패킷 마이닝을 위한 부하 균형 알고리즘

옥지혜[0] 조동섭
이화여자대학교 과학기술대학원 컴퓨터학과
{okwisdom[0], dscho}@ewha.ac.kr

Load Balancing Algorithm for Packet Mining

Ji-hye Ok[0] Dong-sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

네트워크에서 사용되는 정보들은 수많은 패킷으로 구성되어 송수신 되는데 이러한 패킷의 정보를 이용하여 많은 정보를 알아낼 수 있다. 패킷의 정보를 통계적으로 분석해 외부로부터의 침입과 정보의 유출을 방지할 수 있으며 네트워크의 문제점을 파악하여 시스템을 안전하게 관리 할 수 있다. 그리고 각종 프로토콜을 분석해 네트워크의 부하와 사용자의 행위 패턴과 요구사항을 알아낼 수 있다. 그러나 사용자로부터 실시간으로 들어오는 패킷을 하나의 서버가 처리하고 분석하는 경우에는 많은 부하가 생긴다. 본 연구에서는 실시간으로 생성되는 패킷 데이터를 효율적으로 처리하고 분석하는데 있어서 분산 시스템을 이용하여 해결 하고자 한다. 사용자로부터 생성되는 패킷을 IP그룹별로 분리하여 각각의 프로토콜별로 저장하고 처리하는 부하균형을 이룬 패킷 마이닝 분산 시스템을 제안하고자 한다.

1. 서 론

현대 사회는 컴퓨터와 정보통신의 발전에 따른 고도의 정보통신과 정보처리에 기반을 둔 정보화 사회로, 모든 업무가 전자화(digitalize) 되어가고 있다. 최근에 급속히 확산되고 있는 인터넷은 전세계의 모든 전산망을 하나로 묶어 인프라(infra)를 이루고 긴밀하게 상호동작 한다. 이러한 네트워크에서 사용되는 정보들은 수많은 패킷으로 구성되어 송수신 되는데 이러한 패킷의 정보를 이용하여 많은 정보를 알아낼 수 있다. 패킷의 정보를 통계적으로 분석해 외부로부터의 침입과 정보의 유출을 방지할 수 있다. 그리고 네트워크의 문제점을 파악하여 시스템을 안전하게 관리 할 수 있다. 각종 프로토콜을 분석해 네트워크의 부하와 사용자의 행위 패턴과 요구사항을 알아낼 수 있다. 그러나 사용자로부터 실시간으로 들어오는 패킷을 하나의 서버가 처리하고 분석하는 경우에는 많은 부하가 생겨 분산처리가 필요하다.

본 연구에서는 이를 효율적으로 해결하기 위해 패킷 데이터를 IP별로 분산하여 처리하는 시스템을 제안하고자 한다. 서버로부터 들어오는 패킷을 IP 그룹별로 분리하여 각각의 프로토콜별로 저장하고 처리하는 IP Splitting Rule을 개발하여 패킷의 부하를 해결하였다. 그리고 이전의 시스템과 본 연구에서 제안하는 시스템의 성능을 평가를 하는 알고리즘을 구현하여 제안하는 시스템의 효율성을 검증하였다.

이 논문은 2002년도 두뇌한국21사업에 의하여 지원되었음.

2. 패킷 마이닝을 위한 부하 균형 분산 시스템

본 연구에서는 네트워크에서 데이터의 전송단위인 패킷을 분석하여 서버에 접속하는 사용자의 정보를 분석하고 시스템의 보안을 증진시키고자 한다. 그러나 실시간을 들어오는 패킷은 너무 많은 양이고, 패킷정보를 구분하여 분석하기에는 많은 어려움이 따른다. 따라서 본 연구에서 제안하는 시스템에서는 패킷을 효율적으로 처리하기 위하여 사용자로부터 들어오는 IP를 분석하여 IP 그룹별로 분리하여 패킷을 저장한다. 이때 패킷을 각각의 프로토콜별로 구분하여 저장하여 패킷정보를 쉽게 분석할 수 있게 하였다.

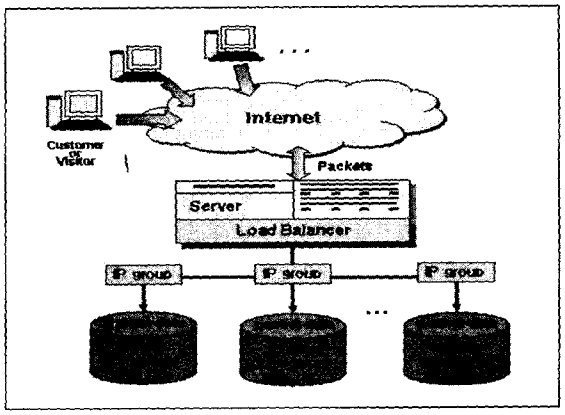

그림 1. 패킷 마이닝을 위한 분산 시스템

3. IP Splitting Rule

제안하는 알고리즘은 사용자로부터 들어오는 IP를 그룹별로 구분하여 프로토콜별로 저장한다. IP Splitting Rule은 그림2와 같고 수행단계는 5단계로 이루어져 있다. 서버에서 접속하는 사용자의 IP 주소를 받아 N비트만큼 Shift 연산을 적용한 뒤, 처리된 결과 값에 M=1인 마스크연산(&)을 S개 적용하면 h의 값은 2^n 개 생성되어진다. 이렇게 생성된 h의 값에 따라 IP 별로 분리되면서 프로토콜별로 패킷이 저장이 된다.

```
u_int32_t nAddress; //IP address
if(pPacket->IsSrvSent()) // If server send

{ // nAddress receive address
    nAddress = pPacket->GetDstAddress();
}
else // if server does not send
{
    // nAddress Transmitter address
    nAddress = pPacket->GetSrcAddress();
}
nAddress=nAddress>>n; //n bit shift
nAddress=nAddress&Mask; //Mask-Operates

if(nAddress==h)
{
   //allocation packet delete
   delete pPacket;
   return;
}
// packet output
g_LogMgr.OutPacketInfo(pPacket);
```

그림 2. IP Splitting Rule

IP Splitting Rule 수행단계

단계 1: 사용자의 IP 입력
단계 2: IP를 n만큼 shift 연산
단계 3: 단계2에서 생성된 값에 S개의 mbit (m=1)& Mask 연산적용
단계 4: h 값 생성
단계 5: h 값에 따라 IP가 그룹화 되어 패킷이 프로토콜별로 저장

3.1 IP Splitting Rule 적용 및 결과

서버에 접속한 사용자의 IP 155.161.121.167이 IP Splitting Rule에 의해 적용되어 처리되는 과정을 그림3에서 나타내었다. 32bit로 구성되어진 사용자의 IP를 30bit 만큼 Shift 연산하여 처리된 결과 값에 M=1인 마스크연산(&)을 적용하면 h의 값은 00, 01, 10, 11 네 가지로 생성되어진다. 이렇게 생성되어진 h의 값에 따라 네 개의 그룹별로 나뉘어 패킷이 저장되어진다. IP 155.161.121.167을 알고리즘에 적용하면 h값은 10이다. 이것은 IP Class B의 값으로 그룹3에 저장된다.

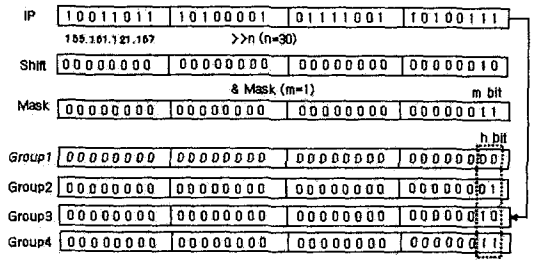

그림 3. IP Splitting Rule 적용

3.2 IP splitting RULE 수행결과

사용자로부터 들어오는 IP를 제안하는 알고리즘에 의해 수행되는 결과를 그림4에서 상세히 나타내었다. IP Splitting Rule에 의해 생성된 h의 값이 h=00이면 그룹1에 패킷정보가 저장되고, h=01이면 그룹2에 패킷 정보가 저장된다. 그리고 h의 값이 h=10이면 그룹3에 패킷정보가 저장되고, h=11이면 그룹4에 패킷 정보가 저장된다. Mask의 수를 늘이면 생성되어지는 h의 수는 2^n 개씩 생기며 h에 따라 IP Class별로 패킷이 분리되면서 저장되게 되어진다. 패킷을 IP 그룹 별로 분리하여 저장을 하면 패킷을 분석 할 때 서버의 부하를 줄일 수 있으며 패킷분석을 효율적으로 할 수 있다.

IP	IP Address	30 bit shift	마스크	h	Group
122.131.177.171	01111010 10000011 10110001 10101011	01	11	01	Group 2
155.161.121.167	10011011 10100001 01111001 10100111	10	11	10	Group 3
203.255.177.177	11001011 11111111 10110001 10110001	11	11	11	Group 4
236.143.132.121	11101100 10001111 10000100 01111001	11	11	11	Group 4
59.198.212.146	00111011 11000010 11010100 10010010	00	11	00	Group 1
203.255.168.171	11001011 11111111 10101000 10101011	11	11	11	Group 4
203.162.155.145	11001011 10100010 10011011 10010001	11	11	11	Group 4
152.187.78.45	10011000 10111011 01001110 00101101	10	11	10	Group 3
121.142.111.89	01111001 01111100 01101111 01011001	01	11	01	Group 2
112.189.68.24	01110010 10111101 01000100 00011000	01	11	01	Group 2
102.67.162.135	11000110 10000011 10100010 10000111	11	11	11	Group 4
75.23.155.124	01001011 00010111 10011011 01111100	01	11	01	Group 2
54.142.34.181	00110110 10001110 00100010 10110101	00	11	00	Group 1
121.146.155.189	01111001 10010010 10011011 10111101	01	11	01	Group 2
132.112.78.151	10000100 01110000 01001110 10010111	10	11	10	Group 3
48.157.178.163	00110000 10011101 10110010 10100011	00	11	00	Group 1
241.145.211.136	11110001 10010001 11010011 10001000	11	11	11	Group 4

그림 4. IP Splitting Rule 수행 결과

3.3 IP Class Group과 IP Splitting Rule의 상관 관계

이러한 IP정보와 제안하는 알고리즘의 필드 값의 상관관계를 그림5에서 나타내었다. 같은 클래스의 패킷의 정보를 저장하려면 IP Splitting Rule에 원하는 IP Class Group의 필드 값을 넣으면 된다. IP Class A 그룹인 패킷정보를 저장하고자 할 때는 n=31 bit shift 하고, m=1(2진수)로 Mask하면 h=0과 h=1인 두 개의 값이 생성된다. h=0에는 IP Class A의 정보가 저장되어지고 h=1은 그 이외의 정보가 저장되어진다. IP Class B 그룹인 패킷정보를 저장하고자 할 때는 n=30

bit shift 연산하고, m=11로 Mask하면 h=00, h=01, h=10 h=11인 네 개의 값이 생성되어진다. 이러한 h의 값에 따라 패킷정보가 4개의 그룹으로 저장된다. 이러한 IP Class의 특성을 IP Splitting Rule에 적용하여 원하고자하는 IP그룹의 패킷을 저장할 수 있다.

class	IP class group A,B,C,D,E
n	IP n bit Shift
m	Mask number
h	Generate h value number

A	31	1	2
B	30	2	4
C	29	3	8
D	28	4	16
E	27	5	32

그림 5. IP Class Group과 IP Splitting Rule의 상관관계

4. 실험 및 성능 평가

본 실험은 패킷데이터가 IP별로 분리되지 않고 하나로 집중되었을 때와 본 연구에서 제안하는 IP Spliting Rule에의해 IP가 그룹별로 분리되었을 때의 트랜잭션의 수행시간 차이를 기술하고자 한다. 실험은 IP 그룹수를 늘렸을 때의 트랜잭션의 CPU 사용시간을 확인하도록 하여 IP를 그룹별 분산처리에 의한 성능 개선부분을 증명하고자 한다. 사용한 트랜잭션은 저장된 프로토콜에 IP로 검색하여 해당되는 레코드를 출력하도록 하였다. 비교 데이터는 프로토콜로 구분된 패킷 데이터 중에서 tcp.log, telnet.log, http.log, udp.log이다.

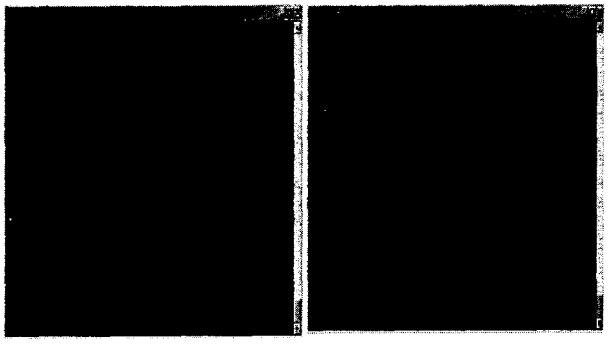

그림 6. 트랜잭션 수행결과

4.1 성능 결과

패킷데이터가 IP별로 분리되지 않고 하나로 집중되었을 경우와 2개로 분리되었을 경우, 4개로 분리되었을 경우, 그리고 8개로 분리되었을 때의 트랜잭션의 수행시간 속도를 비교하여 알고리즘의 성능 테스트를 하였다. 그림 7에서 알 수 있듯이 하나를 사용하였을 때 트랜잭션의 수행속도 보다 IP그룹을 두 개로 두었을 때 평균 1.7배의 속도향상이 되었으며, 4개로 두었을 때에는 3.7배, 8개로 두었을 때에는 7.3배의 성능향상을 나타내고 있다. 실험결과를 통해 패킷을 IP 그룹별로 분산하여 처리하면 수행속도가 빨라지며 그룹의 수가 늘어날수록 더욱더 빠른 트랜잭션 수행결과가 나타남을 알 수 있었다.

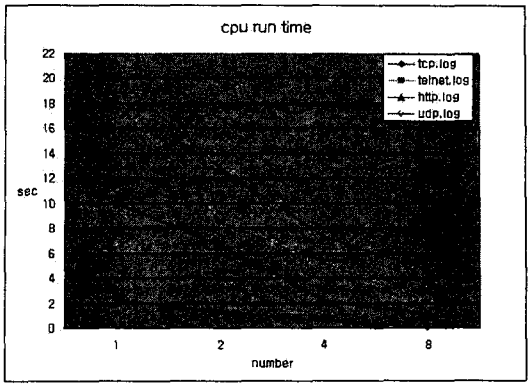

그림 7. 수행결과 분석

5. 결론 및 향후 연구 과제

인터넷에서의 정보전달의 단위인 패킷정보로부터 외부로부터의 침입이나 사용자의 행위 패턴 등 여러 가지 정보 알아내고자 하는 연구가 진행되고 있다. 그러나 하나의 시스템이 이를 처리하기 하기에는 많은 부하 생겨 이를 해결하기 위한 대안이 필요하였다. 본 연구에서 제안한 패킷 마이닝을 위한 분산 시스템은 이를 해결하기 위해 패킷정보를 IP그룹별로 분산하여 프로토콜별로 저장하는 시스템을 설계하고 구현하였다. 또한 이전의 시스템과의 성능 평가를 통해 제안하는 시스템의 효율성을 검증하여 패킷을 효율적으로 처리 할 수 있게 하였다.

향후 패킷 마이닝 알고리즘을 개발 적용하여 시스템 상에서 일어나는 여러 행위의 패턴을 발견하고 이에 따른 문제를 효율적으로 해결하고자 한다. 그리고 편리한 시스템 관리를 위한 관리 인터페이스를 설계할 것이다

참 고 문 헌

[1] 심광현, 외, "분산 가상환경을 위한 네트워크 서버 기술" 정보과학회지 제19권 제5호, 2001년 5월.

[2] 성재모, "DEC를 이용한 분산 컴퓨팅" 정보과학회지 제 14권 제 1호, 1996년 1월.

[3] 이경하, 은유진, 임채호, 정태명, "네트워크 패킷 정보를 기반으로 한 보안 관리", 정보과학회 논문지(A) 제 25권 제 12호, 1998년 12월.

[4] A. Bestavros et al., Distributed Packet Rewriting and its Application to Scalable Web Server Architectures, Proc.6th IEEE Int'l Conf. Network Protocols, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998.

[5] Busch, Costas, "A study on distributed structures," Brown University, 2000.

[6] Ok Ji-hye, "Load Balancing in Distributed System for Packet Mining" ICEE, 2002

[7] http://my.netian.com/~web/net/tcpip/tcpip.htm