

Application Protocol based Anomaly Detection for high speed network

Dong-Ho Kang, Jin-Tae Oh, Ki-Young Kim
Information Security Research Division
Electronics and Telecommunications Research Institute
Daejeon, Korea
Email: dhkang@etri.re.kr

IT R&D Global Leader
ETRI

Abstract. Network intrusion detection systems often rely on matching patterns that are gleaned from known attacks. While this method is reliable and rarely produces false alarms, it has the obvious disadvantage that it cannot detect novel attacks. Accordingly, an alternative approach which can be a combination with pattern matching approach is needed. We have made effort to design and implement protocol anomaly approach to detect known and unknown attacks. This approach extracts a set of service fields from the application payload where many attacks occur and analyzes the value of fields to verify attack. This approach is implemented on the FPGA (Xilinx Virtex II pro) device to process packet at gigabit-per-second data rates.

- **Most IDSs designed for 10/100bps has shown inadequacy in handling such a high incoming data rate for ISPs**
- **Signature based IDS devices rely almost entirely on string matching technique**
- **Consequently, Signature-IDS face the packet leaking problem with the increase in the network speed**

- **SGS has a pattern matching and protocol anomaly approach with detection engine on the FPGA device**
- **Protocol anomaly detection is efficient detection mechanism at higher network speeds. Because the amount of comparison that needs to be performed is much smaller and much more static then pattern matching mechanism**

1. Introduction

With advent of Gigabit and 10Gigabit Ethernet, existing IDS designed for 10/100bps has shown inadequacy in handling such a high incoming data rate for ISPs and premises networks to prevent their network and systems from intrusion[1]. Signature-based IDS devices rely almost entirely on string matching and breaking the string match of a poorly written signature is trivial. Not all IDS devices are signature-based; however, most have a strong dependency on string matching. Consequently, Signature-IDS face the packet leaking problem with the increase in the network speed.

We have made effort to design and implement high-speed Detection Engine against known and unknown attacks that is run as a lower branch of our system named 'Security Gateway System (SGS)'. SGS has a pattern matching and protocol anomaly approach with Detection Engine on the FPGA device as detection mechanism that can be applied to Gigabit-Ethernet links. Protocol anomaly detection is efficient detection mechanism at higher network speeds. Because the amount of comparison that needs to be performed is much smaller and much more static then pattern matching mechanism.

In this paper, we briefly introduce the whole architecture of our system designed to perform intrusion detection on high-speed links. And then, we present the protocol anomaly Detection Engine that is run by of FPGA logic. The remainder of the paper introduces experimental results. Finally, we conclude and suggest directions for further research.

● Three Categories of Intrusion Detection

✦ Pattern Matching

- ▶ The technique of simply looking for patterns
- ▶ Look for a specific pattern somewhere within a packet

✦ Protocol Analysis

- ▶ A bit less specific than pattern matching
- ▶ Look at the packaging of traffic

✦ Anomaly Detection

- ▶ Behavior-based anomaly
- ▶ Protocol anomaly
 - usually falls into the category of RFC compliance checking.
 - If the packet breaks the RFC for a certain protocol in any way, then an alarm is triggered.

- Generally, the core engine of the IDS will rely solely on one specific method, and fractions of the other two methods are implemented in pre- or post-processors to supplement detection capabilities.

2. Related works

2.1 Pattern Matching

Pattern matching is the technique of simply looking for patterns. Generally, this takes place at a much more granular level than protocol analysis or anomaly detection, usually within every individual packet. One example of pattern matching is looking for a string of bytes, which always appears in a specific Trojan such as “Hacked by pl4gu3z” coming from UDP port 6666. The same methodology can be used to look for Denial of Service attacks that rely on sending corrupted packet headers, since we are looking for a specific pattern somewhere within a packet.

2.2 Protocol Analysis

Protocol analysis is a bit less specific than pattern matching and looks at the packaging of traffic, rather than at the payload itself. This is different than straight pattern matching since more advanced calculations are done on each packet. Headers are verified to ensure the packet contains what it says it does, everything adds up as it should, and certain types of encoding aren't used. An example of this is identifying an attack in the OID field of an SNMP packet. The analysis device knows the OID should be a certain number of bytes, but if the next expected field does not appear after that string of bytes, it recognizes something is wrong. Usually this is indicative of an overflow or DoS attack, so an alarm is triggered.

2.3 Anomaly Detection

Anomaly detection is even more indistinct. These ambiguities will be examined more closely below, but this category can really be divided into two sub-categories: behavior-based anomaly detection and protocol-based anomaly detection. Anomaly detection examines traffic at an even higher level than either pattern matching and protocol analysis. Instead of looking at packets for patterns or encoding, this methodology typically focuses on the bigger picture. One of the reasons why this term is so abused is because this methodology can be implemented in a number of different ways. One example is watching the state of connections between hosts. If packets that don't match the established state of the connection start appearing or are severely out of sequence, an alarm is triggered. Another example of anomaly detection is at an even higher level. For example, simply monitoring traffic flows between hosts establishes a baseline that can be considered “normal” activity. In this case, an alarm might be triggered if a Web server suddenly starts accepting connections on port 31337 instead of its normal port of 80. Even something as simple as a web server, which normally only accepts connections, initiating its own sessions out to some host on the Internet can be cause for alarm. This type of detection is normally referred to as flow- or behavior-based anomaly detection, although creative marketing departments have found many of other names for it. Additionally, port scans can also fall into this category. Generally, detecting a port scan requires some type of anomaly/trending algorithm on top of a modified state-tracking engine. A different method—protocol anomaly detection—usually falls into the category of RFC compliance checking. If the packet breaks the RFC for a certain protocol in any way, then an alarm is triggered.

System Architecture



Security Gateway System

- ✦ High speed packet processing
- ✦ Security functions
 - ▶ Firewall
 - ▶ IDS (Intrusion Detection System)
- ✦ Three Xilinx Virtex II Pro FPGA Chip
 - ▶ Anomaly Traffic Inspection Engine
 - ▶ Packet Preprocessing Engine
 - ▶ Intrusion Detection Engine

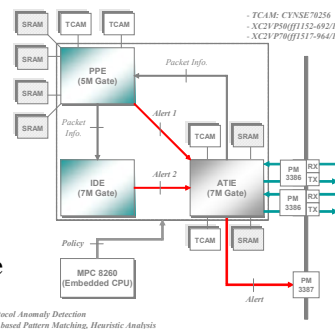


Fig. 1. Security Gateway System Architecture and Components

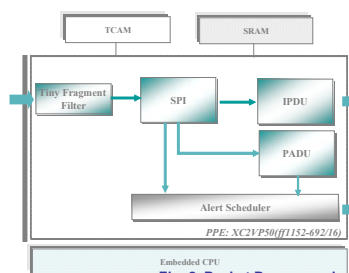


Fig. 2. Packet Preprocessing Engine Architecture

Packet Preprocessing Engine

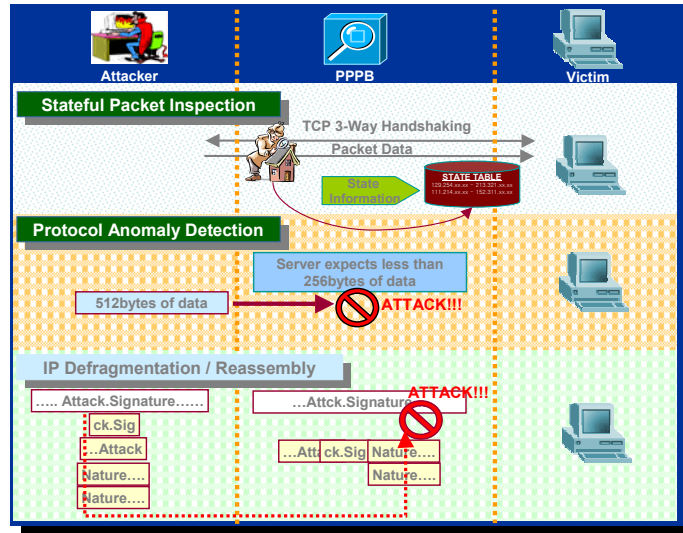
- ✦ IP Defragmentation
- ✦ Stateful Packet Inspection
- ✦ Protocol Anomaly Detection

3. System Architecture

SGS analyzes data packets as they travel across the network for signs of external or internal attack. 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 FPGA logic and kernel logic. SGS is a hardware platform developed to perform high speed packet processing and security functions such as firewall, intrusion detection, and rate-limiting. Total five security boards can be installed. Security board is placed three Xilinx Virtex II Pro FPGA in the data path of a multi-gigabit network also has embedded CPU MPC860 that embedded Linux OS operating in. The device utilizes a 16bit wide data path and when clocked at 125MHz, it is capable of processing data at 2Gbps line rate.

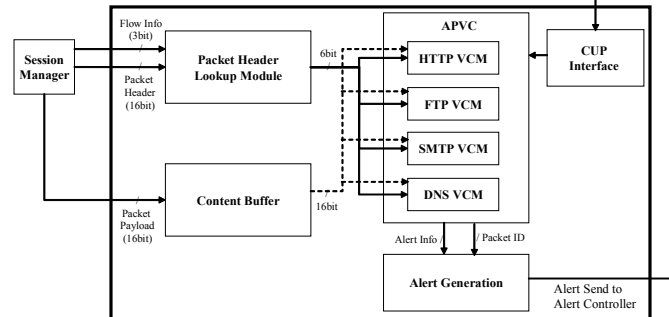
Fig. 1 depicts overall security board composition. Firewall, Rate-limiting, and traffic metering are implemented in ATIC chip. Stateful Inspection, IP Defragmentation, and Protocol Anomaly based detection function in PPE(Packet Preprocessing Engine) chip, and Pattern matching based detection function in IDE (Intrusion Detection Engine) Chip. Each security board has two gigabit port interface. Fig. 2. is Packet Preprocessing Engine chip architecture.

● PPPB (Packet Preprocessing Block)



Internal modules for PPE chip consists of Tiny Fragment Filter, SPI(Stateful Packet Inspection) function, IPD(IP Defragmentation) function, and PAD(Protocol Anomaly Detection) function. Tiny Fragment Filter analyzes whether fragment packet is made small enough to force some of a TCP packet's TCP header fields into the second fragment or not. SPI function watches the state of TCP session and classifies a packet as part of a flow. Packets along with the associated flow state information are passed onto the IPDU and PADU. IPDU function reorders packets that make up an entire session to avoid IDS evasion techniques. PADU function detects attacks that are using protocols outside of their normal usage area which especially includes new attacks that may not yet have been registered by computer security authorities. The major functionality of Packet Preprocessing Engine is to maintain the state of TCP packets and checks protocol validation on high-speed links.

PADE (Protocol Anomaly Detection Engine) Architecture



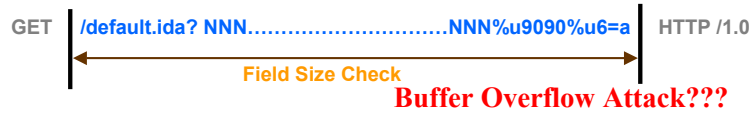
- ✦ **APVC (Application Protocol Validation Check Module)**
 - analyzed the header of application Protocol
- ✦ **VCM (Validation Check Module)**
 - Extracts the value of service fields
 - Calculates the length of value
 - If the length limits the predefined threshold, alert is generated.

4. Protocol Anomaly Detection Engine

Protocol anomaly detection is efficient detection mechanism at higher network speeds. Because the amount of comparison that needs to be performed is much smaller and much more static than signature-mechanism. This mechanism is also capable of detecting new and unknown attacks by distinguishing between a packet streams that breach acceptable application protocol usage rules and a legitimate packet streams. This engine analyzes only received packets after session established. Session manager send flow information consists of client/server direction, Session information.

The Above Figure is a block diagram of our architecture. Protocol Anomaly engine is combined with Session Manger. When a new packet arrived from session manager, it arrives concurrently with flow information. Packet data is passed to the engine through a 16-bit bus. The header information of each packet is compared with the predefined header rule in Packet Header Lookup Module. This Module checks whether there is unusual combinations of TCP flags, IP fragmentation, and unusual TCP options in packet header. If not find any unusual value in the header, the packet's payload in content buffer is sent to APVC (Application Protocol Validation Checker) unit analyzed the header of application Protocol.

● Field Size Check Example



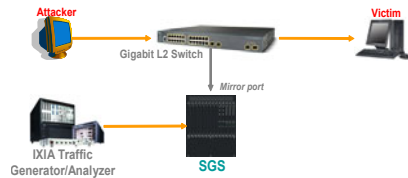
● The list of inspected Service field in application protocol

Application Protocol	Inspection Fields
HTTP Client/Server	Length of URL field Invalid value in URL field Length of Chunk Binary Characters in field Invalid Request Format Accept-Language Field Length
SMTP Client/Server	Length of Command Line Length of Email Address Length of Reply Line Command Syntax Unsafe Command
FTP Client/Server	Command Syntax Length of Command Line Length of Pathname
DNS Client/Server	Length of Lable Length of DNS name Length of UDP/TCP Message Invalid OPCODE

Each VCM (Validation Check Module) extracts the value of service fields can give rise to buffer overflow attack in incoming packet and then calculates the length of value. If the length limits the predefined threshold, alert is generated. The predefined size is rooted in RFCs and appropriate standards can be configured through CPU Interface. The flowing Table shows the list of service field is analyzed by VCM.

Experiment(1)

● Test-bed for Protocol Anomaly Detection



● Attack Tools for protocol anomaly detection

Protocol	Attack Tools & Descriptions
HTTP	Attack tools or exploit code with a very long, malformed HTTP request/response headers (NtoMax, Nikto, IDS informer, Crashiis, Back)
SMTP	Exploit code with a very long, malformed SMTP command line (23 Exploits including Cmail_overflow, zetamail_exp, interscan_mail_bof)
FTP	Exploit code with a very long, malformed FTP command line or illegal FTP protocol (19 exploits including sara-2.0.3, Servu-kill, warftpd-dos, ftpbounce)
DNS	Exploit code with a very long, malformed DNS Query (11 exploits including Zodiac, zlip, bind_tsig)

5. Experimental Result

Security board of SGS has two gigabit interface fiber ports. One port of the board is used to receive normal background traffic from IXIA Traffic Generator and the other is used to receive attack packets through mirroring for inbound traffic on gigabit switch. To generate attack packets, we utilize public attack tools and Exploit codes had been published. The following is the list of attack tools used for test.

Experiment(2)



● Experiments result

Background Traffic	0Gbps	1Gbps	1.5Gbps	2Gbps
Anomaly Detection				
HTTP Anomaly Detection Rate	100%	100%	97%	95%
SMTP Anomaly Detection Rate	100%	100%	100%	100%
FTP Anomaly Detection Rate	100%	100%	100%	95%
DNS Anomaly Detection Rate	100%	100%	100%	100%

IT R&D Global Leader

This system is evaluated by the number of attacks detected. An attack is counted as detected if the system correctly reports the IP address of the attacker and target, time of the attack, and name of the attack. Table shows the experimental result.

Conclusion



- **We described hardware based protocol anomaly detection mechanism**
- **This approach has many advantages**
 - ✦ **Support Multi-gigabit network speed**
 - ✦ **Detect some zero-day attacks**
 - ✦ **Resistance to evasion skills**
- **Future works**
 - ✦ **Needs a combination of pattern matching or others detection method**
 - ✦ **Apply Character distribution algorithm for reducing a false positive alarm**

IT R&D Global Leader

6. Conclusion

In this paper, we described hardware based protocol anomaly detection mechanism. This approach has many advantages. First, it supports multi-gigabit network speed by implementing on FPGA device. Second, it provides the ability to detect some zero-day attacks even before signatures are published because it does not require any prior signature to detect certain classes of attacks. Third, it is resistance to evasion through other similar evasion techniques. Since they do not rely on matching an explicit pattern, variations in the attack generally do not cause a failure to detect as they can in signature-based systems. But it has a limited scope of detection and generated generalized alert as compared with pattern matching method. So it needs a combination of pattern matching or others detection method.

References

1. Bo Song, Ming Ye, Jie Li: Intrusion Detection Technology Research based High-speed Network. IEEE PDCAT'2003 Proceedings
2. Schuehler D.V, Moscola J, Lockwood J: Architecture for a hardware based, TCP/IP content scanning system. IEEE HOTI'03
3. Rachna Vargiya, Philip Chan: Boundary Detection in Tokenizing Network Application Payload for Anomaly Detection. Department of Computer Sciences Technical Report CS-2003-21
4. Matthew V. Mahoney, Philip K. Chan: Learning Models of Network Traffic for Detecting Novel Attacks. Florida Institute of Technology Technical Report CS-2002-08
5. Byoung-Koo Kim, Ik-Kyun Kim, Ki-Young Kim, Jong-Soo Jang: Design and Implementation of High Performance Intrusion Detection System. ICCSA'04
6. Thomas Ptacek, Timothy Newsham: Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Secure Networks Inc (1998)
7. Check Point Software Technologies: Multi-Layer Security: Attack Prevention Safeguards and Attacks Blocked.
<http://cgi.us.checkpoint.com/securitycenter/whitepapers.asp>