

# Real Time Intrusion Detection System for Malformed Packet Attacks

Eun-Yeung Choi<sup>1</sup>, Hyun-Sung Kim<sup>2</sup>, Kee-Young Yoo<sup>1</sup>

<sup>1</sup> Department of Computer Engineering at Kyungpook National University  
Daegu, Korea, 702-701

sionchoi@infosec.knu.ac.kr, yook@knu.ac.kr

<sup>2</sup> Department of Computer Engineering, Kyungil University, Korea, 712-701  
kim@kiu.ac.kr

**Abstract.** The number of bypassing attacks related to malformed packets continues to increase, along with more intelligent and skillful hacking techniques. Most existing intrusion detection systems (IDSs) are unable to detect IP fragmentation attacks, as they could not support a packet reassembly method. Therefore, to cope with these problems, the current paper<sup>3</sup> proposes a network-based IDS that can efficiently detect attacks based on malformed packets. The system is mainly composed of 5 components : Information collecting agent(IA), Simple analyzing agent(SA), Fragment analyzing agent(FA), Collaboration agent(CA) and Decision engine(DE). The IA extracts the important features from network packets. The SA analyzes simple attacks using packet header information, while the FA detects IP fragmentation attacks using an efficient algorithm. The CA collects not only collecting information at the SA and the FA but also collecting other strange information related to the malformed packet. The DE judges whether or not an intrusion has occurred on the basis of information gathered from target systems by CAs.

## 1 Introduction

Cyber terror and abnormal resource use have recently become serious problems over the Internet, resulting in the development of various information protection systems to cope with these problems. Yet, hacking techniques have also become more intelligent and skillful, so that only one malformed packet can stop or even crash a network, and since most attacks are based on large-scale networks, for instance a LAN, WAN, or the Internet, the effects of such attacks are very serious. As shown in table 1, DoS (Denial of Service) attacks generally fall into two categories: stopping a service or resource exhaustion. Stopping a service means crashing or shutting off a specific server that users want to access, whereas, with resource exhaustion attacks, the service itself is still running, but the attacker consumes the computer network resources to prevent legitimate users from reaching the service [?]. Attacks based on malformed packets are particularly serious,

---

<sup>3</sup> Corresponding author : Kee-Young Yoo

as they cannot be properly detected by most IDSs (Intrusion Detection System) mainly due to their various forms and sizes [?]. However, such attacks can be detected by analyzing the characteristics of the packets [?]. Therefore, this paper proposes a network-based IDS that can efficiently detect attacks based on malformed packets. The system is mainly composed of 5 components : Information collecting agent(IA), Simple analyzing agent(SA), Fragment analyzing agent(FA), collaboration agent(CA), Decision engine(DE). The IA extracts the important features from network packets. The SA analyzes simple attacks using packet header information, while the FA detects IP fragmentation attacks using an efficient algorithm. The CA collects not only collecting information at the SA and the FA but also collecting other strange information related to the malformed packet. the DE judges whether or not an intrusion has occurred on the basis of information gathered from target systems by CAs.

**Table 1.** Denial of Service attack categories

	STOPPING SERVICES	EXHAUSTING RESOURCES
LOCALLY	<ul style="list-style-type: none"> <li>- Process killing</li> <li>- System reconfiguring</li> <li>- Process crashing</li> </ul>	<ul style="list-style-type: none"> <li>- Forking processes to fill process table</li> <li>- Filling up whole file system</li> </ul>
REMOTELY	- Malformed packet attacks - Packet floods(SYN flood, smurf, (Land, Teardrop, etc.) distributed denial of service)	

## 2 Malformed packet attacks

This section describes attacks based on malformed packets and hacking techniques that bypass the detection system. A DoS attack can damage and down a system, plus also stop a service and produce resource exhaustion. Although DoS attacks are usually achieved indiscriminately, a refined attack using just one malformed packet can crash a whole system [?].

For example, a ping of death attack, which can be easily achieved using a remote machine, is a DoS attack caused by an attacker deliberately sending an IP packet larger than the 65,536 bytes allowed by the IP protocol. Teardrop is a program that sends IP fragments to a machine connected to the Internet or a network. Fragmentation is necessary when IP datagrams are larger than the maximum transmission unit (MTU) of a network segment. Thus, to successfully reassemble packets at the receiving end, the IP header for each fragment includes an offset to identify the fragment's position in the original unfragmented packet. A land attack consists of a stream of TCP SYN packets that have the same source IP address and TCP port number as the attack host's address and port number. As such, the target receives a packet that appears to be simultaneously leaving and arriving at the same port of the same machine. Some implementations of TCP/IP cannot handle this theoretically impossible condition, thereby

causing the operating system to go into a loop as it tries to resolve the repeated connections to itself.

Most intrusion detection systems (IDSs) are based on a signature matching technique [?][?]. Therefore, some attackers avoid detection by forging packet data as if it is not a signature. Plus, attackers can also avoid detection through fragmentation, as IDSs do not provide a method for reassembling packets [?][?].

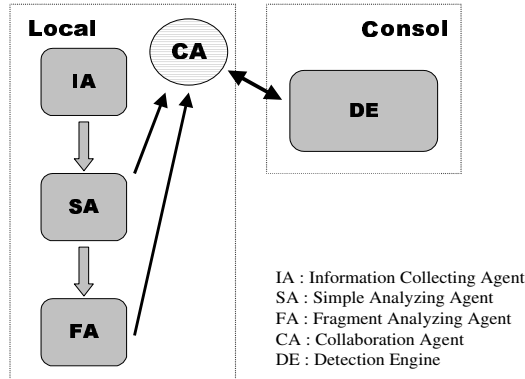
For example, the Tiny fragment attack uses very short IP fragments which are smaller than TCP headers can be used to bypass packet-filtering firewalls. A fragment overlap attack is more elaborate than a tiny fragment attack. In this case, an attacker creates two consecutive fragments. The first fragment has the port number accepted by the filtering equipment, like HTTP(TCP80). The offset is then forged and overlapped by the offset in the second fragment when the fragments are reassembled.

### 3 System configuration

The proposed system has 5 components: Information collecting agent(IA), Simple analyzing agent(SA), Fragment analyzing agent (FA), Collaboration agent(CA) and Decision engine(DE). The overall architecture is shown in figure 1.

#### 3.1 Information collecting agent (IA)

The main task of the IA is to capture all packets from the network. The IA then extracts the important features (parameters) from the network packets for use in the simple analyzer and fragment analyzer. To capture the packets, a BPF driver is used, as provided by Linux, which basically extracts the packet information related to intrusion using functions provided by a pcap-library [?].



**Fig. 1.** System architecture.

The set of features,  $X$ , is in the form of 8-tuples of parameters, which are the main characteristics required to detect attacks using malformed packets. The following shows the set of features:

$$X = (TL, HL, DA, SA, DP, ID, FL, OFS)$$

where each parameter has the following meaning:

- TL : total length of IP datagram in bytes.(header plus data)
- HL : total length of datagram header in four-byte words.
- DA : destination IP address of packet.
- SA : source IP address of packet.
- DP : destination port number.
- ID : identification that shows datagram originated from source host.
- FL : flags used in fragmentation.
- OFS : fragmentation offset that shows relative position of fragment with respect to whole datagram.

The IA stores the features related to the packet fragmentation in a data structure to enable detection of a fragmentation attack at the FA. The features related to IP fragmentation must be inspected correctly to detect an IP fragmentation attack, otherwise bypassing attacks can occur [?]. The IA stores the features related to IP fragmentation, then the FA analyzes them. As such, the ID, HL, TL, OFS, and SA are stored in a data structure. The ID is first hashed and then stored for performance and efficiency reasons, while the rest of the header information is stored at an adjacency linked list.

### 3.2 Simple analyzing agent (SA)

The detection phase has two sub-parts, involving the SA and the FA. The SA checks first whether the packet size is within a valid range, then second if the packet has the same destination and source IP address. As such, an attack is detected by analyzing events with a set of detection rules. To identify intrusive patterns, based on the first 5 features of  $X$ , the sequence of packets must conform to the following conditions:

```

START : IF TL > 65,535 bytes THEN
        IF HL < 20 bytes THEN
            IF DA is the same as SA THEN
                Alert the administrator
            ELSE GOTO START
        ELSE GOTO START ELSE GOTO START

```

### 3.3 Fragment analyzing agent (FA)

Fragmented packets can arrive out of order, as they travel over different paths, yet, if one of the design assumptions is violated, undesired fragments can leak

through the system. Fortunately, however, it is not necessary to remove all the fragments of an offending packet. Since "interesting" packet information is contained in header fields, filters are generally only applied to the first and second fragments [?].

```

IP fragment filtering module( )
{
    IF flag is MF and OFS is zero THEN
        Store hashed ID, TL, SA, DA
    ELSE flag is MF and OFS is non-zero THEN {
        Search for same ID in the structure
        IF second fragment OFS is null THEN
            IF TL/8 > OFS THEN
                Alert and remove from buffer
            ELSE store minimal OFS
        ELSE second fragment OFS is not null THEN
            IF minimal OFS > the present OFS THEN
                IF TL/8 > OFS THEN
                    Alert and remove from buffer
            ELSE store minimal OFS
        }
    }
}

```

**Fig. 2.** IP fragment filtering module.

As shown in figure 2, the FA can also find the second fragment by comparing with the stored minimum offset. Following module describes the procedures of the IP fragment-filtering module. When the initial fragment (with a 0 offset) of an MF flag arrives, the ID is hashed and stored in the structure. If a packet with a non-zeroed offset then comes, a check is made whether the second fragment has the same ID as the ID stored in the list. If the packet is the second fragment, the FA compares the total length of the packet. If the total length is larger than the offset, this means that the fragment is part of a fragment overlap attack. To prevent a buffer overflow, packets need to be deleted at a proper point.

### 3.4 Collaboration agent (CA)

The CA has two functions. First, The CA collects result of analyzing at the SA and the FA then sends to the DE information that gleans at each module. Secondly, The CA collects not only collecting information at the SA and the FA but also collecting other strange information related to the malformed packet.

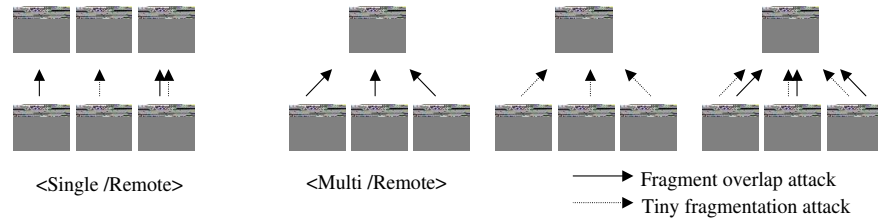
The CA, presents on each target system, migrates autonomously from host to host for exchanging information. Therefore attacks are detected more precise.

### 3.5 Decision engine (DE)

The DE is on the console machine and judges whether or not an intrusion has occurred on the basis of information gathered from target systems by CAs. The DE integrates information and evaluates it. All CAs bring information to the DE independently and, as a result, this information concentrated in the DE.

## 4 Simulation

The proposed system was implemented using a Linux Kernel 2.6 and 1400Mhz Intel pentium-4 PC, plus a pcap-library 0.6.2 was used to collect the packets and Teardrop and Nmap used as the attack tools. Attacks were attempted in various states to evaluate the performance of the proposed system.



**Fig. 3.** Attack scenario.

As shown in figure 3, attacks were attempted in various states to evaluate the performance of the proposed system. For a single-attack host, Teardrop and Nmap were used as a fragment overlap attack and a tiny fragment attack, respectively, plus we have attempted a mixed fragment overlap attack and tiny fragment attack. Three multi-attack hosts have attempted each mentioned attack, and a mixed attack has also tried to investigate success or failure of intrusion detection on multi-attack hosts environments. In the case of a single-attack host, the simulation was performed thirty times, while in the case of multi-attack hosts, ten times of simulations were performed and a detection ratio calculated. The detections results for the single-host attacks were 0% false positive, 0% false negative, 100% detection ratio. Also, the detection results for the multi-host attacks were the same, thereby confirming the effectiveness of the proposed system in detecting fragmentation attacks.

## 5 Conclusion and future work

Malformed packet attacks are difficult to detect, because such packets can bypass an IDS or packet filtering equipment. Accordingly, the current paper proposed a network-based IDS that can effectively detect malformed packets. First, patterns with malformed packet attacks are classified and the features related to the malformed packets extracted. Next, these features are analyzed, while enables malformed packet attacks to be efficiently detected. In particular, an IP fragment-filtering module is proposed that can detect a fragmentation attack, which is normally difficult to detect. The simulation results were 0% false positive and false negative, 100% detection ratio, thereby confirming the accuracy of the proposed IDS in detecting fragmentation attacks. In future work, a case study of malformed packets will be conducted to determine more precise analysis patterns.

## 6 Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

## References

1. Ed Skoudis. *Counter Hack*. Prentice Hall PTR, 2002.
2. Paul E. proctor. *Practical Intrusion Detection Handbook*. Prentice Hall PTR, 2001.
3. Marina Bykova, Shawn Ostermann and Brett Tjaden. Detection Network Intrusions via Statistical Analysis of Network Packet Characteristics. *33rd Southeastern Symposium on System Theory (SSST)*, 309–314, 2001.
4. Behrouz A.Forozan. *TCP/IP Protocol Suite*. Mcgraw-Hill Companies, Inc, 2000.
5. E. Biermann, E. Cloete and L.M Venter. A comparison of Intrusion Detection System. *Computers and Security*, 20:676–683, 2001.
6. Stephen Northcut and Judy Novak. *Network Intrusion Detection An Analyst's Handbook Second Edition*. New Riders, 2001.
7. Hyun-Chul Jung. Attack Techniques using IP Fragmentation. *Korea Computer Emergency Response Team Coordination Center* , 2001.
8. <http://www.cet.nau.edu/mc8/Socket/Tutorials/section4.html>.
9. Sang-Chul Kim. Abnormal IP Packets. *Korea Computer Emergency Response Team Coordination Center*, 2001.
10. Thomas H. Ptacek , Timothy N. Newsham. *Insertion, Evasion, and Denial of Service : Eluding Network Intrusion Detection, Technical Report*. Secure Networks Inc., 1998.
11. Ziemba, Reed and Traina. *Network Security private communication in a public world*. Security Considerations for IP Fragment Filtering, RFC1858, 1995.