Journal of Korea Multimedia Society Vol. 16, No. 6, June 2013(pp. 678-688)
http://dx.doi.org/10.9717/kmms.2013.16.6.678

근거리 통신망에서의 DDoS 봇넷 탐지 시스템 구현

허준호[†], 홍명호[††], 이정민[†††], 서경룡[††††]

요 약

단일 공격과 달리 DDoS 공격들은 네트워크에 분산된 봇넷이 동시에 타겟 서버에 공격을 개시한다. 이 경우 타겟 서버에서는 정상적인 사용자의 편의를 고려해야 하기 때문에 DDoS로 간주되는 패킷에 대하여 접속거부 조치를 취하기 어려운 점이 있다. 이를 고려하여 본 논문에서는 사용자 네트워크단위로 DDoS 공격을 탐지하고 네트워크 관리자가 조치를 취할 수 있도록 하여 전체적으로 봇넷의 규모를 줄여서 타겟 서버의 부담을 줄일 수 있는 DDoS 봇넷 탐지 시스템을 구현 하였다. 본 논문에서 제안한 DDoS 봇넷 탐지 시스템은 공격 트래픽의 시간 단위 흐름을 분석하고 수집한 이상상태에 대한 데이터베이스를 바탕으로 공격을 탐지 하도록 프로그램을 구현하였다. 그리고 패킷들의 평균개수와 표준편차를 이용하여 현재 트래픽의 임계치(Threshold)를 계산하고 이 임계치를 이용하여 DDoS 공격 여부를 판단하였다. 공격의 대상이 되는 서버를 중심으로 이루어졌던 봇넷 탐지 단위를 DDoS 봇에 감지된 공격 모듈이 속한 네트워크 단위 탐지로 전환함으로써 DDoS 공격에 대한 적극적인 방어의 개념을 고려해 볼 수 있었다. 따라서 DDoS와 DoS 공격의 차이점이라 할 수 있는 대규모 트래픽 흐름을 사전에 네트워크 관리자가 차단함으로써 봇넷의 규모를 축소시킬 수 있다. 또한, 라우터 장비 이하의 네트워크 통신에서 트래픽 공격을 사전에 차단할 수 있다면 타겟 서버의 부담뿐만 아니라 WAN 통신에서 라우터의 네트워크 부하를 상당부분 감소시킬 수 있는 효과를 얻을 수 있을 것으로 기대한다.

Implementation Of DDoS Botnet Detection System On Local Area Network

Jun-Ho HUH[†], Myeong-Ho Hong[††], JeongMin Lee[†††], Kyungryong Seo[††††]

ABSTRACT

Different Different from a single attack, in DDoS Attacks, the botnets that are distributed on network initiate attacks against the target server simultaneously. In such cases, it is difficult to take an action while denying the access of packets that are regarded as DDoS since normal user's convenience should also be considered at the target server. Taking these considerations into account, the DDoS botnet detection system that can reduce the strain on the target server by detecting DDoS attacks on each user network basis, and then lets the network administrator to take actions that reduce overall scale of botnets, has been implemented in this study. The DDoS botnet detection system proposed by this study implemented the program which detects attacks based on the database composed of faults and abnormalities collected through analyzation of hourly attack traffics. The presence of attack was then determined using the threshold of current traffic calculated with the standard deviation and the mean number of packets. By converting botnet-based detection method centering around the servers that become the targets of attacks to the network based detection, it was possible to contemplate aggressive defense concept against DDoS attacks. With such measure, the network administrator can cut large scale traffics of which could be referred as the differences between DDoS and DoS attacks, in advance mitigating the scale of botnets. Furthermore, we expect to have an effect that can considerably reduce the strain imposed on the target servers and the network loads of routers in WAN communications if the traffic attacks can be blocked beforehand in the network communications under the router equipment level.

Key words: DDoS, Botnet(봇넷), SYN Flooding, Botnet Detection System(봇넷 탐지 시스템)

※ 교신저자(Corresponding Author) : 서경룡, 주소 : 부산광역시 남구 용소로 45 부경대학교 대연캠퍼스 2호관 분산시스템 연구실(608-737), 전화 : 051) 629-6262, E-mail : krseo@pknu.ac.kr
접수일 : 2013년 1월 29일, 수정일 : 2013년 3월 19일
완료일 : 2013년 3월 27일

[†] 준회원, 부경대학교 컴퓨터공학과 (E-mail : 72networks@pknu.ac.kr)
[††] 준회원 부경대학교 컴퓨터멀티미디어공학 전공 (E-mail : ghd2491@naver.com)
[†††] 준회원 부경대학교 컴퓨터멀티미디어공학 전공 (E-mail : ghd2491@naver.com)
[††††] 정회원 부경대학교 컴퓨터공학과

1. 서 론

최근 시나리오 공격이 확산되면서 정부 차원에서 서버 안정성 향상에 대한 노력이 계속 되어 왔다. 하지만 이런 노력이 주로 국가 기관에 치우친 이유로 소규모 네트워크를 구성하는 단체나 개인은 시나리오 공격에 여전히 취약한 실정이다. 시나리오 공격에는 DDoS (Distributed Denial of Service), DRDoS (Distributed Reflection Denial of Service), Http-Tunnel, MultiHop 등이 있는데[1,2], 단일 공격과 달리 DDoS 공격들은 네트워크에 분산된 봇넷이 동시에 타겟 서버에 공격을 개시한다. 이 경우 타겟 서버에서는 정상적인 사용자의 편의를 고려해야 하기 때문에 DDoS로 간주되는 패킷에 대하여 접속거부 조치를 취하기 어려운 점이 있다. 이를 고려하여 본 논문에서는 사용자 네트워크 단위로 봇넷을 탐지하고 전체적으로 봇넷의 규모를 줄여서 타겟 서버의 부담을 줄이고 네트워크 관리자가 조치를 취할 수 있도록 DDoS 봇넷 탐지 시스템을 구현하였다.

또한, 봇넷은 DDoS 공격, 개인정보 유출, 악성코드 전파, 대규모 스팸메일 발송등 다양한 악의적인 행동을 한다[3-6]. 그중 DDoS 공격은 분산된 여러 대의 봇넷를 이용하여 공격 하고자 하는 대상 서버에 막대한 트래픽을 전송하는 공격이다. 공격 받은 서버는 시스템 과부하로 인하여 정상적인 서비스를 하지 못하거나 다운되는 것이 특징이다.

본 논문에서 제안한 시스템에서는 특정한 컴퓨터에서 DDoS 봇넷 공격을 탐지할 수 있는 기능을 포함하고 있다. 세부적으로 URL을 서버로 계속 요청하는 HTTP flooding, IP 변조 후 SYN 패킷을 대량 전송하는 SYN flooding, 큰 패킷을 대상 호스트로 전송하는 ICMP, UDP flooding을 검출해보고 공격하는 탐지 시나리오를 수립 하였다. 그리고 클라이언트로 부터 도착하는 패킷을 실시간으로 문자열로 비교하여 SQL, HTTP Injection 공격과 서버 내부에 침투할 수 있는 실행 가능한 파일인 웹쉘(Webshell)을 업로드 하는 공격도 검출 대상으로 하였다. 또한, Host 측의 로그를 분석하여 패킷 패턴을 결정하고 탐지하는 방법을 수립하는 기능도 첨가하였다. 한편, Ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP, ICMPv4 패킷을 캡쳐 전송하는 JPcap 라이브러리를 사용하였다.

본 논문의 구성은 2장에서 DDoS 봇넷 탐지 시스템에 대하여 살펴보고, 3장에서 설계 모델에 대하여 설명하고, 4장에서 실험방법 및 성능평가에 대하여 설명하고, 5장에서 결론을 도출 하였다.

2. DDoS 봇넷 탐지 시스템

봇넷(Botnet)은 악의적인 목적을 수행하기 위한 수많은 봇들로 구성된 네트워크로[9] 봇 마스터(Bot master)에 의하여 원격 조정된다. 봇넷은 명령을 하고 네트워크를 관리하는 공격자와 명령을 봇으로 전달하는 역할을 하는 C&C server, 악성코드에 감염된 PC에 상주하면서 실질적인 공격을 담당하는 봇으로 구성된다. 봇넷은 단순히 시스템에 문제를 일으키는 악성코드, 바이러스 등과는 다르게 경제적인 실익을 얻기 위하여 조직적으로 악성 행위를 수행한다. 대표적인 봇넷 탐지 기법으로는 호스트기반 탐지기법으로 가상환경에서 봇 프로그램을 실행하여 트래픽을 분석하는 기법과 네트워크 트래픽 분석을 통한 봇넷 탐지 기법이 있다.

그림 1은 단일 장비나 소프트웨어에 탐지를 의존하던 기존의 네트워크 시스템구조로, 봇넷 탐지 시스템은 일반적으로 통신의 가장 바깥쪽에 위치하여 나가고 들어오는 패킷을 탐지하게 된다. 한편, 방화벽 바깥쪽, 안쪽, DMZ 영역 등의 여러 가지 구축 방법이 존재한다. 그림 1과 같은 탐지 구조는 가장 널리 쓰이

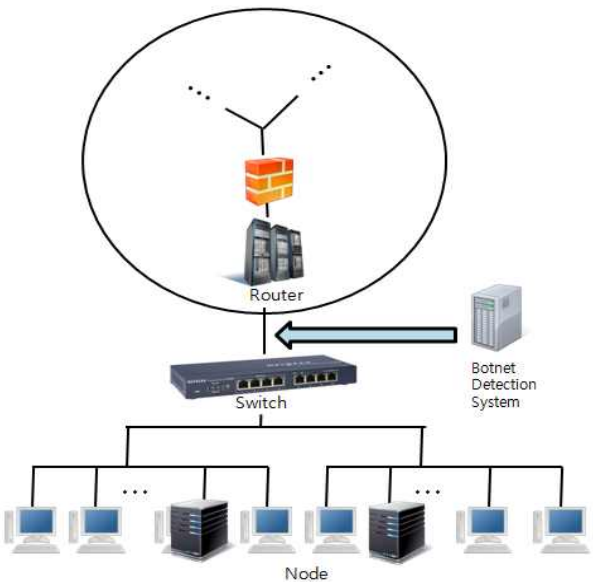

그림 1. 기존의 봇넷 탐지시스템 구조

지만, 일반적으로 탐지 장비의 구입에 소요되는 비용 문제와 단일 장비 또는 모듈에 의해 탐지와 조치가 이루어지기 때문에 부하 문제 또한 고려해 볼 수 있다. 이러한 네트워크 솔루션들은 트래픽을 실시간으로 중간에서 받아서 받아들인 패킷 헤더의 시그니처를 분석하여 SYN, UDP, ICMP등의 공격을 탐지한다.

그림 2는 본 논문에서 제안한 DDoS 봇넷 탐지 시스템으로 클라이언트 측에서 문제가 발생할 경우 클라이언트 측에서 직접 수정하거나 업로드 하지 않고 서버측으로 보고 한다. 그 서버측은 보고 받은 정보를 데이터베이스에 최신화 시키고 주위의 서버에게도 그 정보를 공유한다. 그리고 서버는 각각의 클라이언트에게 최신화시킨 정보를 갱신 시켜주는 형태를 가지고 있다.

또한, 서버는 독립적인 네트워크의 하단에 모두 악성코드에 감염된 봇넷이 존재한다고 할 때, 서버가 내부 네트워크에서의 공격을 감지한다. 트래픽 분석에 앞서 설계한 모델은 크게 적응단계와 조치단계의 두 가지 단계의 모델로 구성된다.

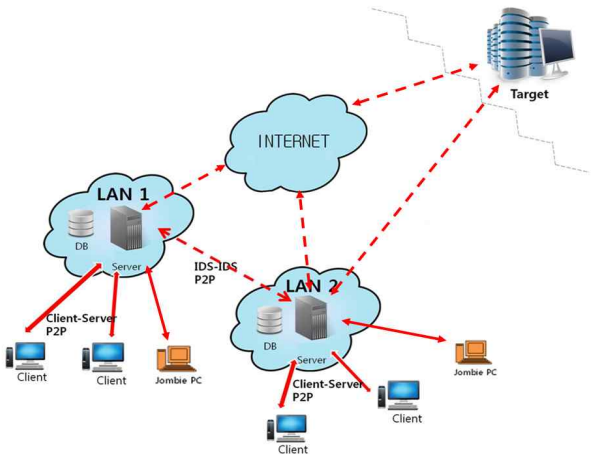

그림 2. DDoS 봇넷 탐지 시스템

3. 설계 모델

기존의 봇넷 탐지 시스템들과의 차이점은 적응단계의 유무이다. 기존의 봇넷 탐지 시스템들의 경우 초기 설계부터 특정한 네트워크 전용 모델로 구성되기 때문에 적응단계가 필요하지 않는 반면, 범용 OS에서 구동하는 애플리케이션 형태의 봇넷 탐지 시스템은 버퍼의 크기, 서비스 포트, 최대 소켓 개수 등이 각각 다른 다양한 환경에서 작동되어야 하기 때문에 적응단계가 필요하다.

적응단계에서는 먼저 설치된 네트워크 시스템에 대한 분석이 필요하다. 분석은 간단한 통계학으로 접근할 수 있는데, 가령 네트워크 시스템에 출입하는 A, B, C 종류의 패킷을 시간 s초 동안, n 번의 회수만큼 수집한다고 하자. 여기서 시간 s초 동안 수집되는 A 패킷의 개수를 C_s 라고 하면 출입되는 모든 A 패킷의 개수는 $T_A = \sum_{i=1}^{n} C_{s_i}$ 가 된다. 따라서 $M_A = T_A/n$은 s초 동안 들어오는 A 패킷 개수의 평균이 된다. 이것으로 부터 패킷 각각의 분산을 구할 수 있는데, 이는 다음과 같다.

$$V_A = \frac{\sum_{i=1}^{n}(C_{s_i} - T_A)^2}{n-1} \quad (1)$$

$$V_B = \frac{\sum_{i=1}^{n}(C_{s_i} - T_B)^2}{n-1} \quad (2)$$

$$V_C = \frac{\sum_{i=1}^{n}(C_{s_i} - T_C)^2}{n-1} \quad (3)$$

표준편차 S_A는 분산 V_A 의 제곱근으로 도출된다. 도출된 A 패킷의 분산 V_A와 표준편차 S_A를 이용하면 시간 s초 동안의 정상 패킷의 평균 개수와 이상 패킷 범위와 확률을 알 수 있다. 하지만 같은 네트워크 시스템일 경우라도 접속 패턴과 네트워크 상태에 따라서 결과가 달라질 수 있다. 따라서 시간 s초 동안 n 번째 검출하는 주기를 다양한 시간대에 걸쳐서 불규칙적으로 구성해야 정확도가 높아진다. 계산된 패킷들의 평균과 표준편차를 이용하여 현재 트래픽의 임계치(Threshold)를 구할 수 있고, 이 임계치를 이용하여 DDoS 공격 여부를 판단하게 된다.

한편, 독립적인 네트워크의 하단에 모두 악성코드에 감염된 봇넷이 존재한다고 할 때, 하위 모듈이 내부 네트워크에서의 공격을 감지한다. 트래픽 분석에 앞서 설계한 모델은 크게 적응단계와 조치단계의 두 가지 단계의 모델로 구성된다.

그림 3은 본 논문에서 구성한 네트워크 모델인데 그 메커니즘을 살펴보면 먼저 각 모듈(Module)들이 소켓을 열고 대기하는 상태에서 외부 모듈이 서버의 역할, 내부 클라이언트의 역할을 수행하게 되어 내부 모듈이 외부 모듈로 접속을 시도한다. 현재의 내부 IP들을 별도의 자료형태로 관리하게 되며 IP의 추가뿐만 아니라 삭제 또한 일정한 프로토콜을 통해 이루어진다.

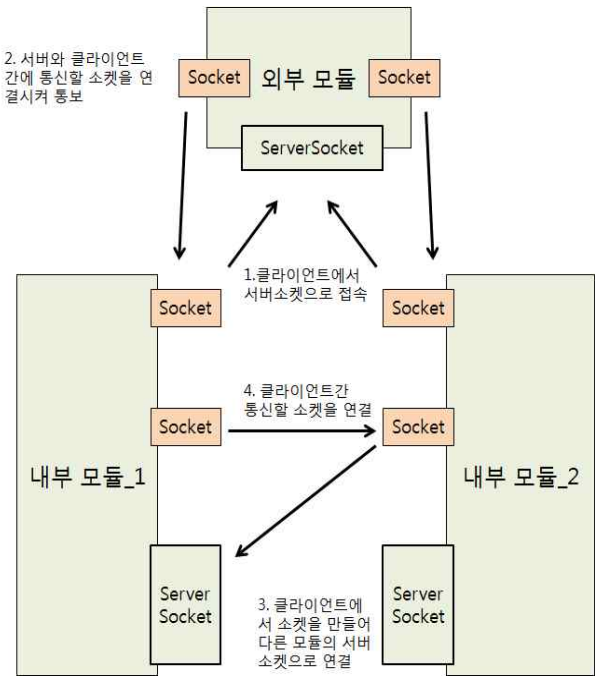

그림 3. 네트워크 모델

한편, 그림 3은 네트워크 모델로 내부 모듈에서는 자체적으로 탐지 규칙을 변경하는 권한은 주어지지 않는다. 이는 악성코드가 탐지 규칙을 임의로 변경하는 것을 막기 위한 것이며, 라우터 내 컴퓨터의 부하를 최소화시키기 위한 목적도 가지고 있다. 그러므로 외부 모듈에서 자신의 소켓에 접속한 클라이언트의 탐지규칙 파일의 해시 값을 비교하고, 만약 다르다면 내부 모듈로 전송을 해서 업데이트가 이루어지는 형태이다.

내부 모듈이 소켓을 닫는 상황, 예를들어 천재지변 및 시스템 종료 등이 발생하면 외부 모듈에서 해당 IP를 제거해 주어야 한다. 이를 위해서 클라이언트는 외부 모듈에게 접속 종료를 알리는 패킷을 전송하고, 이를 통해 IP의 제거가 이루어지게 된다.

그림 4는 본 논문에서 제안하는 봇넷 탐지 시스템의 클래스 다이어그램이다. 봇넷 탐지 시스템은 위와 같이 크게 네 개의 영역으로 나누어진다. 영역 ①의 클래스들은 Jpcap 라이브러리를 이용하여 패킷을 받아오는 역할을 수행한다. 그리고 UserManager와 Network Manager 클래스는 받아온 패킷을 내부 탐지 시스템의 Address 주소별로 분류하는 작업을 수행하게 된다.

한편, 영역 ②는 앞서 기술한 임계치를 계산하고 적용하는 연산을 수행한다. 그리고 연산에 필요한 패킷의 개수는 영역 ①의 CounterPacket으로 부터 받아오게 된다. 그 다음 영역 ③은 체계적인 저장과 필터링을 위해 패킷을 프로토콜별로 분류하는 연산을 수행한다. 패킷의 구조는 대부분 유사하기 때문에 추상 클래스인 JDPacketAnalyzer로 부터 구체화 단계를 거치게 된다.

마지막 영역 ④는 차후에 논할 데이터베이스를 규칙에 따라 저장하고 연산을 수행한다. 데이터베이스

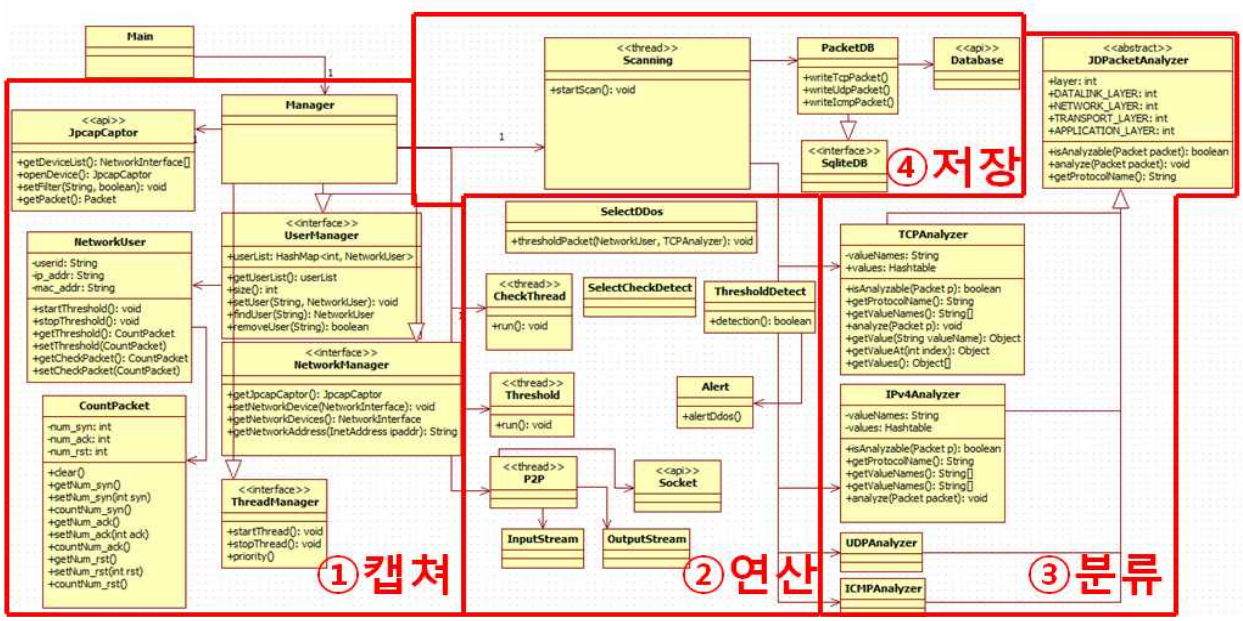

그림 4. 봇넷 탐지 시스템의 클래스 다이어그램

는 규칙에 따라 간략화 된 패킷의 정보를 표현하고, 탐지된 정보에 대해 관리자에게 알리는 기능을 하게 된다.

영역 ②의 부분을 살펴보면 Threshold, Check-Threshold, ThresholdDetect 클래스를 중심으로 동작한다. Threshold 클래스는 각 유저에 대한 패킷 임계값을 정해주는 역할을 하는 클래스이다. 이는 개별적으로 실행되면서 스캔된 패킷들을 이용하여 주어진 수식으로 임계값을 결정하게 된다. 그리고 Check-Thread 클래스는 유저 각각의 패킷에 대해 임계값을 기준으로 하여 공격 여부를 판별을 해 주는 역할을 한다. 또한, ThresholdDetect 클래스는 캡처된 패킷들을 이용하여 공격을 탐지하는 역할을 수행한다. 한편, 스캔 초기에는 임계값이 정해져 있지 않으므로 default Threshold 함수를 이용하여 초기 임계값을 지정해준다.

스캐닝 된 패킷들 개별 단위로 탐지하는 것이 아니라 시간 단위로 패킷들을 한꺼번에 탐지하기 때문에 시간 관련 라이브러리를 이용하였다. 일정한 시간 동안 패킷들을 메모리에 저장하여 지정된 임계값과 비교하여 침입 탐지 유무를 탐지하게 된다.

한편, 그림 5의 UserManager 클래스는 본 논문에서 제안하는 시스템에 연결되는 하위 유저들에 대한 IP를 해시코드로 변환된 것과, 유저 고유 번호를 이용하여 HashMap으로 메모리에 저장하여 DDoS 봇넷 탐지 시스템에 연결되는 하위 유저에 대한 추가, 삭제, 유저정보들에 대해서 프로그램 내에서 언제든지 실시간적으로 표시 해주게 된다. 따라서 어디에서든 하위 유저에 대한 정보가 필요한 경우 find 함수를 이용하여 필요한 유저에 대한 정보를 얻을 수 있으며 많은 하위 유저가 연결 되어도 HashMap을 이용하여 하위 유저를 관리하기 때문에 필요한 유저에 대해서 보다 빠르게 찾아낼 수 있다.

그림 5. UserManager 클래스

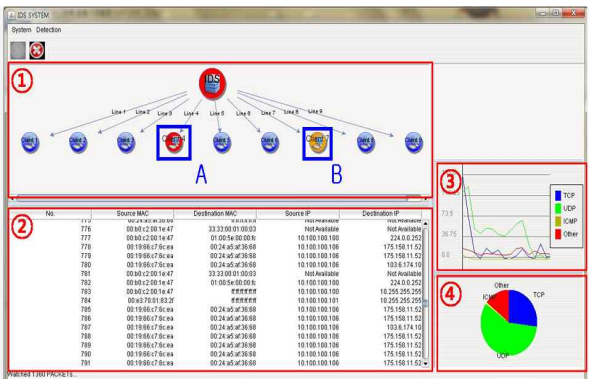

그림 6. 사용자 인터페이스

그림 6은 본 논문에서 제안한 DDoS 봇넷 탐지 시스템 프로그램의 사용자 인터페이스로 총 네 개의 영역으로 나누어진다. 영역 ①은 현재 서버가 관리하고 있는 내부 네트워크의 상태를 나타낸다. 관리자의 입장에서 네트워크의 전반적인 상태를 파악해야 할 필요성이 존재하기 때문에 각 네트워크의 시스템 운영체제와 MAC 주소와 같은 기본적인 내용을 포함한 시스템 전반의 상태가 표시된다. 또한, 네트워크 전체 흐름과 필터링을 통한 특정 노드의 트래픽 상태를 볼 수 있도록 사용자 인터페이스가 설계되었다. 영역 ①는 세 가지의 상태가 존재한다.

4번 노드인 A는 DDoS 및 스캐닝 공격이 감지된 것을 의미하고, 7번 노드인 B는 DDoS 및 스캐닝 공격이 의심되는 상태를, 나머지 노드들은 정상 상태를 나타낸다.

영역 ② 부분 에서는 그림 7과 같이 현재 입출력되는 패킷의 정보를 간략하게 나타내는데 출발지, 목

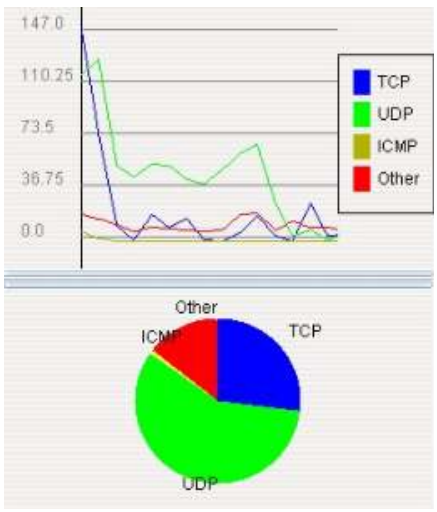

그림 7. 네트워크 상태 그래프

적지의 IP와 MAC 주소가 표시된다. 영역 ③과 영역 ④부분은 각각 종류에 따른 단위 시간당 패킷의 양과 패킷 총량의 비율을 나타낸다.

4. 실험방법 및 성능평가

4.1 실험방법

본 논문에서 제안한 DDoS 봇넷 탐지 시스템은 마이크로소프트사 윈도우7 상에서 기반 프레임워크로 뛰어난 객체지향 언어인 자바(Java)로 구현 하였으며, 알려지지 않은 봇넷에 대한 유출탐지 기능에 대한 성능 비교실험을 위하여 봇넷의 유형별 동작특성을 감안하여 대표적인 봇넷을 분류하고 봇넷 탐지 및 제거 솔루션으로 기존 백신보다 빠른 자동 수집 및 분석 체계를 가지고 있는 프로그램들을 선전하여 비교 하였고, 선행연구[17,18]에서의 실험환경과 성능평가 등을 비교 분석 하였다.

선행연구[17]에서와 같이 여러 상용화 제품들은 분산된 컴퓨터 시스템들이 감염되거나 침해를 당하지 않도록 시그니처를 적용시키는 방법을 채택하고 있기 때문에, 새로운 알려지지 않은 봇넷은 실험의 타당성을 위하여 기존의 소프트웨어[12-16]들로 탐지할 수 없는 실험 환경을 구성 하였다.

그림 8, 9와 같이 선행연구[17,18]와의 차이점을 살펴보면 임계치를 이용하여 클라이언트 측에서 문제가 발생할 경우 클라이언트 측에서 직접 수정하거나 업로드 하지않고 서버측으로 보고 한다는 점이다. 그 서버측은 보고받은 정보를 데이터베이스에 갱신시키고 주위의 서버에게도 그 정보를 공유한다. 그리고 서버는 각각의 클라이언트에게 갱신시킨 정보를 갱신시켜 주는 형태이다. TCP 통신은 네트워크의 트래

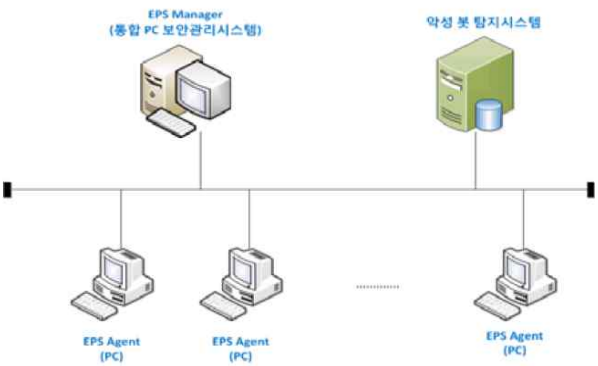

그림 8. 좀비 PC 대응시스템 실험 환경

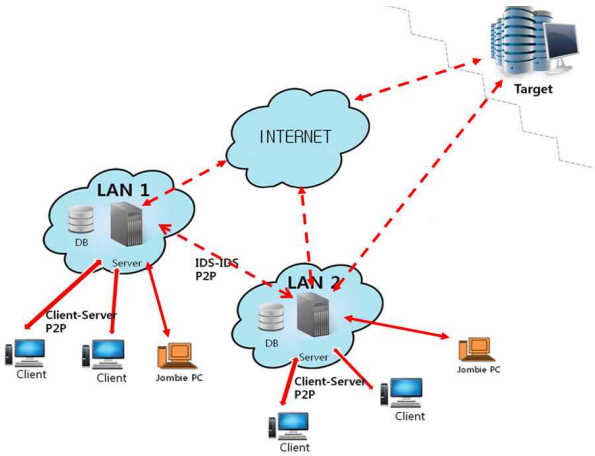

그림 9. DDoS 봇넷 탐지 시스템의 실험 환경

픽과 시간대에 따라 변하기 때문에 단순 계산으로 도출된 임계치로는 DDoS 공격을 검출하기 어렵다. 따라서 계산된 임계치에 정상 상태의 트래픽과 DDoS 공격 때의 SYN 패킷 평균의 중간 값을 더해 최종 임계치를 도출하였고, 이를 DDoS를 판단하는 기준으로 선정하였다.

한편, 그림 10의 공격 탐지 규칙 공유모델을 살펴보면 먼저 탐지한 데이터베이스를 인접한 봇넷 탐지 시스템으로 전송한다. 그러면 전송받은 규칙 데이터베이스와 자신의 규칙을 비교하여 추가 해두면 동일한 공격이 발생할 경우, 패킷 옵션을 보지 않고 패킷을 차단하는 것이 가능해진다.

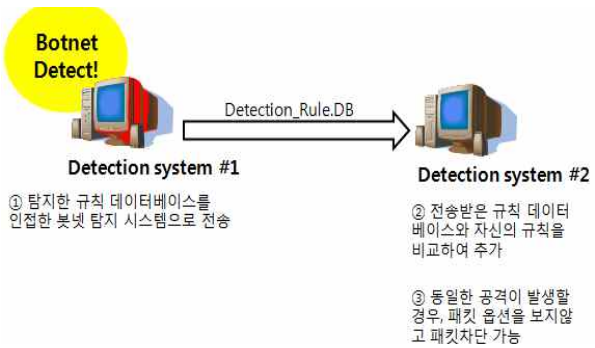

그림 10. 공격 탐지 규칙 공유모델

그림 11과 같은 데이터베이스 형태의 규칙은 네 가지의 항목으로 구성된다. 첫 번째는 RULE 항목으로, 단순하게 침입으로 간주하여 차단하는 BAN 옵션과 임계치에 다다르지 않았지만 정상치보다 많은 개수의 패킷이 지속적으로 출입 할 때의 상황인 Warning 옵션으로 구성된다.

[RULE] [ADDRESS] [WAY OF ATTACK] [TIME STAMP]
Ex) [BAN] [210.110.59.101] [RST flooding] [1346218179]
[Warn] [212.101.5.10] [SYN flooding] [1346213212]
[.] [.] [.] [.]
[.] [.] [.] [.]

그림 11. 데이터베이스의 형태

그리고 두 번째로 봇의 공격대상이 되는 목적지 IP항목이 존재하고, 다음으로 DDoS 봇넷 공격방법과 공격이 이루어지는 시각 항목이 위치한다. 시각은 1970년 1월 1일부터의 누적시를 Milli Second로 나타내는 TIME STAMP로 채택하였다.

4.2 SYN Flooding을 이용한 성능평가

본 논문에서 제안한 임계치를 이용한 실제 네트워크 트래픽의 적용은 아래와 같다. 성능평가 대상인 SYN Flooding공격은 대량의 트래픽을 가하여 서버의 제한된 자원을 점유하여 정상적인 서비스가 불가능하게 만드는 공격이며 가장 널리 알려진 DDoS 공격방법 중의 하나이다. 공격을 받는 서버는 백로그큐(Backlog Queue)에 이를 저장하고 SYN, ACK 패킷을 공격자 IP로 보내게 된다. 공격자로부터 ACK 패킷이 도착하지 않으므로 백로그큐가 꽉 차게 되고, 이후에 들어오는 적법한 SYN 패킷은 처리가 불가능하게 되는 트래픽 공격 방법이다. 이런 SYN Flooding 공격을 막는 방법은 보안 패치로 대기시간을 줄이고 DDoS 봇넷 시스템을 설치하는 것이다. 일정 시간 내에 동시 접속자 수를 점유해야 하고 짧은 시간 안에 똑같은 형태의 패킷을 보내기 때문에 쉽게 인지가 가능하고 그에 해당하는 IP주소 대역을 접속 금지시키거나 확인 후 방화벽 또는 라우터에서 해당 접속을 금지시킴으로써 시스템의 서비스 중지를 막을 수 있다.

그림 12는 정상적인 네트워크 상태의 모습으로 TCP 네트워크 연결 상태는 세션의 시작을 알리는 SYN 패킷과 응답 패킷인 ACK 패킷이 3단계에 걸쳐 나타난다. 이러한 과정을 3-Way Handshaking 이라고 하는데, 비 연결형 방식인 TCP 통신에서는 반드시 필요한 절차라고 할 수 있다.

이에 반해 그림 13은 SYN Flooding 공격이 이루어지고 있을 때의 네트워크 상태로 대부분 SYN 옵션을 가진 TCP 패킷임을 관찰할 수 있다. 공격자는 IP Spoofing을 통해 출발지 IP 주소를 임의로 생성하

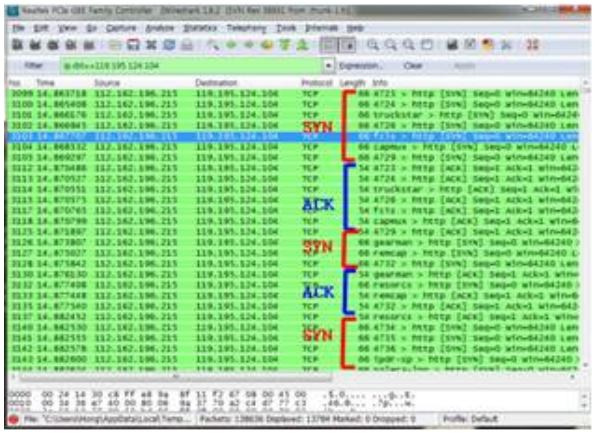

그림 12. 정상적인 네트워크 상태

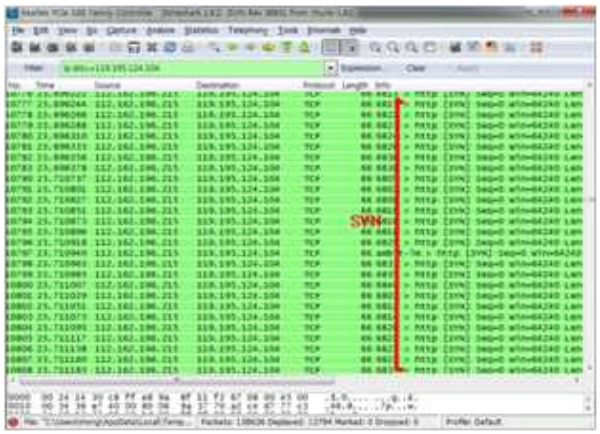

그림 13. SYN Flooding 공격시 네트워크 상태

기 때문에 공격자의 추적은 더욱 어려워지게 된다. 서버는 각각의 SYN 패킷을 서로 다른 사용자로 인식하고 ACK 패킷을 전송한 후에 SYN-ACK 패킷을 지속적으로 기다리기 때문에 트래픽 점유가 일어나는 것임을 알 수 있다.

4.3 트래픽 분석

트래픽 분석은 가상 서버를 이용하여 일반적인 네트워크 시스템을 구성 하고, SYN 패킷의 흐름을 트래픽 분석하였다. 이상 징후의 트래픽 분석을 하기 위해서 일반적인 PC의 인터넷 트래픽을 분석할 필요가 있었다. 따라서 50회에 걸쳐 5초씩 TCP 패킷의 각 옵션의 개수를 분석하였으며, 분석한 결과는 다음과 같다. 그림 14는 SYN 트래픽 분석으로, TCP 연결의 시작을 알리는 SYN 옵션의 경우 인터넷을 사용하지 않는 대기상태에서는 발생 빈도가 현저하게 낮은 것을 알 수 있었다.

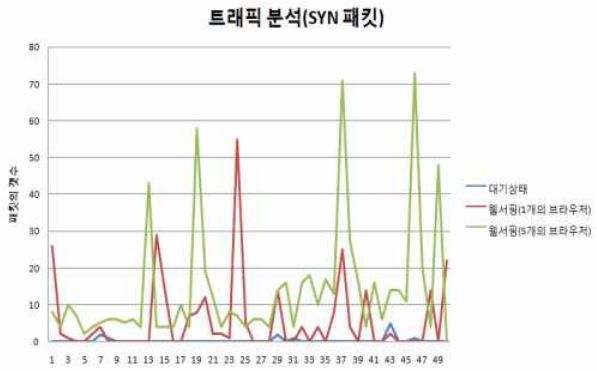

그림 14. SYN 트래픽 분석

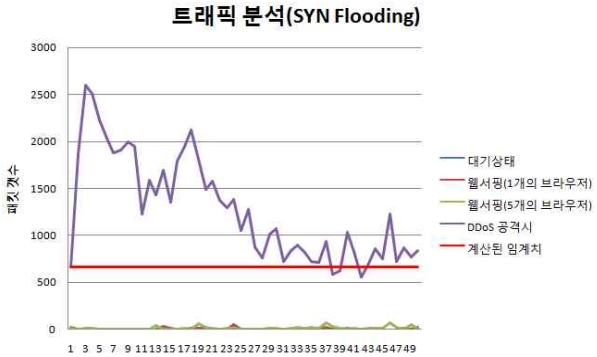

그림 15. DDoS 공격시 트래픽 분석

5개의 브라우저를 실행한 후 인터넷을 사용한 환경에서는 1개의 브라우저를 사용하는 경우보다 패킷의 개수가 빈발한 것을 관찰할 수 있었다. 여기서 표준편차의 표준단위 공식은 z=(x-평균)/표준편차이다. 여기서 표준 정규분포표로 부터 계산된 임계치는 X=13.92+(0.49)*16.34, 즉 22 임을 알 수 있다.

하지만 TCP 통신은 네트워크의 트래픽과 시간대에 따라 변하기 때문에 단순 계산으로 도출된 임계치로는 DDoS 공격을 검출하기 어렵다. 따라서 계산된 임계치에 정상 상태의 트래픽과 DDoS 공격 때의 SYN 패킷 평균의 중간 값을 더해 최종 임계치를 도출하였고, 이를 DDoS를 판단하는 기준으로 선정하였다. 테스트 환경에서 정상상태의 SYN 개수의 평균인 13.92와 DDoS 공격의 SYN 패킷의 평균인 1275.62의 중간 값인 644.77과 계산된 임계치 21.92를 더하여 667이 DDoS를 판단하는 기준이 된다.

그림 15는 DDoS 공격 시의 트래픽 분석을 보여주며, 실선은 계산된 임계치를 나타낸다. 계산된 임계치를 바탕으로 트래픽 분석이 지속적으로 이루어지면 DDoS와 같은 공격을 탐지할 수 있을 것이다. 분석 결과에서 해당 모듈이 악성코드에 감염되어 특정 타겟으로 SYN 트래픽 공격을 시도할 때는 5초 동안 최대 2500개의 SYN 옵션을 가진 패킷이 송신되는 것을 관찰할 수 있었다.

4.4 실험평가 및 결과

기존의 상용화 제품들은 분산된 컴퓨터 시스템들이 감염되거나 침해를 당하지 않도록 시그니처를 적용시키는 방법을 채택하고 있다. 따라서 새로운 봇넷의 출연은 기존의 안티바이러스 소프트웨어로 탐지할 수 없도록 하여 실험의 타당성을 가지도록 하였고[17], 제품 A는 이스트소프트웨어사의 알약[12], 제품 B는 시만텍사의 노턴 안티바이러스[13], 제품 C는 안랩사의 Smart Defense[14], 제품 D는 디지털온넷사 AD-SPIDER 다잡아[15], 제품 E는 엔프로텍트사 온라인백신 V6[16]을 이용하여 비교 분석하였다.

실험결과는 표 1과 같이 기존의 안티 바이러스 제품군들은 알려진 봇넷에 대하여 탐지 후 사용자의

표. 1 선행연구와의 비교분석

실험대상	알려진 봇넷(K/M)		알려지지 않은 봇넷(U/M)	
	결과	감염 후 동작	결과	감염 후 동작
제안시스템	○	사용자에게 경고음으로 알려줌	○	사용자에게 경고음으로 알려줌
비교시스템	○	비의도 접속 탐지	▣	비의도 접속 탐지
제품 A	○	탐지 후 삭제	▣	탐지지속
제품 B	○	탐지 후 삭제	▣	탐지지속
제품 C	○	탐지 후 삭제	▣	탐지지속
제품 D	○	탐지 후 삭제	▣	탐지지속
제품 E	○	탐지 후 삭제	△	탐지 후 삭제
○: 탐지, △:부분탐지, ▣:미탐지				

확인과정 후 삭제를 실행하였으며, 알려지지 않은 봇넷에 대해서는 프로세스 실행 상태로 탐지를 지속하였다. 한편, 제품 E의 경우 알려지지 않은 악성 코드의 일부에 탐지 후 삭제를 실행 하였는데 이는 악성 코드의 유형이 같은 그룹에 제한적으로 적용된 결과를 보여 주었다.

제안한 DDoS 봇넷 탐지 시스템은 일반적으로 트래픽 공격은 단시간이 아닌, 지속적으로 이루어지는 공격이기 때문에 임계치보다 낮은 개수의 패킷이 도달하더라도 한 차례 이상의 이상상태가 존재하게 되면 탐지될 수 있는 강점을 가지고 있다. 실험에서 세 개의 클라이언트가 존재하는 가상의 네트워크 환경을 구성한 후, DDoS 공격을 시도했을 때 시도한 공격이 모두 탐지되는 것을 관찰할 수 있었다.

실험 결과와 같이 기존의 제품군들은 알려진 봇넷에 대하여 탐지 후 사용자의 확인 과정 후 삭제를 실행 하였으며, 알려지지 않은 봇넷에 대해서는 프로세스 실행 상태로 탐지를 지속 하였다. 본 논문에서 제안한 시스템은 설계 목표와 같이 임계값을 벗어나는 트래픽을 분석하여 사용자 비의도 접속 탐지를 수행하였다.

5. 결 론

본 논문에서는 공격 트래픽의 시간 단위 흐름을 분석 하였고, 수집한 이상상태에 대한 데이터베이스를 바탕으로 공격을 탐지하고, 이를 사용자에게 알려주는 프로그램을 구현하였다. DDoS 봇넷 탐지 시스템은 실시간으로 출입하는 패킷을 분석하고, 임계치를 계산하는 방식을 이용하였다. 따라서 소규모 네트워크 내의 컴퓨터가 각각 동작한다면 시스템의 부하가 증가한다는 단점을 가진다. 또한, 비슷한 형태의 공격이 독립적인 방식으로 이루어질 경우 DDoS 공격 탐지에 더욱 많은 시간이 소요될 수도 있다. 이를 보완하기 위하여 탐지된 공격 규칙 IP, 공격 종류, 공격 단계, 공격 시간 등을 단말 시스템 간에 공유를 고려해 볼 수 있다. 이는 개별적으로 공격 규칙을 확보하는 것보다 더욱 방대한 양의 공격 규칙을 확보할 수 있게 되어 하나의 노드가 DDoS 공격을 시도하거나 받게 되더라도 네트워크 전체가 공격을 탐지한 것과 같은 효과를 낼 수 있는 장점을 가지게 하였다. 그리고 일반적으로 트래픽 공격은 단시간이 아닌, 지속적으로 이루어지는 공격이기 때문에 임계치보다 낮은 개수의 패킷이 도달하더라도 한 차례 이상의 이상상태가 존재하게 되면 탐지될 수 있는 강점을 가지고 있다. 테스트에서 세 개의 클라이언트가 존재하는 가상의 네트워크 환경을 구성한 후, DDoS 공격을 시도했을 때 시도한 공격이 모두 탐지 되었다.

비교한 선행연구[17] 유출트래픽 분석 기반의 침입탐지시스템 설계 및 구현 시스템은 설계 목표와 같이 알려져 있는 상태 유무에 관계없이 사용자 비의도 접속 탐지를 수행 하였지만, 본 논문에서 제안한 봇넷 탐지시스템은 알려진 봇넷과 알려지지 않은 봇넷 모두를 탐지하였고, 문제가 발생한 곳을 사용자 인터페이스의 색상과 경고음으로 알려 주었다. ISP 사업자들은 대단위의 Overlay 봇넷을 구축하는 해커를 단독으로 찾기 어렵다. 따라서 외부로 부터의 DDoS 공격의 유입을 신속히 탐지하여 관련 공격 정보를 유관기관에 신속하게 통보하여 ISP 내부 네트워크에 존재하는 좀비화된 봇들을 찾아내도록 사용자에게 경고음으로 알려주는 기능도 추가 하였다.

본 논문에서 제안한 시스템은 공격의 대상이 되는 서버 중심으로 이루어졌던 봇넷 탐지 단위를 DDoS 봇에 감지된 공격 모듈이 속한 네트워크 단위 탐지로 전환함으로써 DDoS 공격에 대한 적극적인 방어의 개념을 고려해 볼 수 있도록 하였다. DDoS와 DoS 공격의 차이점이라 할 수 있는 대규모 트래픽 흐름을 사전에 네트워크 관리자가 차단함으로써 봇넷의 규모를 축소시킬 수 있다는 점이 본 연구의 가장 중요한 목적이라 할 수 있다. 라우터 장비 이하의 네트워크 통신에서 트래픽 공격을 사전에 차단할 수 있다면, 타겟 서버의 부담뿐만 아니라 WAN 통신에서 라우터의 네트워크 부하를 상당부분 감소시킬 수 있는 효과를 기대해볼 수 있다. 따라서 대 규모 봇넷 탐지 장비 구축과 네트워크 봇넷 탐지 모듈 상용화를 병행한다면 트래픽 공격의 방어에 큰 시너지 효과를 얻을 것으로 기대한다.

또한, 공격이 탐지되거나 의심되는 패킷의 데이터베이스를 봇넷 탐지 모듈 간에 교환함으로써 폭넓은 방어가 이루어질 수 있도록 하였다. 트래픽 공격은 IP 또는 패킷의 형태라는 유사성이 존재하기 때문에 봇넷망에서 탐지된 공격 규칙들을 또 다른 봇넷망으로 전송하여 일일이 패킷을 모두 검사하지 않더라도 공격 여부를 판단할 수 있어 같은 방식의 공격에 대

한 탐지 모듈의 부하를 상당부분 감소시킬 수 있고, 네트워크 관리자가 신속한 조치를 취할 수 있게 되어 유용하게 사용될 수 있을 것으로 기대한다.

참 고 문 헌

[1] MIT Lincoln Lab, *1999 DARPA Intrusion Detection Scenario Specific Datasets*, LINCOLN LABORATORY, 1999.

[2] MIT Lincoln Lab, *2000 DARPA Intrusion Detection Scenario Specific Datasets*, LINCOLN LABORATORY, 2000.

[3] H. Debar, M Dacier, and A Wespi, "A Revised Taxonomy for Intrusion-Detection Systems," *Annals of Telecommunications,* 55(7-8), pp. 361-378, 2000.

[4] Haiqin Liu, Yan Sun, and Min Sik Kim, "Fine-Grained DDoS Detection Scheme Based on Bidirectional Count Sketch," *IEEE Computer Communications and Networks (ICCCN)*, pp. 1-6, 2011.

[5] J. Frank, "Artificial Intelligence and Intrusion Detection: Current and Future Directions," *Proc. the 17th National Computer Security Conference*, pp. 1-11, 1994.

[6] HS. Javitz and A. Valdes, "The Sriides Statistical Anomaly Detector," *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pp. 316-326, 1991.

[7] PA. Porras and PG. Neumann, "Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances," *Proc. the National Information Systems Security Conference*, pp. 1-13, 1997.

[8] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," *7th Annual USENIX Security Symposium*, pp. 2435-2463, 1998.

[9] H.R. Zeidanloo and A.A. Manaf, "Botnet Detection by Monitoring Similar Communication Patterns," *International Journal of Computer Science and Information Security*, Vol. 7, No. 3, pp. 36-45, 2010.

[10] J. Markoff, *Russian gang hijacking PCs in vast scheme*, http://www.nytimes.com/2008/08/06/ technology/06hack.html, The New York Times, 2008.

[11] http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/download, "Jcap download"

[12] http://alyac.altools.co.kr/Main/Default.aspx, "Alyac"

[13] http://www.symantec.com,"Semantec"

[14] http://kr.ahnlab.com/b2b/securityinfo/html/renewasecreport,"Anlab news report"

[15] http://www.ad-spider.com/spyware, "Ad-spider"

[16] http://www.nprotect.com/v6/service, "nprotect"

[17] 신동진, 양해술, "유출트래픽 분석기반의 침입탐지시스템 설계 및 구현," 한국콘텐츠학회논문지, 제9권, 제4호, pp. 131-141, 2009.

[18] 김기현, 조용환, 김광훈, "네트워크 탐지 정보를 이용한 좀비 PC 대응시스템," 한국엔터테인먼트산업학회 2011 춘계학술대회 논문집, pp. 186-194, 2011.

[19] 윤성열, 하도윤, 정현철, 박선천, "SIP 환경에서의 DDoS 공격 탐지를 위한 확장된 TRW 알고리즘 검증," 멀티미디어학회논문지, 제13권, 제4호, pp. 594-600, 2010.

허 준 호

2007년 8월 제주대학교 해양과학대학 해양생산과학부 (증식학, 해양학, 해양생물공학) 이학사
2007년 8월 제주대학교 공과대학 통신컴퓨터공학부 컴퓨터공학과 복수전공 공학사
2012년 8월 부경대학교 전산교육학과 석사
2012년 9월~부경대학교 컴퓨터공학과 박사과정
관심분야 : 디지털 포렌식, 분산시스템, 그린 IT

홍 명 호

2013년 2월 부경대학교 공과대학 컴퓨터멀티미디어공학 공학사
관심분야 : 네트워크 포렌식, 정보보호

이 정 민

2007년 3월~부경대학교 공과대학 컴퓨터멀티미디어 공학 전공
관심분야 : 네트워크, 컴퓨터 보안

서 경 룡

1983년 2월 부산대학교 전기기계공학과 공학사
1990년 2월 한국과학기술원 전기 및 전자공학과 석사
1995년 8월 한국과학기술원 전기 및 전자공학과 박사
1991년 10월~현재 부경대학교 컴퓨터공학과 교수
관심분야 : 분산시스템, 컴퓨터네트워크