-17, April, 2000.

[4] Packet Strom, "DDoS Attack Tools," http://packetstrom.widexs.nl/distributed/indexdate.shtml, 2002.

[5] Samir Palnitkar, Verilog HDL, Prentice Hall , 1996.

[6] Douglas J Smith, HDL Chip Design, Doone Publications 1996.

[7] Thomas Ptacek and Timothy Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Secure Networks Inc., 1998.

[8] Kevin Timm, IDS Evasion Techniques and Tactics, http://www.securityfoucs.com.

[9] Eric Hacker, IDS Evasion with Unicode, http://www.securityfocus.com.

[10] Tom Rodriguez, what are Unicode vulnerabilities on Internet Information Server (IIS) ?, http://www.sans.org.

[11] NIPC(National Infrastructure Protection Center), "find ddos," http://www.nipc.gov/ warnings/advisories /2001/01-005.htm, 2001.

[12] Xilinx company, http://www.xilinx.com.

[13] H. Jonathan Chao, Xiaolei Guo, Quality of Service Control in High-Speed Networks, Wiley-Interscience Publication, 2002.

[14] Werner Almesberger, “Linux Network Traffic Control – Implementation Overview”, http://lcawww. epfl.ch/Publications/Almesberger/TR98_037.ps, EPFL ICA, 1999.

[15] Geoff Huston, Internet Performance Survival Guide – QoS Strategies for Multiservice Networks, Wiley Computer Publishing, 2000.

[16] Netherlabs, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul B Schroeder, Jasper Spaans, “Linun 2.4 Advanced Routing HOWTO”, TLDP.

[17] RFC 2396 , Uniform Resource Identifiers (URI): Generic

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.

Sang-Kil Park received the M.S. and Ph.D. degrees in Computer Science and Information Security Coordination from Jeon-Nam National University in 2000 and 2004, respectively. Since 2000, he stayed in Information Security Division in ETRI(Electronics Telecommunications and Research Institute) to study and develop Network Security related Topics in S/W and FPGA.

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, Gyeongsan, Korea.