비정상 연결시도를 탐지한 포트 스캔 탐지 시스템의 설계 및 구현

라용환* · 천은홍**

요 약

네트워크에 연결된 컴퓨터 시스템에 대한 공격을 탐지하기 위하여 침입 탐지 시스템을 설치한다. 기존의 침입 탐지 시스템은 포트 스캔 공격을 탐지하기 위해서 동일 발신지 주소를 갖는 시스템에서 특정시간에 일정 임계값을 초과하는 연결 설정 요청 패킷이 발생했는지를 검사하여 공격을 탐지하므로 실제 공격이 아닌데 공격이라고 탐지하는 False Positive가 높고, 특정시간의 임계값 보다 더 긴 주기로 공격을 시도하는 Slow 스캔 공격과 발신지 주소를 위조하여 공격하는 Coordinated 스캔 공격을 탐지하기 힘들다.

본 논문에서는 TCP의 비정상 연결 시도에서 응답하는 RST/ACK 플래그 패킷을 탐지하여 포트 스캔을 판단하는 탐지 규칙과 데이터 저장 구조를 제안하고, 이를 기반으로 포트 스캔 탐지 시스템을 설계하고 구현하였다. 제안된 시스템은 RST/ACK 플래그 패킷을 탐지하여 공격을 판단하므로 False Positive를 감소시키고, 적은 양의 데이터를 저장하여 긴 시간동안 데이터 유지할 수 있어 Slow 스캔을 탐지할 수 있고, 공격 대상에서 응답한 RST/ACK 패킷을 검사하여 Coordinated 스캔공격을 효율적으로 탐지할 수 있다.

Implementation and Design of Port Scan Detecting System Detecting Abnormal Connection Attempts

Yong Hwan Ra* · Eun Hong Cheon**

ABSTRACT

Most of computer systems to be connected to network have been exposed to some network attacks and became to targets of system attack. System managers have established the IDS to prevent the system attacks over network. The previous IDS have decided intrusions detecting the requested connection packets more than critical values in order to detect attacks. This techniques have False Positive possibilities and have difficulties to detect the slow scan increasing the time between sending scan probes and the coordinated scan originating from multiple hosts.

We propose the port scan detection rules detecting the RST/ACK flag packets to request some abnormal connections and design the data structures capturing some of packets. This proposed system is decreased a False Positive possibility and can detect the slow scan, because a few data can be maintained for long times. This system can also detect the coordinated scan effectively detecting the RST/ACK flag packets to be occurred the target system.

Key words : Intrusion Detection, Port Scan, Slow Scan

* (주)STGSecurity 연구원

** 우석대학교 컴퓨터교육과 교수

1. 서 론

현대 정보화 사회에서는 인터넷과 네트워크에 연결된 컴퓨터 시스템이 급속하게 증가되고 있으며, 인터넷을 통하여 수집된 정보는 우리에게 유익한 정보를 제공하고 있는 반면 많은 부작용을 낳고 있다. 특히 컴퓨터 시스템의 오용과 불법 접근이 개인과 조직에 의해 이루어지고 있는데 이러한 불법 접근으로 인하여 정보의 누출과 시스템의 파괴가 발생하고 있다[1, 2]. 이와 같은 컴퓨터 시스템에 대한 불법 접근이 증가함에 따라 방화벽이나 VPN 등의 보안 시스템을 구축하여 이를 방지하고 있는데 이러한 보안 기술은 피할 수 없는 취약점을 가지고 있고, 완전한 보안 시스템을 구현하기에 충분하지 않아서 이를 보완하기 위하여 컴퓨터에 대한 공격을 식별하고 시스템을 감시할 수 있는 보안 기술에 대한 연구가 이루어져 왔다[1].

초기 컴퓨터 침입자들은 공격을 시도하기 위하여 네트워크와 컴퓨터에 대한 깊은 이해가 필요하였으나, 오늘날은 거의 누구나가 공격 도구를 손쉽게 구할 수 있고 광범위하게 이용 가능하기 때문에 공격을 위한 컴퓨터 시스템내의 취약점을 쉽게 수집할 수 있다. 네트워크 공격자는 공격 개시에 앞서 잠재적 공격 대상에 관한 다량의 정보를 수집하는데, 네트워크에 연결된 컴퓨터의 정보를 수집하기 위하여 포트 스캔 등의 기술을 이용한다[2].

공격 시스템에 대한 취약점을 파악하기 위한 공격 도구로 포트 스캔 등의 방법을 사용하는데, 공격자는 포트 스캐너를 이용하여 공격 대상 시스템에서 어떤 표준 포트들과 서비스들이 실행되고 있는지, 대상 시스템에 어떤 운영체제가 설치되어 있고 어떤 어플리케이션들이 존재하는지 등의 시스템 공격에 필요한 정보를 파악한다. 포트 스캐너는 특정 주소에 대하여 어떤 포트들이 메시지에 반응하는지, 어느 취약점들이 존재하는지 등을 알려주는 프로그램[2]으로 대표적인 포트 스캐너는 Nmap, Netcat, Nessus, CyberCop, Scanner, Secure Scanner, Internet Scanner 등이 있다. 공격자는 이런 도구를 이용하여 신원 확인이나 인증없이 익명으로 스캔을 시도하여 보안 관리자의 주의를 끌지 않고 공격에 필요한 정보를 수집하여 공격을 시도한다[1, 2].

NIST(National Institute of Standards and Technology)는 "침입 탐지는 컴퓨터 시스템 또는 네트워크에서 발생하는 이벤트를 모니터링하는 절차이며 기밀성, 무결성과 유효성을 위협하는 시도, 그리고 컴퓨터 또는 네트워크의 보안 메커니즘을 우회하는 침입의 서명을 분석하는 것"으로 정의하였다[3]. 침입 탐지 시스템의 주요 목적은 컴퓨터 시스템의 데이터를 수집하고, 보안에 관련된 이벤트를 찾기 위해 수집된 데이터를 분석하고, 관리자에게 그 결과를 표현하는 것이다[4]. 또한 침입 탐지 시스템을 개발하는 것은 적절한 탐지 알고리즘을 찾는 것뿐만 아니라 어떤 데이터를 수집할 것이며, 무슨 공격을 탐지할 것인지를 결정하는 것이다[1, 4].

본 논문에서는 기존의 침입 탐지 시스템의 포트 스캔 탐지 알고리즘을 개선하여, 기존 침입 탐지 시스템에서 탐지하기 힘든 포트 스캔 공격을 효과적으로 탐지 할 수 있는 네트워크 기반의 침입 탐지 시스템을 설계하였다.

기존의 침입 탐지 시스템은 포트 스캔 공격을 탐지하기 위해서 동일 발신지 주소를 갖는 시스템에서 특정시간에 일정 임계값을 초과하는 연결 설정 요청 패킷이 발생했는지를 검사하여 공격을 탐지 한다[5, 6]. 이러한 시스템은 빠르게 포트 스캔 공격을 탐지하기에는 매우 효율적이지만, 다음과 같은 단점을 가진다. 첫째, 실제 공격이 아닌데 공격이라고 탐지하는 False Positive가 높다. 둘째, 특정시간의 임계값 보다 더 긴 주기로 공격을 시도하는 Slow 스캔 공격의 경우, 일정시간에 캡쳐해야 하는 다량의 데이터로 인하여 탐지하기가 힘들다. 셋째, 발신지 주소를 위조하여 공격하는 Coordinated 스캔 공격의 경우에는 발신지 주소가 동

일한 주소가 아니기 때문에 탐지하기가 매우 힘들다[5].

따라서, 본 논문에서는 네트워크를 통한 포트 스캔 탐지를 위하여 TCP의 비정상 연결 시도에서 발생하는 RST/ACK 플래그를 탐지하여 이를 기반으로 시스템을 설계하여 False Positive를 낮추어 좀 더 정확하게 탐지함을 보인다. 또한 적은 양의 데이터를 캡쳐하여 공격을 탐지함으로 좀더 효율적인 시스템을 설계하며, 기존 시스템에서 탐지하기 힘들었던 Slow 스캔과 Coordinated 스캔을 탐지하는 기법을 제안하고 구현하였다.

2. 관련 연구

컴퓨터 시스템에 대한 불법 접근을 능동적으로 방지하기 위하여 침입 탐지 시스템에 대한 중요성이 커짐에 따라 많은 연구가 이루어지고 있고 다양한 형태의 침입 탐지 시스템이 개발되고 있다[1, 7]. 본 장에서는 네트워크 기반 침입 탐지 시스템의 포트 스캔 탐지 기법의 연구 동향에 대해 살펴본다.

네트워크에 연결된 시스템들의 증가로 네트워크 자신이 목표가 되는 공격들의 숫자가 증가하고 있는데, 예를 들면, 스푸핑, TCP hijacking, 포트 스캐닝, 핑과 같은 공격들은 호스트 기반의 침입 탐지 시스템으로 공격을 탐지할 수 없다[8]. 이런 이유들로 네트워크 패킷 스니퍼와 같은 도구가 개발되고 네트워크 기반의 침입 탐지 시스템이 등장하였다. 네트워크 패킷 스니퍼는 네트워크에서 발생되는 이벤트들에 대한 정보를 수집하기 위해 네트워크 기반의 침입 탐지 시스템에서 공통적으로 사용되는데, 발신지 패킷과 수신지 패킷의 세부적인 내용에 대한 정보를 사용자에게 제공한다[8, 9].

네트워크 기반의 침입 탐지 시스템은 기존의 컴퓨터 시스템과 운영체제에 영향을 받지 않고 설치될 수 있고, 공격의 목표가 되는 호스트에 설치되지 않기 때문에 감사 데이터 정보가 공격자에 의해 수정될 수 없다. 또한 네트워크의 전략적 포인트에 설치될 수 있기 때문에 설치된 침입 탐지 시스템의 위치를 지나는 모든 네트워크 트래픽을 감시할 수 있어 감시 영역이 상대적으로 매우 넓다[8, 9]. 반면, 침입이 탐지 되었을 때, 공격자들을 식별하는 것이 간단하지 않다. 왜냐하면 수집된 패킷의 정보는 공격자의 신원과 직접적인 관련이 없기 때문이다. 또한 만약 패킷이 암호화되었다면 중요한 정보는 숨겨지기 때문에 패킷을 분석하는 것은 불가능하다[8].

네트워크 기반의 침입 탐지 시스템은 연구 조직과 학교의 많은 프로젝트에 사용되는 방법이며, 대부분의 상업적인 제품들은 네트워크 기반의 형태로 개발되어 왔다[1, 4]. NSM(Network Security Monitor)은 최초의 네트워크 기반의 침입 탐지 시스템으로 포트 스캔을 탐지하기 위해 만들어진 침입 탐지 시스템이다[10].

포트 스캔은 TCP/IP 프로토콜을 이용하여 질의를 보내면 이에 응답을 보내는 특성을 기반으로 동작한다[8]. 포트 스캔은 공격자가 네트워크에 연결된 공격 대상 시스템의 포트에 대한 정보를 파악하기 위한 접속 시도로써, 기본적으로 포트 스캔의 탐지는 시스템의 다른 포트에 짧은 시간에 다수의 연결 설정 패킷이 전송된다는 가정에 기초 한다[5, 6]. 포트 스캔을 탐지하기 위한 침입 탐지 시스템으로 NSM[10], SNORT[11-13], EMERALD[14], SPADE[15] 등이 있는데, 포트 스캔은 방화벽과 침입 탐지 시스템을 우회하며 발전하여 왔다.

NSM은 포트 스캔을 탐지하기 위해 개발된 최초의 네트워크 기반의 침입 탐지 시스템으로, 특정 시간에 같은 발신지 주소로 15개 이상의 연결을 시도하는 발신지 주소를 조사하여 공격을 탐지한다. 따라서 임계값 보다 느린 주기로 공격하는 Slow 스캔과 발신지 주소를 위조하여 공격하는 Coordina-

ted 스캔을 탐지하기 어렵다[10].

SNORT는 오픈 소스로써, 시그니처 기반의 네트워크 침입 탐지 시스템으로 패킷 캡쳐 라이브러인 libpcap을 사용하여 패킷을 수집하며, 다양한 공격을 탐지하고 광범위하게 사용된다. 이것은 다양한 형태로 패킷을 저장할 뿐만 아니라 실시간으로 경고를 한다. 주요 구성 요소는 전처리기, 탐지 엔진, 경고/로깅 모듈이며, 모든 구성요소는 플러그-인으로 구현되었기 때문에 유연성이 높다. SNORT에서 포트 스캔 탐지 기능은 전처리기 플러그-인에 의해 만들어진다. SNORT의 포트 스캔 전처리기는 특정한 시간 간격 동안 특정 수의 검사 패킷이 오는지 검사한다. 이 검사 패킷은 한 호스트만을 향할 수도 있고 여러 네트워크 머신으로 향할 수도 있다. 중요한 것은 특정 시간동안의 패킷 수가 어떤 임계값을 넘는 것이다. 만약 임계값을 넘는다면 포트 스캔 전처리기는 경고를 발생시킨다. SNORT의 포트 스캔 전처리기는 발신지 주소를 위조하여 공격하는 Coordinated 스캔을 탐지하기 힘들고, 임계값보다 느리게 연결 요청을 시도하는 Slow 스캔으로 쉽게 탐지를 피할 수 있다[12, 13].

EMERALD(Event Monitoring Enabling Responses to Anomalous Live Disturbances)는 포트 스캔을 탐지하기 위해 SRI International에서 개발한 시스템이다. EMERALD는 감시되는 네트워크에 연결된 발신지 주소를 관찰하고 통계적인 프로파일을 만들어, 짧은 주기의 프로파일과 긴 주기의 프로파일을 비교한다. 포트 스캔을 탐지하기 위해 SYN 패킷이 갑자기 증가한 발신지 주소를 찾아 공격을 탐지한다. 따라서 Slow 스캔 공격과 Coordinated 스캔 공격을 탐지할 수 없다[14].

SPADE(Statistical Packet Anomaly Detection Engine)는 Silicon Defence사에서 상업용으로 개발하였으며, SNORT의 전처리기 플러그-인으로 만든 네트워크 기반의 비정상 침입 탐지 시스템이다. 이것은 Stealthy 스캔만 다루기 때문에 TCP의 SYN 패킷만을 관찰한다. 비정상 스코어를 할당하기 위해 비정상 테이블을 유지하며, 평상시의 호스트별 포트들의 접근 빈도를 저장하여 두고 어떤 접근이 이루어질 때 자주 접근되지 않는 곳이면 침입 시도로 탐지한다. 포트 스캔 수집기는 비정상 탐지기와 다른 머신으로 분리되어 동작하며, 비정상 탐지기와 수집기의 커뮤니케이션은 소켓을 통하여 이루진다. SPADE는 자주 접근되지 않는 포트에 접근이 되면 침입으로 판단하기 때문에 자주 접근되었던 포트를 대상으로 하는 스캔은 탐지하기 힘들다[5].

3. 포트 스캔 탐지 시스템의 설계

본 논문의 공격 탐지 대상인 포트 스캔 공격 기법에 대하여 설명하고, 실제 공격을 분석하여 이를 기반으로 네트워크 통한 공격을 효율적으로 탐지하기 위한 공격 탐지 규칙을 제안하고 데이터 저장 구조를 설계하며 수집한 데이터를 이용한 공격 탐지 시스템을 설계한다.

3.1 포트 스캔 기법

포트 스캔은 공격하려는 시스템의 작동 여부와 시스템에서 제공하고 있는 서비스를 확인하기 위한 기법으로 ICMP, UDP, TCP 스캔 등이 있다[2, 5, 8, 16]. 본 논문에서는 TCP 스캔에 초점을 맞추어 연구하였는데, TCP 포트 스캔 기법의 종류는 TCP Open 스캔, TCP Half Open 스캔, Stealthy 스캔 등이 있다.

TCP Open 스캔은 가장 간단한 유형의 포트 스캔 방법으로 TCP의 3-단계 핸드셰이크를 기반으로 동작한다. 시스템은 스캔 시도에 대하여 포트가 열려 있는 경우와 닫혀 있는 경우에 다르게 동작한다. 공격 대상의 포트가 열려 있는 경우 다음과 같은 과정으로 동작한다. 첫째, 공격자가 공격 대상에게 연결 요청을 하는 SYN 패킷을 보낸다.

둘째, 공격 대상의 포트가 열려 있으면 연결 요청을 받아들이는 SYN/ACK 패킷을 보낸다. 셋째, 공격자는 ACK 패킷을 전송하여 3-단계 핸드셰이크로 동작한다. 공격 대상의 포트가 닫혀 있는 경우에는 다음과 같이 동작한다. 첫째, 공격자가 공격 대상에게 SYN 패킷을 보낸다. 둘째, 공격 대상은 포트가 닫혀 있기 때문에 연결 요청을 받아들이지 못하고 RST/ACK 패킷으로 응답한다. TCP Open 스캔은 가장 기본적인 스캔이며 정확한 결과를 얻을 수 있고 속도가 빠르지만, 시스템에 로그를 남기므로 공격에 대한 탐지가 용이하다[5, 8, 16].

Half-Open 스캔은 공격자가 공격 대상으로부터 SYN/ACK 패킷을 받은 경우 해당 포트가 열려 있는 상태만 확인하고 RST 패킷을 보냄으로써 완전한 연결을 성립하지 않게 하여 로그가 남지 않게 한다. Half-Open 스캔은 포트가 열려 있는 경우에 다음과 같은 과정으로 동작한다. 첫째, 공격자가 공격 대상에게 SYN 패킷을 보낸다. 둘째, 공격 대상의 포트가 열려 있으면 공격 대상에서 연결 요청을 받아들이는 SYN/ACK 패킷을 보낸다. 셋째, 공격자는 RST 패킷으로 응답함으로써 연결을 종료한다. 포트가 닫혀 있는 경우에는 다음과 같이 패킷을 전송한다. 첫째, 공격자는 공격 대상에게 SYN 패킷을 보낸다. 둘째, 공격 대상은 포트가 닫혀 있으므로 연결 요청을 받아들이지 못하고 RST/ACK 패킷으로 응답한다.

Half-Open 스캔은 3-단계 핸드셰이크로 동작하지 않고 SYN 패킷만 보내어 연결 설정을 확인한 후, 즉시 연결을 종료하므로 세션에 대한 로그를 남기지 않는다. SYN 패킷을 이용한 스캔 방법은 세션을 성립하기 위한 정당한 패킷과 구별할 수 없기 때문에 매우 효과적이다[5, 8, 16].

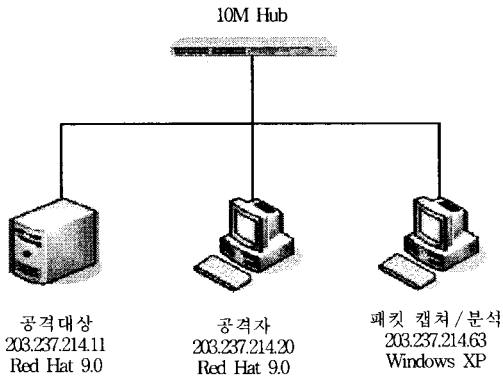
Stealthy 스캔은 Half-Open 스캔처럼 상대방 시스템에 로그를 남기지 않는 것뿐만 아니라, 공격 대상을 속이고 자신의 위치를 숨기는 스캔을 말한다. Stealthy 스캔은 FIN, NULL, XMAS 스캔 등이 있는데, FIN 스캔은 연결 종료를 위해 사용되는 FIN 플래그를 설정하여 스캔을 시도하고, NULL 스캔은 플래그(PSH, URG, SYN, RST, FIN, ACK)를 설정하지 않고 NULL 상태로 시도하는 스캔이다. XMAS 스캔은 NULL 스캔과 반대로 모든 플래그를 설정하여 시도하는 스캔이다. Stealthy 스캔은 포트가 열려 있는 경우, 공격자가 공격 대상에게 FIN, NULL, XMAS 패킷을 보내면 공격 대상은 아무런 응답이 없다. 포트가 닫혀 있는 경우에는 첫째, 공격자는 공격 대상에게 FIN, NULL, XMAS 패킷을 보낸다. 둘째, 공격 대상은 포트가 닫혀 있으므로 연결 요청을 받아들이지 못하여 RST/ACK 패킷으로 응답한다. Stealthy 스캔의 FIN, NULL, XMAS 스캔은 비정상적으로 패킷의 플래그들을 설정하여 방화벽을 통과하고 침입 탐지 시스템을 회피한다.

시간차를 이용한 Slow 스캔과 발신지 주소를 위조하는 Coordinated 스캔이 있는데, Slow 스캔은 아주 긴 시간 간격으로 패킷을 보내어 방화벽과 침입 탐지 시스템이 스캔 공격의 패턴에 대한 정보를 얻기 힘들게 하여 공격 탐지를 어렵게 하는 스캔이며, Coordinated 스캔의 경우는 발신지 주소를 위조하여 패킷을 보내기 때문에 공격자의 주소를 확인하기 어려워 공격을 탐지하기가 어려운 스캔이다[5, 8, 16].

3.2 포트 스캔 공격 분석

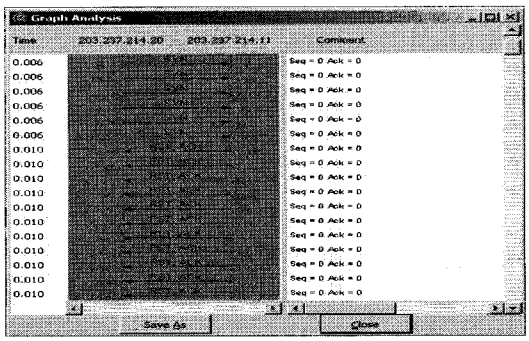
본 절에서는 실제로 포트 스캔 공격을 시도하고 분석하였는데, 실험 환경은 허브에 직접 연결되어 있는 Linux 시스템의 공격자와 공격 대상이 같은 네트워크에 속해 있는 (그림 1)과 같은 환경으로 가정하였다.

포트 스캔 도구로는 현재 가장 널리 사용되는 Nmap을 이용하여 공격하였고, 네트워크 트래픽 모니터링 도구로는 Ethereal을 사용하였다. Nmap을 통하여 공격을 시도하고 네트워크 상의 패킷 캡춰 도구인 Ethereal을 이용하여 패킷을 캡쳐하고 그 결과를 TCP Flow 그래프로 나타내었다.



(그림 1) 실험 환경

Nmap의 -sS 옵션을 이용하여 공격자가 공격 대상에 SYN 패킷을 보내어 Half Open 스캔을 시도하면 공격 대상의 대기하기 않는 포트에서 (그림 2)와 같이 RST/ACK 패킷이 발생하는 것을 볼 수 있다.



(그림 2) Half Open 스캔의 공격 결과

Stealthy 스캔의 Fin, Null, Xmas 스캔 공격을 위하여 Nmap의 -sF, -sN, sX의 옵션으로 공격자 주소에서 공격 대상으로 공격을 시도하였는데, Half Open 스캔의 경우처럼 공격 대상의 대기하지 않는 포트에서 RST/ACK가 발생하는 것을 확인할 수 있다. Half Open 스캔의 경우처럼 공격 대상의 대기하지 않는 포트에서 RST/ACK가 발생하는 것을 확인할 수 있다.

이와 같은 공격을 기반으로 포트 스캔 공격의 패턴을 요약하면 다음과 같다. TCP Open 스캔, Half Open 스캔, FIN 스캔, NULL 스캔, XMAS 스캔, Slow 스캔은 공격자 주소와 공격 대상 주소가 고정되고, 공격 대상 포트가 변한다. Coordinated 스캔은 공격자 주소를 위조하여 공격하기 때문에 공격자 주소만 변하고 다른 것은 포트 스캔 공격과 동일하다. 이러한 공격 패턴을 정리하면 <표 1>과 같다.

〈표 1〉 공격 패턴

구분	공격자 주소	공격자 포트	공격 대상 주소	공격 대상 포트
포트 스캔	F	V	F	V
Coordinated 스캔	V	V	F	V

주) F : 고정, V : 가변

3.3 데이터 수집과 공격 탐지 규칙

포트 스캔 공격은 TCP의 정상적인 연결 설정의 세션을 완성하지 않고, 비정상적인 연결 시도로 공격을 한다. Half Open 스캔의 경우, 공격자가 SYN 패킷을 보내어 공격 대상에 연결 설정을 요청할 때 포트가 열려 있으면 공격 대상은 정상적인 SYN/ACK로 응답하는데 공격자는 RST 패킷을 보내어 연결을 강제 종료하고, 포트가 닫혀있으면 공격 대상은 RST/ACK 패킷으로 응답한다. 이와 같이 비정상적인 연결을 시도하면 RST 패킷이 발생하는데, 연결 설정 과정에서 RST 패킷이 발생하면 공격으로 가정할 수 있기 때문에, 본 논문에서는 비정상적인 연결 시도에서 발생하는 RST 플래그를 가진 패킷, 즉 RST 패킷을 수집한다.

RST 패킷은 TCP가 정상적인 상태에서도 발생할 수 있는데[17], RST 패킷이 발생하는 경우는 다음과 같다. 첫째, 목적지 시스템의 포트에 해당되는 프로세스가 대기하고 있지 않거나, 교환되는

패킷의 순서번호가 틀리면 발생한다. 둘째, 정상적인 연결이 이루어졌어도 기존에 연결된 패킷의 순서번호가 충돌되거나 이중 연결 설정을 시도하면 이전 연결 상태를 종료하기 위하여 보내진다. 셋째, 연결 설정이 이루어진 후에 긴 시간동안 휴지 상태에 있으면 연결을 종료하기 위해 전송한다[17]. 첫 번째는 비정상적인 상태에서 발생하는 경우로써 공격으로 판단할 수 있지만, 두 번째와 세 번째는 정상적인 상태에서 발생하는 경우로 공격으로 판단할 수 없다. 따라서 본 논문에서는 RST 패킷이 발생하면 정상적인 연결 설정이 이루어졌는지를 검사하여, 정상적인 연결 설정이 이루어지지 않은 패킷을 수집한다.

포트 스캔 공격에 있어서 RST 패킷이 발생되는 경우는 다음과 같다. 첫째, 공격자가 SYN 스캔을 시도하면, 공격 대상 시스템의 포트가 열려 있다면 공격 대상에서 SYN/ACK 패킷을 전송하지만, 공격자는 연결 설정을 하지 않기 위해서 RST 패킷을 보내 연결을 강제로 종료한다. 포트가 닫혀 있다면 공격 대상은 공격자에게 RST/ACK 패킷을 보낸다. 둘째, 공격자가 FIN, NULL, Xmas 스캔과 같은 Stealthy 스캔을 시도하면, 포트가 닫혀 있으면 공격 대상이 공격자에게 RST/ACK 패킷을 보낸다.

〈표 2〉 공격의 종류에 따른 RST 패킷

공격의 종류	RST 패킷 발생
Half Open, Slow, Coordinated 스캔	열린 포트 - 공격자에서 RST 발생 닫힌 포트 - 공격 대상에서 RST/ACK 발생
Stealthy (FIN, NULL, Xmas) 스캔	닫힌 포트 - 공격 대상에서 RST/ACK 발생

공격 패턴의 분석에 따른 공격 탐지 규칙은 다음과 같다. <표 1>의 공격 패턴을 보면, 공격자가 포트 스캔을 할 때 공격 대상의 주소를 고정시키고 공격 대상의 포트를 변화시켜 공격을 시도한다. 이때 공격 대상의 변화되는 포트가 대기하지 않으면 RST/ACK 패킷이 발생한다. 따라서 공격 대상의 주소에서 발생되는 RST/ACK 패킷의 발신지 주소가 고정되고 발신지 포트가 변하는 패킷을 탐지하여 포트 스캔 공격을 탐지한다. 공격의 종류에 따라 RST 패킷이 발생되는 경우를 요약하면 <표 2>와 같다.

3.4 포트 스캔 탐지 시스템의 설계

본 절에서는 실제 공격을 통하여 생성된 공격 탐지 규칙을 기반으로 포트 스캔 탐지 시스템을 설계한다. 네트워크 상의 모든 패킷을 수집하여 저장하는 패킷 저장 구조와 공격 탐지 규칙을 적용하기 위한 RST/ACK 패킷 정보 저장 구조에 대해 살펴보면 다음과 같다.

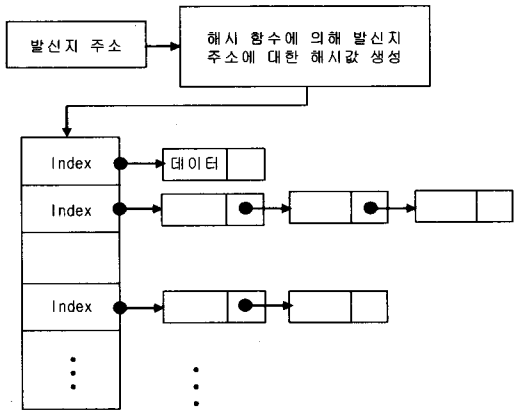
3.4.1 패킷 저장 구조

TCP 패킷의 헤더의 발신지 주소로 해쉬하여 해쉬 값을 얻고 해쉬 테이블의 인덱스에 접근하여 정보를 저장한다. 만약 기존에 저장된 정보가 있으면, 기존 노드의 다음에 정보를 저장하고 연결 리스트로 연결한다. 저장되는 정보는 발신지 주소, 발신지 포트, 목적지 주소, 목적지 포트, 플래그이다. 플래그는 연결 설정 과정에서 주고받는 플래그 비트이다. 저장된 정보의 데이터 구조는 (그림 3)과 같다.

발신지주소	발신지포트	목적지주소	목적지포트	플래그	Next

(그림 3) 저장된 정보의 구조

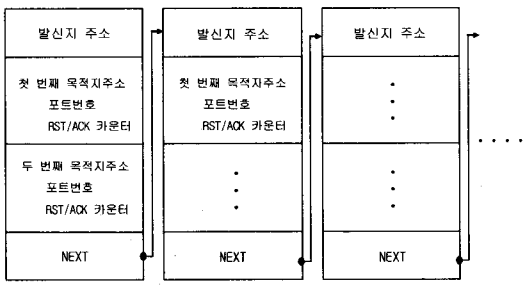
저장된 정보는 RST/ACK 패킷이 발생했을 경우 연결 설정을 검사하기 위해 사용되는데, 발신지 주소가 같은 패킷의 정보는 해쉬 테이블의 인덱스에 의해 연결 리스트로 연결되어있기 때문에, 빠르게 접근하여 정보를 찾을 수 있다. 패킷의 정보 저장 구조는 (그림 4)와 같다.



(그림 4) 패킷 정보 저장 구조

3.4.2 RST/ACK 패킷 정보 저장 구조

RST/ACK 패킷이 발생하면 정상적인 연결 시도에서 발생한 것인지, 비정상적인 연결 시도에서 발생한 것인지 검사하고, 비정상적인 연결 시도에서 발생한 것이라면 발신지 주소를 검사하여 기존에 저장된 정보가 없다면 정보를 새로운 노드를 만들어 저장하고 연결 리스트로 연결한다, 만약 기존에 저장된 정보가 있다면, 기존 정보에 저장하는데, 저장되는 정보는 발신지 주소, 첫 번째 목적지 주소, 첫 번째 포트 번호, RST/ACK 카운터, 두 번째 목적지 주소, 두 번째 목적지 포트 번호, RST/ACK이다.



(그림 5) RST/ACK 패킷 정보 저장 구조

RST/ACK 패킷의 정보 저장 구조는 (그림 5)와 같다. 첫 번째와 두 번째를 나눈 이유는 Coordinated 스캔을 탐지하기 위한 것으로 Coordinated 스캔은 발신지 주소를 위조하여 공격하기 때문에, 같은 발신지 주소에서 발생한 RST/ACK 패킷의 목적지 주소 세 개가 다르면 Coordinated 스캔으로 탐지한다. 같은 발신지 주소에서 발생한 RST/ACK 패킷의 목적지 주소가 같다면 RST/ACK 패킷의 카운터를 증가시켜 포트 스캔을 탐지한다.

4. 포트 스캔 탐지 시스템의 구현 및 결과

비정상 연결 시도를 탐지한 포트 스캔 공격을 탐지할 수 있는 시스템을 구현하고, 구현된 시스템에 포트 스캔 공격을 시도하여 공격이 탐지됨을 확인하였다.

4.1 포트 스캔 탐지 시스템의 구현

포트 스캔 탐지 시스템은 Linux 환경에서 네트워크상의 패킷을 캡쳐하기 위해 패킷 캡쳐 드라이버인 libpcap[13]을 사용하여 혼잡모드로 네트워크상의 모든 패킷을 수집하였으며 C 언어로 구현하였다. 본 논문에서 포트 스캔 공격을 탐지하기 위하여 구현한 포트 스캔 탐지 시스템은 패킷 정보 저장 모듈, 연결 설정 저장 모듈, RST/ACK 패킷 정보 저장 모듈, 포트 스캔 탐지 모듈 등으로 구성된다.

4.1.1 패킷 정보 저장 모듈

패킷 정보 저장 모듈은 패킷 캡쳐 드라이버에서 캡쳐된 패킷을 검사하여 TCP 패킷의 헤더 정보를 저장하는 모듈이다.

캡쳐한 패킷은 임시로 저장될 정보의 구조체를 할당하고, 캡쳐된 패킷의 헤더 정보를 저장한다. 저장된 발신지 주소로 해쉬하여 해쉬 값을 얻고, 해쉬 테이블의 인덱스에 접근한다. 만약 발신지

주소에 저장된 값이 있으면, 기존의 저장된 정보에 연결 리스트로 연결하고, 저장 정보가 없으면, 새로운 노드를 할당하고 저장한다. 정보를 저장한 후 플래그 값이 RST/ACK 인지 확인하고, 만약 있다면 연결 설정 검사 모듈로 이동한다.

```
Algorithm 4.1 패킷 정보 저장 모듈
if (TCP packet) {
    1. temp 구조체에 패킷의 헤더 정보 저장
    2. 발신지 주소 해쉬, 해쉬 값으로 해쉬 테이블
       의 Index 접근
    if (저장 정보 존재) {
        1. host 구조체 동적 할당
        2. temp 구조체 정보를 host 구조체에 저장
        3. 기존의 저장된 정보에 연결 리스트로 연결
           if (RST/ACK 플래그) {
              1. 연결 설정 검사 모듈로 이동
           }
        } else {
              1. host 구조체 동적 할당
              2. temp 구조체 정보를 host 구조체에
                 저장
           if (RST/ACK 플래그) {

              1. 연결 설정 검사 모듈로 이동
           }
        }
}
```

(그림 6) 패킷 정보 저장 모듈

4.1.2 연결 설정 검사 모듈

연결 설정 검사 모듈은 RST 플래그가 정상적인 연결에서 발생한 것인지, 비정상적인 연결에서 발생한 것인지 검사하는 모듈이다.

RST/ACK 패킷의 목적지 주소로 해쉬하여 해쉬 값을 얻고, 해쉬 테이블을 검사하여 저장 정보에 접근한다. 저장된 정보에 SYN 플래그와 ACK 플래그가 있는지 검사한다. SYN 플래그와 ACK 플래그가 없다면 RST/ACK 패킷 저장 모듈로 이동하고, SYN 플래그와 ACK 플래그가 있다면, RST/ACK 패킷의 발신지 주소를 해쉬하여 해쉬 값을 얻고, 해쉬 테이블을 검사하여 저장 정보에 접근한다. 저장된 정보에서 SYN/ACK 패킷이 있는지 검사하고 있다면 발신지 주소와 목적지 주소의 해쉬 테이블 정보를 삭제한다. 만약 SYN/ACK 패킷이 없다면 RST/ACK 패킷 저장 모듈로 이동한다.

```
Algorithm 4.2 연결설정 검사 모듈
1. 패킷 저장 정보에 목적지 주소로 접근
2. SYN 패킷과 ACK 패킷 검사
if (SYN 패킷과 ACK 패킷) {
    1. 패킷 저장 정보에 발신지 주소로 접근
    2. SYN/ACK 패킷이 있는지 검사
    if (SYN/ACK 패킷) {
        1. 발신지 주소와 목적지 주소의 패킷 저장
           정보 삭제
    } else {
        1. Return 연결 설정 검사 False
        2. RST/ACK 패킷 저장 모듈로 이동
    }
else {
        1. 연결 설정 검사 False 리턴
        2. RST/ACK 패킷 저장 모듈로 이동
}
```

(그림 7) 연결 설정 검사 모듈

4.1.3 RST/ACK 패킷 정보 저장 모듈

RST/ACK 패킷 정보 저장 모듈은 공격 탐지 규칙을 적용하기 위하여 RST/ACK 패킷을 저장하는 모듈이다. RST/ACK 패킷의 정보 저장 연결 리스트에서 발신지 주소를 검사하여 발신지 주소가 있다면 포트 스캔 탐지 모듈로 이동한다.

```
Algorithm 4.3 RST/ACK 패킷 정보 저장 모듈
if (ip_state 구조체의 연결 리스트에 발신지 주소) {
    1. 포트 스캔 탐지 모듈로 이동
} else {
    1. ip_state 구조체 동적 할당
    2. temp 구조체의 정보를 ip_state 구조체에 저장
}
```

(그림 8) RST/ACK 패킷 정보 저장 모듈

4.1.4 포트 스캔 탐지 모듈

포트 스캔 탐지 모듈은 저장된 RST/ACK 패킷의 정보를 갱신하며, 목적지 주소를 비교하여 같으면 RST/ACK 카운터를 증가시켜, 카운터가 임계값과 일치하면 포트 스캔 공격을 탐지하는 모듈이다.

temp 구조체의 목적지 주소와 RST/ACK 패킷의 저장된 정보의 첫 번째 목적지 주소나 두 번째 목적지 주소가 같으면 RST/ACK 카운터를 증가시킨다. 임계값과 RST/ACK 카운터가 같다면 포트 스캔 경보를 발생시키고, 저장된 RST/ACK 패킷 정보를 삭제한다. 만약 두 번째 목적지 주소가 NULL이라면 temp 구조체의 두 번째 목적지 주소와 목적지 포트를 저장하고 RST/ACK 카운터를 증가시킨다. 만약 temp 구조체의 목적지 주소가 첫 번째와 두 번째의 목적지 주소와 일치하지 않는다면 Coordinated 스캔 경보를 발생시키고, 저장된 RST/ACK 패킷 정보를 삭제한다.

```
Algorithm 4.4 포트 스캔 탐지 모듈
if (temp 구조체의 목적지 주소 == 첫 번째 목적지
   주소) {
    1. RST/ACK 카운터 증가
     if (RST/ACK 카운터 == 임계값) {
          1. 포트 스캔 경보 발생
          2. 저장된 RST/ACK 패킷 정보 삭제
    }
} else if (temp 구조체의 목적지 주소 = =두번째
   목적지 주소) {
    1. RST/ACK 카운터 증가
    if (RST/ACK 카운터 == 임계값) {
          1. 포트 스캔 경보 발생
          2. 저장된 RST/ACK 패킷 정보 삭제
    }
    } else if (두 번째 목적지 주소가 NULL) {
          1. 목적지 주소와 목적지 포트 저장
          2. RST/ACK 카운터 증가
    }
else {
          1. Coordinated 스캔 경보 발생
          2. 저장된 RST/ACK 패킷 정보 삭제
}
```

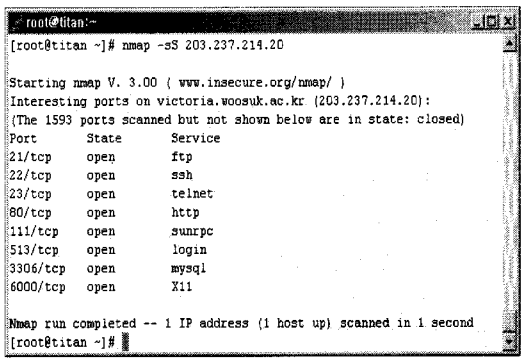
(그림 9) 포트 스캔 탐지 모듈

4.2 포트 스캔 탐지 시스템의 구현 결과

공격 대상의 호스트에 구현된 포트 스캔 탐지 시스템을 실행하고, 공격자가 공격 대상의 호스트에 포트 스캔 공격을 시도하여 포트 스캔 탐지 시스템의 탐지 결과를 확인하였다. 공격자와 공격 대상은 허브를 통하여 같은 네트워크에 연결되어 있으며, 공격자의 주소는 203.237.214.22이고, 공격 대상의 주소는 203.237.214.20이다. 구현된 포트 스캔 탐지 시스템은 공격 대상의 호스트에 설치하여 실험하였으며, 공격 대상의 호스트에 공격이 시도되면, 설치된 포트 스캔 탐지 시스템이 공격을 탐지한다.

4.2.1 Half Open 스캔 공격 탐지

Nmap의 -sS 옵션으로 공격자의 주소 203.237.214.22에서 공격 대상의 주소 203.237.214.20으로 Half Open 스캔 공격을 시도한 결과는 (그림 10)과 같다. 공격자는 공격 대상의 주소에 21, 22, 23, 80, 111, 513, 3306, 6000번의 포트가 열려 있는 것을 확인할 수 있다.

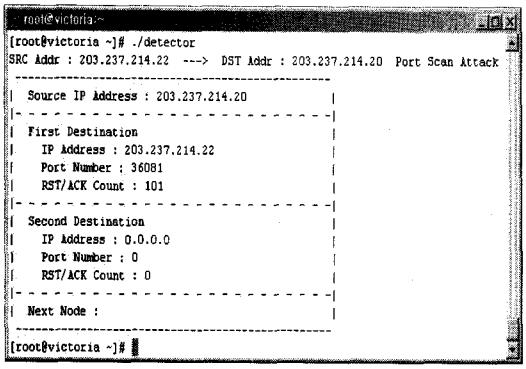


(그림 10) Half Open 스캔 공격 시도

공격 대상의 주소에 포트 스캔 탐지 시스템을 설치하고, 공격자가 Half Open 스캔 공격을 시도할 때 공격을 탐지한 결과는 (그림 11)과 같다.

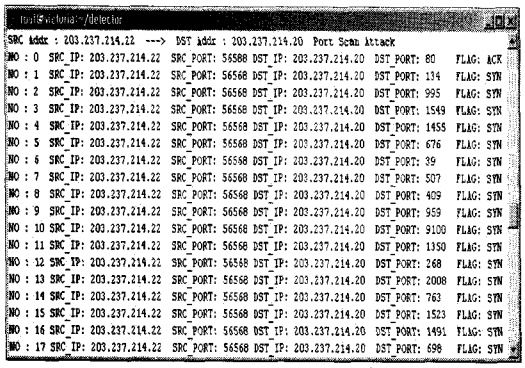
공격자의 주소 203.237.214.22에서 공격 대상의 주소 203.237.214.20으로 포트 스캔 공격을 시도했

다는 경보가 발생하고, RST/ACK 패킷의 발신지 주소, 목적지 주소, RST/ACK 패킷 발생 횟수의 정보를 보여주어 포트 스캔 공격에 대한 정보를 확인할 수 있다.



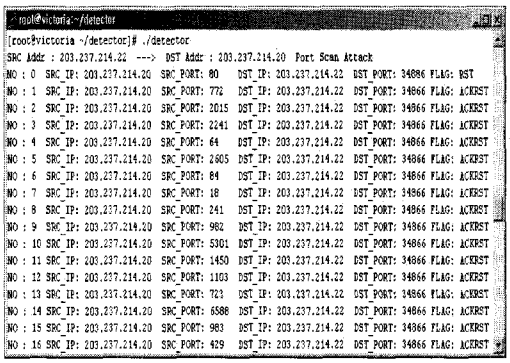
(그림 11) Half Open 스캔 탐지 결과

Half Open 스캔 공격시 공격자의 주소에서 발생한 패킷을 저장한 결과는 (그림 12)와 같다. 발신지 주소가 같은 패킷의 정보가 연결 리스트로 연결되어 순차적으로 저장되었다. 저장된 정보는 발신지 주소, 발신지 포트, 목적지 주소, 목적지 포트, 플래그이다.



(그림 12) 공격자에서 발생한 패킷

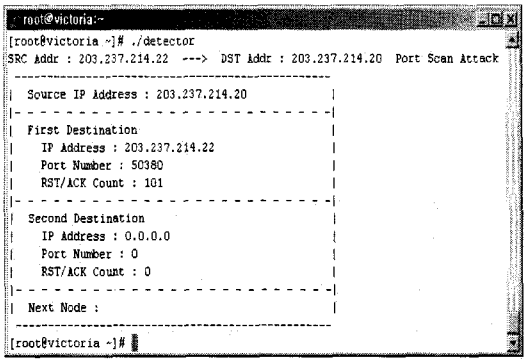
공격자가 Half Open 스캔 공격을 시도할 때, 공격 대상의 주소에서 발생한 패킷을 저장한 결과는 (그림 13)과 같다.



(그림 13) 공격 대상에서 발생한 패킷

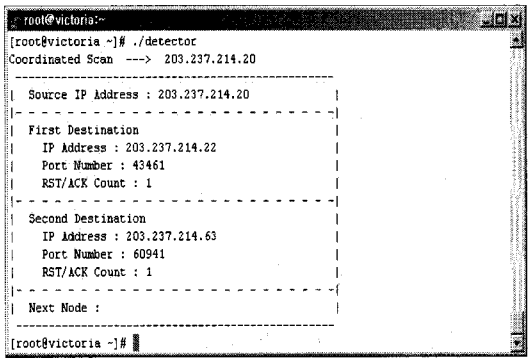
4.2.2 Stealthy 스캔 공격 탐지

Stealthy 포트 스캔 공격 중 FIN 스캔의 탐지 결과이다. Nmap의 -sF 옵션으로 FIN 스캔 공격을 시도한 후 FIN 스캔 공격을 탐지한 결과와 Nmap의 -sN 옵션으로 NULL 스캔 공격을 시도한 후 NULL 스캔 공격을 탐지한 결과 및 Nmap의 -sX 옵션으로 공격을 시도한 후 XMAS 스캔 공격을 탐지한 결과는 Half Open 스캔 공격을 시도한 결과 유사하게 나타남을 확인하였다. Stealthy 포트 스캔 공격 중 Slow 스캔 공격의 탐지 결과이다. Nmap의 -T 옵션으로 공격 시간의 간격을 주어 Slow 스캔 공격의 탐지 결과는 (그림 14)와 같다.



(그림 14) Slow 스캔 공격 탐지 결과

Nmap의 -S 옵션으로 공격자의 주소를 위조하고, -p 옵션으로 포트 번호를 바꾸어 가며 Coordinated 스캔 공격을 시도하고 탐지한 결과는 (그림 15)와 같다.



(그림 15) Coordinated 스캔 공격 탐지 결과

5. 결론 및 향후 과제

본 논문에서는 네트워크를 통한 포트 스캔 시도를 위하여 TCP의 비정상 연결 시도에서 발생하는 RST/ACK 플래그를 탐지하여 이를 기반으로 시스템을 설계하였다. 기존의 침입 탐지 시스템은 포트 스캔 공격을 탐지하기 위해서 동일 발신지 주소를 갖는 시스템에서 특정시간에 일정 임계값을 초과하는 연결 설정 요청 패킷이 발생했는지를 검사하여 공격을 탐지하기 때문에 쉽고 빠르기는 하지만 다음과 같은 단점을 가지고 있다. 첫째, 실제 공격이 아닌데 공격이라고 탐지하는 False Positive가 높다. 둘째, 특정시간의 임계값 보다 더 긴 주기로 공격을 시도하는 Slow 스캔 공격의 경우, 일정시간에 캡쳐해야 하는 다량의 데이터로 인하여 탐지하기가 힘들다. 셋째, 발신지 주소를 위조하여 공격하는 Coordinated 스캔 공격의 경우에는 발신지 주소가 동일한 주소가 아니기 때문에 탐지하기가 어렵다.

본 논문에서는 이런 문제점을 개선하기 위해 TCP의 비정상 연결 시도에서 발생하는 RST/ACK 패킷을 탐지하여 포트 스캔 탐지 규칙을 생성하였다. 포트 스캔 탐지 규칙은 RST/ACK 패킷의 발신지 주소는 고정되고 발신지 포트가 변하는 패킷을 탐지한다. 이를 기반으로 네트워크 상에서 캡쳐한 RST/ACK 패킷을 저장하고 RST/ACK 패킷의 개수를 누적시키고 탐지 규칙을 적용하여 포트 스캔 공격을 탐지하였는데, 본 논문에서 제안한 포트 스캔 탐지 시스템은 다음과 같은 특징이 있다.

첫째, TCP의 비정상 연결 시도에서 발생하는 RST/ACK 패킷을 탐지하여 공격을 탐지하므로 기존 시스템 보다 False Positive를 감소시킨다. 둘째, RST/ACK 패킷의 개수를 가지고 공격을 탐지하기 때문에 적은 양의 데이터를 저장하여 긴 시간을 유지할 수 있으므로, Slow 스캔을 탐지할 수 있다. 셋째, Coordinated 스캔은 공격자가 위조된 주소로 공격을 시도하는데, 공격 대상이 같기 때문에 공격 대상에서 발생하는 RST/ACK 패킷을 검사하여 공격을 탐지할 수 있다.

향후에 연구되어야 할 과제로는 공격이 탐지되었을 때 공격을 효과적으로 방어하기 위한 연구와 RST 패킷과 네트워크 공격과의 관계를 체계적으로 조사하여 본 논문에서 제안한 포트 스캔 탐지 시스템을 확장하는 것이다. 또한 실제 네트워크 환경에 적용하기 위해 보다 정확히 공격을 판단할 수 있는 임계값의 설정과 다양한 환경에서 성능 분석이 필요하다.

참 고 문 헌

[1] Aleksandar Lazarevic, Vip Kummar, Jaideep Srivastava, "Managing Cyber Threats : Issues, Approaches and Challenges", pp. 19-78, Springer 2005.

[2] Chares P. Pfleeger, Shari Lawrence Pfleeger, "Security in Computing 3/E", Prentice Hall,

2002.

[3] R. Bace and P. Mell, "NIST Special Publication on Intrusion Detection System", 2001.

[4] Emilie Lundin and Erland Jonsson, "Survey of Intrusion Detection Research", Chalmers University, technical report 02-04, 2002.

[5] Stuart Staniford, James A. Hoagland, and Joseph M. McAlerney, "Practical automated detection of stealthy portscans", Journal of Computer Security, pp. 105-136, 2002.

[6] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing", Proceedings of the IEEE Symposium on Security and Privacy, 2004.

[7] Stefan Axelsson, "Research in Intrusion Detection System : A Survey", Chalmers University, technical report 98-17, 1998.

[8] Stephen Northcutt, Judy Novak, "Network Intrusion Detection An Analyst's Handbook 2/E", New Riders, 2000.

[9] V. Paxson, "Bro : A System for Detecting Network Intruders in Real-Time", in Proceedings of the 7th USENIX Security Symposium, San Antonio, January 1998.

[10] L. T. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A network security monitor", 1990 Symposium on Research in Security and Privacy, pp. 296-304, Oakland, 1990.

[11] Brian Caswell, Jay Beale, James C. Foster, "Snort Intrusion Detection 2.0", Syngress, 2003.

[12] Martin Roesch, "Snort-Lightweight Intrusion Detection for network", proceedings of LISA '99, 13th System Administration Conference, 1999.

[13] http://www.snort.org.

[14] P. Porras and A. Valdes, "Live traffic analysis of TCP/IP gateways", 1888 Internet Society Symposium on Network and Distributed System Security, San Diego, 1998.

[15] http://www.silicondefense.com/software/spice/index.html.

[16] Fyodor, "The Art of Port Scanning", Phrack Magazine, Vol. 7, Issue 51, 1997.

[17] Jon Postel, Transmission Control Protocol, RFC 793, 1981.

라 용 환

2005년 우석대학교 컴퓨터공학과 졸업(공학사)

2007년 우석대학교 컴퓨터공학과 졸업(공학석사)

2007년~현재 (주)STGSecurity 연구원

천 은 홍

1981년 광운대학교 응용전자공학과 졸업(공학사)

1985년 아주대학교 전자공학과 (공학석사)

1998년 아주대학교 컴퓨터공학과 (공학박사)

2005년~2006년 미국 Univ. of Louisiana 객원 교수

1988년~2006년 우석대학교 컴퓨터공학과 교수

2006년~현재 우석대학교 컴퓨터교육과 교수