Indian Journal of Science and Technology, Vol 8(25), DOI: 10.17485/ijst/2015/v8i25/80830, October 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Improved Pattern Matching Method for Intrusion Detection Systems under DDoS Attack

Jae-Kook Lee[1], Joon Woo[1*] and Jong-Hun An[2]

[1]Korea Institute of Science and Technology Information, South Korea; jklee@kisti.re.kr, wjnadia@kisti.re.kr
[2]Korea Internet and Security Agency (KrCERT/CC), South Korea; aj@kisa.or.kr

Abstract

Increasing popularity of the internet usage is causing many DDoS attacks such as 7.7 DDoS in 2009 and 3.4 DDoS in 2011 in South Korea. A DDoS attack is one of the simplest and most powerful cyber-attacks. DDoS attack is getting a huge problem because the unspecified individuals (called zombie PCs) are used in loading malicious codes while attacking a single site or system. Network security appliances such as IDSes (Intrusion Detection Systems) and DDoS attack prevention systems detect attacks by signature-based pattern matching. These appliances check the full data payload of packets to find abnormal signatures. Then reducing the packet inspection time is one of the most important challenges of improving the performance of these systems because pattern matching methods affect the total execution time. In this paper, we analyze network attack events are collected on two DDoS defense systems which are operated by KISA (KrCERT/CC) in Korea. Then we propose improved pattern matching method using multicore process.

Keywords: DDoS, Intrusion Detection System, Pattern Matching, OpenMP

1. Introduction

People are now able to obtain desired information anywhere and anytime as a result of the pervasiveness of the Internet. Unfortunately, in addition to the ability to obtain useful information, cyber threats are also increasing. Distributed Denial of Service (DDoS), in particular, is a major cyber threat. In South Korea, following the 7.7 DDoS attack in 2009, on March 4, 2011 many homepages of public institutions, financial institutions, and portals were damaged after being targeted by DDoS attacks. In[1] published by PROLEXIC Inc. in 2013, indicated that the average bandwidth of the DDoS attack increased from 5.9 Gbps in the previous quarter to 48.25 Gbps (an increase of over 718%). Further, DDoS attack traffic constitutes approximately 3% of total internet traffic in[2].

DDoS attacks are mainly layer 4 attacks of the TCP and UDP type, which can reduce network bandwidth and disrupt connection support for network equipment and devices. In recent times, however, the attack has changed to a new type that can disrupt server support by targeting web applications. It is difficult to differentiate attacks targeting applications from normal traffic; as a result, these types of attacks can significantly affect the targeted server, even in the case of small amounts of connection and traffic. It is essential to check the payload of packets up to network layer L7 in order to detect these types of attacks that can disrupt server support. Then network security appliances such as IDSes, which are used to check the full data payload of packets, detect and prevent attacks through signature-based pattern matching. Recently more and more companies have been deploying IDS in their network[3]. However, reducing the packet inspection time is one of the most important challenges of improving the performance of these systems because pattern matching methods affect the total execution time under DDoS attacks.

In this paper, we identify the changes of DDoS attack types by analyzing the traffic of Cyber-Shelter system and IX (Internet eXchage Point) DDoS response system, which are built and operated by KISA (KrCERT/CC).

*Author for correspondence

Then, we propose an improved pattern matching method of IDS. It is using multicores.

The remainder of this paper is organized as follows. Section 2 discusses the analyses conducted on the traffic of the Cyber-Shelter system and the IX DDoS response system. Section 3 outlines the improved pattern matching technique using multicore, and the results validated through a simple experiment. Finally, Section 4 concludes this paper.

2. DDoS Attack Traffic Analysis

2.1 DDoS Cyber Shelter

The DDoS Cyber Shelter has been operated since 2010 to minimize the damage caused by DDoS attacks on businesses that are not fully prepared. The DDoS Cyber Shelter is designed to cope with traditional attacks which consume network resources through cooperation with the ISP, and detect and block attacks on the application layer at the system level. That is, the Shelter is equipped with a multi-layer defense system (identification of false requests by zombie PCs, application of differentiated service priorities, contents caching to expand server capacity, and utilization of a web firewall to eliminate web security vulnerabilities) to defend against attacks on the application layer, which can affect server availability most seriously with a small quantity of traffic.

However, the target, scope, and period of the service are limited, so it doesn't negatively affect the service in the private sector, because various information security services and products such as the security control service and the DDoS attack response service have already been commercialized. In addition, service receivers are encouraged to use a private service when the service period expires, in order to improve awareness of the need to deal with DDoS attacks, and to promote the information security

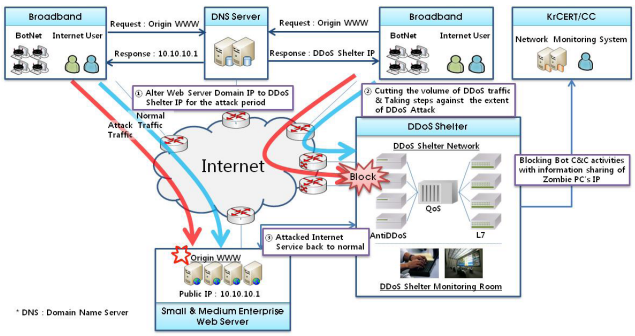

Figure 1. Cyber shelter concept and operation.

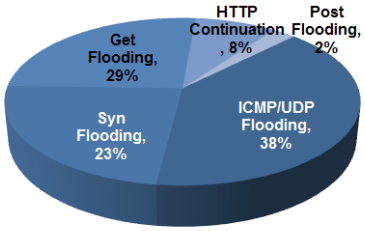

Figure 2. Statistics for each attack type of traffic are collected in cyber shelter system.

market[4]. Figure 1 illustrates the concept and operation of the cyber shelter.

Figure 2 shows the analysis result of DDoS target traffic conducted for each attack type collected by the cyber shelter in 2011. Bandwidth exhaustion attacks such as 'ICMP/UDP Flooding' and 'Syn Flooding' constitute 61% of the total attack types, whereas attack types such as 'Get Flooding' and 'HTTP Continuation', and 'Post Flooding' attacks constitute more than 39%, which can still disrupt server support.

2.2 Interworking Network Segment DDoS Attack Response System

Internet eXchange (IX) service is a physical infrastructure through which Internet Service Providers (ISPs) exchange Internet traffic between their networks (autonomous systems). The primary purpose of an IX service is to allow networks to interconnect directly, via the exchange, rather than through one or more 3rd party networks. The advantages of the direct interconnection are numerous, but the primary reasons are cost, latency, and bandwidth. KrCERT/CC detects DDoS attacks occurring between ISPs by installing DDoS attacks response system (IX-DDoS system) in the interworking network segment between ISPs, as shown in Figure 3, and quickly responds to attacks that are mutually propagated between ISPs. The IX-DDoS system was developed in and has been operated since 2008 with the aim of detecting and responding rapidly to DDoS attacks on the Internet interface route, which connects ISPs. ISPs can be very vulnerable to DDoS attacks as they install and operate DDoS response equipment inside their networks, rather than through the Internet interface route, to prevent external attacks and to provide a stable customer service.

Table 1 shows the TOP 10 attack types detected by the IX-DDoS system from August to December 2011. As can be seen in Table 1, "HTTP Cache-Control Attack," which is a resource exhaustion attack on the server, is consistently ranked in the top 5. As stated above, in recent

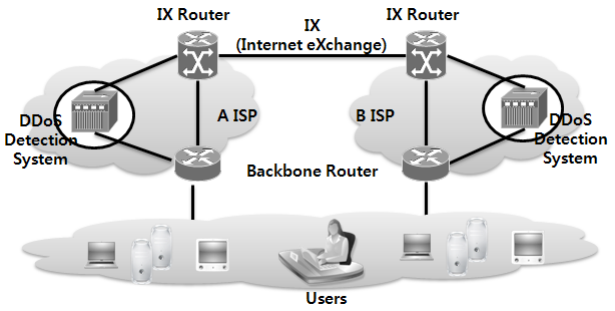

Figure 3. Network architecture of IX-DDoS response system.

Table 1. TOP 5 DDoS attacks

TOP	Aug	Sep
1	TCP Connect DoS	TCP Connect DoS
2	ICMP Tear Drop	**HTTP CC Attack**
3	**HTTP CC Attack**	TCP ACK Flooding
4	UDP Src. IP Flooding	Win32/Slamer Worm
5	TCP ACK Flooding	UDP Tear Drop
TOP	**Oct**	**Nov**
1	Win32/Slamer Worm	Win32/Slamer Worm
2	TCP Connect DOS	**HTTP CC Attack**
3	TCP Invalid port	TCP Connect DOS
4	UDP Tear Drop	TCP ACK Flooding
5	**HTTP CC Attack**	UDP Src. IP Flooding
TOP	**Dec**	
1	TCP Invalid port	
2	**HTTP CC Attack**	
3	TCP Invalid port	
4	Ping Flooding	
5	TCP Connect DoS	

times, many DDoS attacks that result in the exhaustion of server support have been occurring. Such attack types must be verified in the header of a packet as well as up to the payload part.

3. Pattern Matching Performance Improvements

Snort, a representative open source intrusion detection system, consumes 40–70% of the total execution time for signature-based pattern matching that checks the payload part, and uses 60–85% of the total execution commands. Such pattern matching significantly affects the performance of IDS/IPS.

Table 2. Pseudo code (Pattern matching method using multiple sub-patterns)

```
(00) while( text <= textend ) { // compare until end of
     payload
(01)  subPattern = mksub(text, b); // make sub-patterns
(02)  p = subPatternList[subPattern];
(03)  while( p != 0 ) { // if there is matched sub-pattern
(04)    matched = pattSearch(p->m , p->d); // using meta-
        information of sub-pattern
(05)    if( matched ) { // if not matched
(06)      report the matching result; // reports the result
(07)    }
(08)  text += S; // shift for S degree
(09) } }
```

Table 3. OpenMP pseudo code (Pattern matching method using multicores)

```
(00) void oursearch() {
(01)  #pragma omp parallel num_threads(CORES) { //
      allocate process for cores
(02) #pragma omp for schedule(static[dynamic][chunk])
     // setup openmp configuration(static, dynamic and
     chunk size)
(03)       for(i=0; i < packets; i++) {
(04)         match = compare(packet_buf[i], 0, packet_len[i]);
             // multiple pattern matching
(05)         if(match) report the matching result; // if
             matched, the result is reported
(06)  } } }
```

Representative pattern matching techniques include finite automata based Aho-Corasick algorithm and the shift table based Wu-Manber algorithm. When a large number of signatures need to be pattern matched, the Wu-Manber algorithm is at a disadvantage because the distance of the shift, which can pattern match without string compare, converges to one. In[5] we propose the pattern matching technique using multiple sub-patterns has been developed to overcome the aforementioned disadvantage. This technique uses repeated phrase (for phrase or while phrase) for pattern matching, such as illustrated in the following pseudo code.

To improve the performance, we utilize the CPU multicore. An API, OpenMP, which can support parallel programing, was used in order to utilize multicores. The following is the pseudo code implemented using OpenMP.

Figure 4 shows the performance measurement results according to the number of cores for a dynamic schedul-

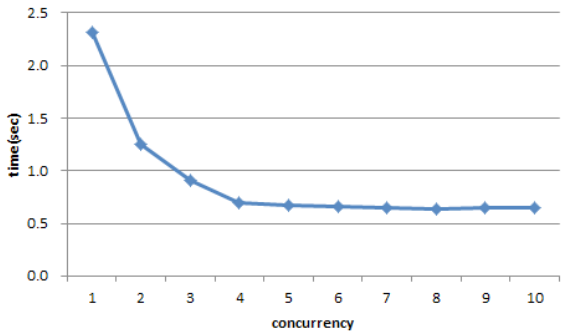

Figure 4. Pattern matching consumption time according to packet scheduling.

ing algorithm and processing 32 packets at once in each core by setting the size of a chunk to 32. Minimization of the search time up to the number of cores of the test system (Intel Core 2 Quad) was validated. The performance is not linearly improved according to the number of cores because work is assigned in the core and the process of collecting the results additionally utilizes the multicore.

4. Conclusions

While the internet is widely used, cyber-attacks like DDoS are continuously changed and increased. Signature-based checking should be performed up to layer L7 of the incoming network traffic in order to more accurately detect attacks. The performance of the pattern matching used to check up to layer L7 determines the total performance of the intrusion detection system.

In this paper, the change of attack type to an attack that disrupts server support was confirmed by analyzing the traffic of the IX-DDoS system and cyber shelter, which are operated by KrCERT/CC. Furthermore, a multicore CPU was utilized to improve the pattern matching performance, which determines the performance of the intrusion detection system. It was confirmed that the time consumed in pattern matching decreased according to the increase in the number of actual cores by parallel processing repeated phrases using the pattern matching technique with multiple cores.

5. References

1. Quarterly global DDoS attack report released by Prolexic. 2013 Apr. Available from: http://www.prolexic.com/knowledge-center-ddos-attack-report-2013-q1/pr.html
2. World network infrastructure security report volume VI released by Arbor Networks. 2011 Feb. Available from: http://www.arbornetworks.com/reportt
3. Sheth C, Thakker R. Performance evaluation and comparison of network firewalls under DDoS attack. International Journal of Computer Network and Information Security. 2013 Oct; 5(12):60–7.
4. KrCERT/CC. Available from: http://eng.krcert.or.kr/service/ddos.jsp
5. Lee J-K, Kim H-S. A hybrid multiple pattern matching scheme to reduce packet inspection time. Journal of the Korea Institute of Information Security and Cryptology. 2011; 21(1):27–37.
6. Jianming Y, Yibo X, Jun L. Memory efficient string matching algorithm for network intrusion management system. Tsinghua Science and Technology. 2007 Oct; 12(5):585–93.
7. Norton M. Optimizing pattern matching for intrusion detection; 2004. Available from: http://docs.idsresearch.org/OptimizingPatternmatchingForIDS.pdf
8. Baker AR, Esler J. Snort IDS and IPS toolkit. Syngress; 2007 Mar.
9. Nishimura T, Fukamachi S, Shinohara R. Speed-up of Aho-Corasick pattern matching machines by rearranging states. Proc Eighth International Symposium on SPIRE 2001; 2001 Nov. p. 175–85.
10. Tuck N, Sherwood T, Calder B, Varghese G. Deterministic memory-efficient string matching algorithms for intrusion detection. IEEE INFOCOM 2004; 2004 Mar. p. 333–40.
11. Wu S, Manber U. A fast algorithm for multi-pattern searching. Technical Report TR 94-17. University of Arizona at Tuscan; 1994 May.
12. Hong YD, Ke X, Yong C. An improved Wu-Manber multiple patterns matching algorithm. IPCCC 2006; 2006 Apr. p. 675–80.
13. Sunday DM. A very fast substring search algorithm. Communication of the ACM. 1990; 33(8):132–42.
14. Wu S, Manber U. AGREP-A fast approximate pattern-matching tool. Proceedings of USENIX Technical Conference; 1992. p. 153–62.