한국시뮬레이션학회 논문지
Vol. 21, No. 4, pp. 91-102 (2012. 12)
http://dx.doi.org/10.9709/JKSS.2012.21.4.091

네트워크 포트스캔의 위험에 대한 정량화 방법

박성철[1†] · 김준태[1]

A Method for Quantifying the Risk of Network Port Scan

Seongchul Park · Juntae Kim

ABSTRACT

Network port scan attack is the method for finding ports opening in a local network. Most existing IDSs(intrusion detection system) record the number of packets sent to a system per unit time. If port scan count from a source IP address is higher than certain threshold, it is regarded as a port scan attack. The degree of risk about source IP address performing network port scan attack depends on attack count recorded by IDS. However, the measurement of risk based on the attack count may reduce port scan detection rates due to the increased false negative for slow port scan. This paper proposes a method of summarizing 4 types of information to differentiate network port scan attack more precisely and comprehensively. To integrate the riskiness, we present a risk index that quantifies the risk of port scan attack by using PCA. The proposed detection method using risk index shows superior performance than Snort for the detection of network port scan.

Key words : Intrusion Detection System, Port Scan, Risk Index, Principal Component Analysis

요 약

네트워크 포트스캔 공격은 내부 네트워크에 있는 시스템에서 열려 있는 포트를 알아내기 위한 방법이다. 기존 대부분의 침입탐지시스템(Intrusion Detection System; IDS)들은 단위 시간당 시스템 또는 네트워크에 몇 번의 패킷을 보냈는지의 횟수를 기록하여 전송한 패킷의 횟수가 임계치보다 높은 소스 인터넷 주소(source IP address)에 대해서 포트스캔 공격이 수행되었다고 간주하였다. 즉, 네트워크 포트스캔 공격을 수행한 소스 인터넷 주소에 대한 위험 정도는 IDS들이 기록한 포트스캔 공격 횟수에 의존하였다. 그러나 단순히 포트스캔 공격 횟수에 기반을 둔 위험성의 측정은 느린 포트스캔 공격에 대해 거짓 부정(false negative)이 높아져 포트스캔 탐지율이 낮아진다는 문제가 있다. 본 연구에서는 네트워크 포트스캔 공격에 대해 좀 더 정확하고 포괄적인 구분을 하기 위해 4가지 형태의 정보를 요약한다. 포트스캔 공격에 대한 위험성을 집약적으로 나타내기 위하여 주성분분석(principal component analysis, PCA)에 의해 이러한 정보들을 정량화한 위험지수를 제안한다. 실험을 통해 제안한 위험지수를 이용한 탐지가 포트스캔 탐지율에 있어서 Snort보다 우수하다는 것을 보인다.

주요어 : 침입탐지시스템, 포트스캔, 위험지수, 주성분분석

1. 서 론

인터넷의 발달로 인해 정보 공유 및 필요한 정보 수집이 더욱 편리해지고 있으나, 그 이면에는 수많은 해커들에 의해 불법적인 정보 공유 및 정보 수집이 이루어져 인터넷을 사용하는 사용자들에게 불안감을 확산시키며 좀 더 자유롭고 활발한 인터넷상의 활동을 저해하고 있다(Lazarevic, 2005). 해커들이 해킹을 시도하기 위해서는 공격 대상에 해당하는 시스템이나 네트워크에 대한 정보를 수집하는 포트스캔 단계가 필수적이다. 포트스캔은 침입 대상이 되는 시스템에 어떤 포트가 열려있는 지를 찾는 것을 말하며 그 포트스캔이 네트워크나 시스템에 직접적인 큰 피해를 주지는 않지만 침입 대상 시스템에 있는 취약한 포트에 대한 정보를 수집하는 첫 번째 단계에 해당한다. 공격자는 포트스캔을 통해 실질적인 공격을 위해

접수일(2012년 8월 23일), 심사일(1차 : 2012년 11월 19일), 게재 확정일(2012년 12월 18일)
[1] 동국대학교 컴퓨터공학과
주 저 자 : 박성철
교신저자 : 김준태
E-mail; jkim@dongguk.edu

침입 대상 시스템의 취약점을 분석하여 가장 취약하거나 공격이 용이한 포트를 선택하게 된다(Fyodor, 1997).

포트스캔은 TCP/IP 프로토콜 스택의 여러 가지 성질을 이용하여 어떠한 포트가 열려 있고 어떠한 서비스를 제공하고 있는지 알아내는 것으로서, TCP/IP 프로토콜의 레이어(layer)중 트랜스포트 레이어(transport layer)에 해당하는 TCP 헤더와 UDP 헤더를 이용한다. 포트스캔은 스캔방식에 따라 TCP 헤더의 성질을 이용하는 스캔으로는 완전 연결(open) 스캔, 불완전 연결(half open) 스캔, 또는 스텔스 스캔(stealth scan)이 있으며, 그리고 UDP 헤더의 성질을 이용하는 UDP 스캔 등으로 나눌 수 있다(IANA, 2010).

TCP 헤더의 성질을 이용한 포트스캔을 분류하는 기준은 3-way handshakes에 의해 완전 연결(open)인지 아니면 불완전 연결(half open) 인지로 나뉘며 이들은 각각 Open Scan 또는 Half-Open Scan이라 부른다. Open Scan 또는 Half-Open Scan은 3-way handshakes의 SYN, SYN/ACK, 그리고 ACK의 3단계 중 전체 단계에 관여되어 있는지 아니면 1단계에만 이용하는지를 구분하는 것이다.

TCP 헤더의 성질을 이용한 포트스캔을 분류하는 또 다른 기준은 침입탐지시스템이나 침입차단시스템을 우회하기 위한 방법으로 3-way handshakes에 사용하는 SYN 외의 플래그들(flags) 중 독자적인 ACK 플래그, FIN 플래그 또는 NULL 플래그를 이용한 스캔이 있고 이들을 스텔스 스캔이라 칭한다. URG 플래그, FIN 플래그 및 PSH 플래그를 모두 혼합한 Xmas Scan이라고 부르는 스텔스 스캔도 존재한다. 또 다른 스텔스 스캔은 TCP 헤더의 윈도우 사이즈, 또는 TCP Fragmentation을 이용한다.

포트스캔 공격은 수신된 패킷들의 횟수나 포트스캔 시그너처에 의해 수신된 패킷을 패턴 매칭하여 그 패킷의 수가 보안 관리자에 의해 설정된 임계값 이상인 것을 검사함으로써 탐지해 낼 수 있다. 문제는 거짓 긍정과 거짓 부정을 어떻게 줄일 수 있는가이다. 포트스캔 공격탐지 시스템에서 거짓 긍정이라는 것은 어떤 프로그램에서 포트스캔을 의도하지는 않았지만 포트스캔 공격탐지 시스템에서 의도된 포트스캔으로 인지하는 경우를 말한다. 이러한 거짓 긍정은 룰을 어떻게 설정하느냐에 따라 정상적인 행위가 될 수도 있고 공격이 될 수도 있다. 예를 들어 Snort는 외부 호스트를 포트스캔 공격으로 탐지하기 위해 T 초에서 N 연결 시도를 검사한다. 엄격한 포트스캔 공격을 탐지하기 위해 T를 60으로 두고 N을 10으로 둔다고 가정했을 때 60초 내에 10번 이상 포트스캔이 탐지된다면 포트스캔 공격으로 간주한다. 그런데 로컬 네트워크 내에 패킷을 많이 발생시키는 어떤 호스트 내에 정상적인 프로그램이 존재한다면 포트스캔 공격을 일으킨다고 보고될 것이다. 그러므로 이러한 거짓 긍정을 줄이기 위해 탐지 횟수를 높인다면 거짓 부정이 발생하여 포트스캔 공격인데도 포트스캔 공격이 아니라는 판정을 하게 될 것이다(Snort, 2010).

포트스캔 공격을 실행하는 공격자는 포트스캔 공격을 탐지하는 시스템이 포트스캔 공격에 대해 거짓 부정 판정이 되기를 바랄 것이다. 공격자는 거짓 부정으로 판정되게 하려면 탐지 횟수에 대한 임계값보다 적게 포트스캔을 실행하면 된다. 포트스캔 공격탐지 시스템은 임계값보다 낮은 포트스캔에 대해서는 일반 패킷이라고 판정하기 때문이다. 이러한 포트스캔은 스텔스 포트스캔 중 느린 포트스캔(slow port scan)이라고 부른다(Staniford, 2002).

포트스캔 공격을 탐지함에 있어서 임계값에 의해 발생하는 거짓 긍정 및 거짓 부정의 단점을 보완하고, 포트스캔 공격탐지에서 임계값을 이용한다는 점을 악용한 느린 포트스캔을 탐지하기 위해서는 많은 양의 포트스캔 공격에 대한 패킷들을 저장하고 그 저장된 패킷들을 분석해야 한다. 많은 패킷들을 계속적으로 저장한다는 것과 그 패킷들을 분석한다는 것은 시간적으로 성능적으로 큰 비용을 지불하여야 한다. 본 연구에서는 거짓 긍정 및 거짓 부정의 단점을 극복하고 느린 포트스캔을 탐지하기 위해 포트스캔 패킷들을 요약하고 그 요약된 정보들을 분석하기 위한 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 포트스캔 공격 탐지에 대한 관련 연구를 살펴보고 3장에서는 포트스캔 공격의 탐지, 포트스캔 공격탐지 요약 정보, 그리고 구성 과정을 설명한다. 4장에서는 그 요약 정보를 바탕으로 주성분분석을 수행하는 과정을 설명한다. 5장에서는 실험을 통해 Snort와 본 연구를 비교 및 분석한다(Mai, 2006). 6장에서는 결론 및 향후 연구과제에 대해 논의한다.

2. 관련 연구

그 동안 포트스캔 공격탐지에 대한 여러 가지 연구가 있어 왔다. 단순한 방법으로는 포트스캔 공격의 프로토콜의 특성이나 시간 분포 등을 분석하는 연구가 있었으며, 좀 더 복잡한 방법으로는 임계값을 설정하는 방법, 확률적으로 포트스캔을 결정하는 방법, 네트워크 트래픽을 그래프로 표현하여 포트스캔을 탐지하는 방법, 패킷의 프로토콜 분포 및 시간 분포, 그리고 주성분분석을 이용한 방법 등이 있었다(Lee, 2003).

첫 번째, 임계값에 의한 탐지 방법으로 Northcutt과 Novak 등은 포트스캔을 탐지하기 위한 표준 방법이라는 것이 주어진 시간 내에 발생한 관심 있는 사건들의 수를 세는 것이라고 언급했다(Northcutt, 2002). 예를 들어 여기에서 사건들의 수는 지난 60초 내에 같은 호스트 상에 여러 포트들을 접근한 수를 의미한다. 이 방법은 가장 보편적으로 사용되는 표준 방법이다. Heberlein 등은 이 표준 방법을 응용했다. 대부분 포트스캔 탐지 접근법들이 일반적으로 비정상 행위의 간단한 형태를 탐지하는 것을 목적으로 하듯이 그들이 개발한 NSM(network security monitor)이 룰들을 사용해서 포트스캔 공격을 탐지한 후, 포트스캔 공격들을 T초의 간격 내에 N 사건들을 공격으로 탐지했다(Mai, 2006). 그리고 오픈 소스로 유명한 Snort에서는 외부 호스트를 포트 스캐너로 탐지하기 위해 T 초에서 N 연결 시도를 검사했지만 거짓 긍정과 거짓 부정이 높은 수치를 보였다. Snort는 탐지 룰을 기반으로 다양한 공격과 스캔을 탐지할 수 있는 장점이 있는 반면 침입에 대한 로그가 IP 주소별로 관리되어 서비스 거부 공격 시 무제한의 로그 파일이 생성되는 단점이 있다(Fyodor, 1997; Mai, 2006). 또 다른 오픈 소스인 Bro에서는 실패한 연결 요청 횟수를 저장하여 포트 스캐너를 탐지한다(Paxson, 1998). Kim 등은 포트스캔 탐지 알고리즘에서 이용하는 포트스캔을 탐지하기 위한 임계값을 자동으로 계산하는 방법을 제안하였다(Kim, 2010). 전통적으로 포트스캔 공격을 탐지하기 위해 관리자가 경험적으로 고정된 임계값을 결정하는 것은 변화하는 상황에 대처할 수 없다고 지적하였다. 그들이 제안한 임계값은 임의의 이전 기간 동안 수집된 트래픽 데이터의 통계값에 의해 계산되어 실시간으로 포트스캔 공격의 탐지에 있어서 자동적으로 설정되며, 또한 상황 변화에 적응적으로 변화할 수 있다.

두 번째, Leckie와 Kotagiri 등은 실시간으로 포트스캔을 탐지하기 위해 포트스캔 횟수 및 비정상 접근 횟수에 대한 측정을 한 뒤 확률을 계산했다. 그러나 스텔스 포트스캔을 측정할 수 없다. Bro와 비슷한 방법을 사용한 TRW는 모든 연결 요청을 저장한 뒤, 포트스캔인지 아닌지를 결정하기 위해 순차 가설 검증을 사용하는 우도율을 계산했다. Staniford와 Leckie 등은 포트 스캐너의 소스 주소의 우도를 사용해서 포트스캔 공격을 탐지하는데 사용했다 (Leckie, 2002; Staniford, 1996).

세 번째, GrIDS에서는 네트워크 트래픽 정보를 일상적인 네트워크 활동 구조를 그리는 활동 그래프들(activity graphs)에 합산하고, 활동 그래프의 특성들을 분석함에 의해 포트 스캔을 탐지했다(Northcutt, 2002). Mai 등은 샘플링이 여러 트래픽 특징들을 왜곡하며 성공 탐지율과 거짓 긍정 면에서 포트스캔 공격탐지 알고리즘의 성능을 저하시킴을 증명했다(Jung, 2004).

마지막 방법으로 Kikuchi 등은 포트스캔 공격 탐지를 위해 주성분분석을 이용하였다(Kikuchi, 2009). 그들은 주성분분석을 하기 위해 호스트 및 포트 별로 포트스캔 횟수를 행렬 형태로 저장하고, 공분산을 계산 후 고유값과 고유벡터를 구하였다. 그리고 고유값을 내림차순으로 하여 고유벡터들의 행렬을 구성하였다. 주성분 기저와 호스트의 포트스캔의 관측된 횟수의 곱으로써 직교 확장을 계산한 후, 그 관측된 횟수와 직교 확장을 모두 더한 값에 호스트 별 평균을 합하여 포트스캔 횟수를 예측하는데 사용하였다. 또한 같은 방법으로 잃어버렸거나 침입차단시스템에 의해 기록되지 않은 포트스캔 공격 횟수를 알아내는데 이용하였다.

위에 소개된 관련 연구들은 약간의 차이를 보이지만 비교적 짧은 기간의 포트스캔 공격 탐지 데이터들을 대상으로 한다는 것이 공통적이다. 단기간의 포트스캔 데이터만 대상으로 한다는 것은 탐지의 정확성 면에서 한계를 보일 수밖에 없다. 그러므로 본 연구는 포트스캔 공격탐지의 정확도를 증가시키기 위해 장기간의 포트스캔 공격 데이터를 요약한 후, 그 요약 정보를 바탕으로 주성분분석에 의하여 포트스캔 공격별로 포트스캔 위험지수(port scan risk index; PSRI)를 부여하는 것을 제안한다.

3. 포트스캔 공격의 탐지

포트스캔을 행하는 목적은 크게 두 가지로 분류할 수 있다. 하나는 네트워크 자체를 공격하여 운영을 마비시키거나 또는 호스트를 DOS(Denial of Service) 공격하는 것이다. 두 번째는 포트스캔을 행하는 목적이 아닌 사전 탐색의 의미로서 사용될 수 있는데 어떤 목적 호스트에 침투할 예정이거나 아니면 취약점을 가진 호스트를 찾기 위한 방법으로서 활용될 수 있다. IDS가 포트스캔 공격을 탐지하는 과정은 네트워크 카드에 의해 무작위 모드(promiscuous mode)로 네트워크에 흘러 다니는 모든 패킷들을 모두 받아들이게 되며, 패킷을 디코딩하는 과정을 거쳐 일반 침입 탐지의 사전 단계로서 포트스캔 공격에 대한 처리를 하게 된다. 포트스캔 공격이라는 판명이 있게 되면 해시 테이블의 버킷에 연결된 슬롯에 포트스캔 정보의 검색 및 삽입 또는 수정을 통해 포트스캔 공격탐지 정보 테이블의 정보 구성이 이루어지게 된다.

3.1 포트스캔 공격탐지의 요약 정보

포트스캔 공격탐지 정보를 장시간 보관하고 빠른 정보의 삽입, 수정 및 검색이 용이하도록 해시 테이블을 사용한다. 해시 테이블의 Key는 충돌이 발생할 수 있으므로 연결리스트에 의한 체이닝(chaining) 기법을 이용한다. 해시 테이블의 Key로 Source IP(SIP)를 사용하고 그 슬롯(slot)의 내용은 Protocol Type(PType), 최초 탐지 시각(FTime), 마지막 탐지 시각(LTime), 포트스캔 탐지 시간(Active Time; AT), 탐지된 포트스캔 횟수(Port Scan Count; PC), 빈도 횟수(Frequency Count; FC), 그리고 포트스캔이 탐지되지 않은 시간(Idle Time; IT) 등으로 구성된다. 구성 정보의 설명은 Table 1과 같다. 구성 정보의 설명을 이해하기 위해서는 세션에 대한 정의가 필요하다. 포트 스캔이 시작된 후 시간 h내에 동일한 IP 주소로부터 다음 개별 포트 스캔이 계속해서 이어진다면 하나의 세션으로 간주된다. 임의의 시간 h가 지난 후 포트 스캔이 포착되었다면 다른 세션으로 간주한다. 시간 h는 보안 관리자가 시스템에 설정한 임의의 시간이며 대부분 너무 크지 않는 범위로 정해진다. 보안 관리자가 h를 결정하는 이유는 네트워크 상황 또는 현재 상황에 따라 다르기 때문이다.

Table 1의 포트스캔 공격탐지 정보는 8개로 구성되는데 이중 포트스캔 공격인지 아닌지의 여부는 핵심 정보인 FC, IT, PC, 및 AT에 의해서만 판단되고 나머지 4개는 참고 자료로 사용이 된다. 핵심 정보 4개의 수치에 따라 포트스캔 공격인지 아닌지를 판별할 수 있다. 핵심 정보 4개 중 AT와 PC는 기존에 사용하던 임계값 방식(몇 초 동안 몇 번의 포트스캔 횟수)과 비슷하지만 FC와 IT는 본 연구에서 새롭게 넣은 정보이다.

단순(normal) 포트스캔인지 아니면 교묘한(slow) 포트스캔인지는 포트스캔 실행 시간과 포트스캔 횟수인 AT와 PC의 수치로 판단하기 힘들다. 단순 포트스캔은 IDS의 감지 여부를 염려하지 않고 패킷을 보내 시스템의 정보를 얻거나 시스템의 오작동을 유도한다. IDS는 몇 초 내에 몇 번의 포트스캔이 있었는지의 임계값에 주목하여 포트스캔 공격으로 간주하기 때문에 단순 포트스캔은 탐지가 쉽다. 단순 포트스캔을 설명하기 위해 Snort의 포트스캔 공격 탐지 방법에 의해 예를 들면 PC의 수치가 1000 그리고 AT가 30인 경우 Snort의 포트스캔 룰에 의해 비정상으로 분류되어 공격으로 간주하게 된다.

Table 1. Summary information of port scan attack detection

Field Name	Description
SIP	포트스캔을 시도한 공격자 IP
PType	포트스캔 패킷의 프로토콜 형태
FTime	공격자가 처음으로 포트스캔을 실행한 시간
LTime	공격자가 마지막으로 포트스캔을 실행한 시간
FC	공격자가 포트스캔을 실행한 세션의 누적 수
IT	한 세션이 끝난 후 다음 시작 때까지의 누적 시간
PC	각 세션들에서 실행된 포트스캔 패킷의 누적 수
AT	각 세션들마다 실행된 누적 시간

AT와 PC가 임계값 방식과 다른 점은 포트스캔 공격 세션이 바뀌어도 계속적으로 값을 유지함과 동시에 새로운 값을 누적하게 되는 것이다. AT는 각 세션들마다 실행된 시간을 계속 누적한 시간이고 PC는 포트스캔을 실시한 IP가 DOS 공격이나 시스템에 침투하기 위해 시스템의 정보 수집을 목적으로 보낸 패킷에 대한 각 세션의 누적 횟수를 의미하기 때문에 Snort가 포트스캔을 탐지하는 방식과 다르다. 예를 들면 한 번의 포트스캔에 의해 AT의 수치가 300 그리고 PC가 5000인 경우, Snort는 “30초 이내에 60번의 포트스캔 횟수”라는 포트스캔 룰에 의해 이것을 비정상으로 분류하여 공격으로 간주하게 된다. 그러나 각 세션마다 AT의 수치가 30 그리고 PC가 50으로 100번 누적된 경우라면 Snort의 포트스캔 룰은 각각을 탐지하여 공격이 아니라고 판단하게 된다. 결론적으로 AT와 PC는 세션마다의 수치를 누적하기 때문에 기존 임계값 방식으로는 포트스캔 공격 탐지가 용이하지 않다.

포트 스캔을 실시한 공격자는 자신의 의도를 감추고 싶어 한다. 그러므로 공격자는 일정한 시간 내에 포트스캔 횟수를 임계값보다 적게 하여 포트스캔을 실시하고 일정시간 쉬었다가 다시 포트스캔을 재계한다면 IDS는 포트스캔 의도를 포착하지 못하게 된다. 교묘한 포트스캔은 이러한 임계값에 의해 탐지되지 않기 위해 일정 임계값 이내로 실시되므로 IDS가 탐지하지 못한다. 물론 임계값 이내로 보내서 포트스캔 공격으로 판정되지 않은 횟수를 모두 저장한다면 교묘한 포트스캔을 탐지하는 것이 어려운 것은 아니다. 문제는 IDS가 포트스캔에 대한 탐지 양이 많기 때문에 일정 주기로 탐지 정보를 삭제해야 한다는 것이다. 그리고 포트스캔 공격 탐지에 있어서 한 기간만을 대상으로 하는 것이 아니라 가능한 긴 기간 동안의 정보를 표현할 수 있는 IT와 FC가 있다. 포트스캔의 의도를 포착하지 못하도록 쉰 것에 대한 정보인 IT는 포트스캔 시작 후 다시 재계까지 포트스캔을 하지 않은 임의 시간이다. 일정한 시간 포트스캔을 계속하여 멈출 때까지를 한 세션으로 본다면 FC는 세션의 횟수를 의미한다.

각 세션마다 AT의 수치가 30 그리고 PC가 60으로 100번 누적된 경우, 공격인지 아닌지를 판단할 수 있기

위해서는 AT 및 PC와 더불어 FC와 IT가 필요하며, 일단 IT를 제외시키면 {AT, PC, FC} = {3000, 6000, 100}라고 표현할 수 있다. 이는 Snort가 포트스캔을 탐지하기 위해 사용하는 한 세션의 포트스캔 정보 {AT, PC} = {30, 60}의 100번을 단순히 저장하였다고 생각할 수 있다. Snort의 "30초 이내에 60번의 포트스캔 횟수"라는 포트스캔 룰에 의해 포트스캔이 보내질 때마다의 합인 100번을 검사한다면, 모두 포트스캔 공격이라고 판정될 것이다. 그러나 {AT, PC, FC} = {3000, 5000, 100}라고 한다면 포트스캔 공격 탐지를 우회하기 위한 의도적인 패킷들의 송신으로 생각할 수 있다. 누적의 결과인 AT의 3000과 PC의 5000만 본다면, 포트스캔 공격으로 생각할 수 있지만, Snort의 포트스캔 룰에 의해 포트스캔이 보내질 때마다의 합인 100번을 검사한다면, 모두 포트스캔 공격이 아니라고 판정될 것이다. 여기서 FC의 역할은 몇 세션내에 AT와 PC의 누적 합이 되었는지를 나타낸다.

FC와 더불어 IT의 추가도 생각할 수 있다. 포트스캔을 하는 공격자는 포트스캔 공격 탐지를 우회하는 것을 일정한 시간 간격으로 할 수 있지만 우연한 패킷의 발생으로 가장하고 싶어 할 수 있다. 그러므로 세션 간 간격을 일정하지 않게 많이 두어 더욱 교묘하게 만들 수 있다. 세션 간 일정한 간격으로 포트스캔을 실시하지 않은 누적된 시간을 IT라고 정의하고 이것을 위의 표현에 추가하면 {AT, PC, FC, IT} = {30000, 500000, 10000, 500000}라고 표현할 수 있다. IT는 10000번의 세션들 사이에 평균적으로 50초 정도를 포트스캔을 실행하지 않고 쉰 것이고 Snort의 포트스캔 룰에서 30초를 염두에 두었다고 생각할 수 있기 때문에 교묘한 포트스캔 공격으로 분류할 수 있다. 위의 설명은 IT와 FC는 교묘한 포트스캔을 탐지하는데 효과적임을 설명하였으며, {AT, PC, FC, IT}에 의해 오랜 시간 동안의 포트스캔 공격에 대한 정보를 요약하여 효율적으로 포트스캔을 탐지할 수 있다. 본 장은 긴 기간 동안의 상황을 반영하는 포트스캔 공격 정보를 요약하는 연구를 하였고, 다음 장에서는 이 요약 정보를 이용하여 포트스캔 위험 지수를 생성하는 방법에 대해 설명한다.

4. 포트스캔 공격탐지 정보의 주성분분석

포트스캔 공격탐지 요약 정보 중 AT, PC, FC, 및 IT만이 공격인지 아닌지의 판단 여부가 되기 때문에 포트스캔 공격탐지의 중요한 분석 척도가 된다. 그러나 4가지 포트스캔 공격탐지 요약 정보의 단순한 계산만으로는 공격의 위험성 정도를 정량화하기는 어렵다. 이 문제의 해결방안으로서 위험성 정도를 정량화하기 위한 4가지 정보의 집약 기법이 필요하다. 이 정보들의 특성은 횟수와 시간(초)으로 구성되어 있으며 모두 독립변수이고 종속변수는 없다. 따라서 총체적으로 위험성 정도를 평가 할 수 있도록 주성분분석을 이용하여 포트스캔의 위험지수를 생성할 수 있는 방법을 제안한다.

포트스캔 공격탐지의 요약 정보를 대상으로 하는 주성분분석을 통해 위험성 정도의 평가를 실시한다. 포트스캔 공격탐지 요약 정보에 대한 주성분분석의 흐름은 첫 번째 포트스캔 공격탐지 요약 정보를 가지고 고유값(eigenvalue)과 고유벡터(eigenvector)를 구하고, 두 번째는 고유값과 고유벡터를 선택하기 위해 기여율을 계산하며, 세 번째는 앞서 계산한 기여율에 의해 선택된 고유벡터로써 위험성 정도를 정량화한 위험지수를 계산하는 것이다.

4.1 포트스캔 공격탐지 정보의 주성분분석

주성분분석을 하게 될 포트스캔 공격탐지 요약 정보는 AT의 벡터 ($a_1, a_2, \cdots, a_n$), PC의 벡터 ($b_1, b_2, \cdots, b_n$), FC의 벡터 ($c_1, c_2, \cdots, c_n$), IT의 벡터 ($d_1, d_2, \cdots, d_n$) 등으로 이루어진다. 포트스캔 공격탐지 요약 정보의 주성분분석을 하기 위해서는 5단계를 거치게 된다.

단계 1은 포트스캔 공격탐지 요약 정보의 각 항목을 정규화하는 것이다. 포트스캔 공격탐지 요약 정보의 각 항목은 크기 분포에 있어 상당한 차이가 날 수 있기 때문에 평균에서 떨어진 정도라든지 데이터의 분포된 정도를 바탕으로 정규화하게 된다. 예를 들어 집합 $x = \{x_1, x_2, \cdots, x_n\}$이 있다고 할 때, x를 평균, σ_x를 표준편차라고 하면 표준값은 $(x_i - x)/\sigma_x$를 사용하여 변환한다. a와 σ_a는 AT의 벡터 ($a_1, a_2, \cdots, a_n$), b와 σ_b는 PC의 벡터 ($b_1, b_2, \cdots, b_n$), c와 σ_c는 FC의 벡터 ($c_1, c_2, \cdots, c_n$), 그리고 d와 σ_d는 IT의 벡터 ($d_1, d_2, \cdots, d_n$)의 각각 평균과 표준편차이다.

단계 2는 포트스캔 공격탐지 요약 정보의 정규화된 값들의 상관행렬을 구하는 것이다. 집합 $x = \{x_1, x_2, \cdots, x_n\}$의 표준편차 σ_x, 집합 $y = \{y_1, y_2, \cdots, y_n\}$의 표준편차 σ_y, 집합 x와 y의 공분산 σ_{xy}라고 하면 상관계수 r은 $\sigma_{xy}/\sigma_x \cdot \sigma_y$에 의해 구해진다. 이것을 이용하여 AT의 벡터 ($a_1, a_2, \cdots, a_n$), PC의 벡터 ($b_1, b_2, \cdots, b_n$), FC의 벡터 ($c_1, c_2, \cdots, c_n$), IT의 벡터 ($d_1, d_2, \cdots, d_n$)의 각각 상호 상관관계를 Table 2와 같이 계산하면 포트스캔 공격탐지 요약 정보의 상관행렬이 구해진다.

Table 2. Correlation matrix about normalized summary information of port scan attack detection

	a of AT	b of PC	c of FC	d of IT
a of AT	1	$\sigma_{ab}/(\sigma_a \cdot \sigma_b)$	$\sigma_{ac}/(\sigma_a \cdot \sigma_c)$	$\sigma_{ad}/(\sigma_a \cdot \sigma_d)$
b of PC	$\sigma_{ab}/(\sigma_a \cdot \sigma_b)$	1	$\sigma_{bc}/(\sigma_b \cdot \sigma_c)$	$\sigma_{bd}/(\sigma_b \cdot \sigma_d)$
c of FC	$\sigma_{ac}/(\sigma_a \cdot \sigma_c)$	$\sigma_{bc}/(\sigma_b \cdot \sigma_c)$	1	$\sigma_{cd}/(\sigma_c \cdot \sigma_d)$
d of IT	$\sigma_{ad}/(\sigma_a \cdot \sigma_d)$	$\sigma_{bd}/(\sigma_b \cdot \sigma_d)$	$\sigma_{cd}/(\sigma_c \cdot \sigma_d)$	1

단계 3은 상관행렬의 행렬식을 통해 포트스캔 공격 탐지 요약 정보에 대한 고유값 λ와 고유벡터를 얻는 것이다. 단계 2에서 구해진 상관행렬은 4x4인 정방행렬이므로 이것으로부터 고유값을 얻기 위해서는 특성방정식을 이용할 수 없으므로 여인수(cofactor) 전개를 통해 구할 수 있다. 포트스캔 공격 탐지 요약 정보의 상관행렬이 4x4인 정방행렬이기 때문에 주성분분석에서 4개의 고유값이 존재하게 된다. 4개의 고유값 λ를 λ_1, λ_2, λ_3, λ_4라고 하면 이 값에 의해서 고유벡터를 구할 수 있다. λ_1의 고유벡터 (x_1, x_2, x_3, x_4), λ_2의 고유벡터 (x_5, x_6, x_7, x_8), λ_3의 고유벡터 (x_9, x_{10}, x_{11}, x_{12}), 그리고 λ_4의 고유벡터 (x_{13}, x_{14}, x_{15}, x_{16}) 등을 각각 얻을 수 있다.

단계 4는 포트스캔 공격탐지 요약 정보에 대한 상관행렬을 얻은 후, 행렬식의 가장 큰 고유값에 대응하는 고유벡터에 의한 요인 패턴(factor pattern)을 구하는 것이다. 고유값 중 $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$일 경우 가장 상위 값을 선택하게 된다. 따라서 가장 큰 고유값 λ_1에 대응하는 고유벡터는 (x_1, x_2, x_3 x_4)이 된다.

단계 5는 단계 4에서 구한 고유값 λ_1의 고유벡터를 이용하여 포트스캔 공격에 대한 위험성의 정도를 표시하는 위험 지수 방정식을 구하는 것이다.

$$PSRI = x_1u_1 + x_2u_2 + x_3u_3 + x_4u_4 \quad (1)$$

여기서 집합 u={u_1, u_2, u_3, u_4}는 포트스캔 공격탐지 요약 정보의 정규화된 값들로서, u_1은 AT의 값, u_2는 PC의 값, u_3은 FC의 값, 그리고 u_4는 IT의 값을 의미한다. PSRI 방정식에 곱해진 고유벡터의 성분들 중 x_1은 AT에, x_2는 PC에, x_3은 AT에, 그리고 x_4는 IT에 곱하여 본 연구에서 목표로 하는 포트스캔 공격에 대한 위험지수를 얻을 수 있다. 이 위험지수를 사용하여 심각한 포트스캔 공격인지 아닌지를 판별하는 기준으로 사용할 수 있으며, 위험지수가 일정 임계값 이상이면 심각한 포트스캔 공격으로 판정하게 된다.

4.2 포트스캔 공격탐지 정보의 주성분분석에 대한 고유값의 누적 기여율

포트스캔 공격탐지 요약 정보의 주성분분석에서 각 주성분은 서열이 존재하며 전체 주성분에 기여하는 정도를 계산할 수 있다. 고유값의 가장 큰 값이 제1주성분(first principal component, 1st PC)이 되며, 그 다음 큰 값이 제2주성분(second principal component, 2nd PC), 이와 같은 방법으로 제3~4주성분(3th~4th principal component, 3th~4th PC)이 결정된다. 각 주성분들의 기여율은 자신의 고유값 나누기 전체 고유값의 합으로 나타낸다. 제i주성분의 기여율 C_i는 다음과 같이 계산한다. 여기서 n은 고유값의 개수이다.

$$C_i = \frac{\lambda_i}{n} \times 100 \quad (2)$$

몇 개의 주성분을 선택할지는 누적기여율(Accumulation Contribution Rate)의 계산된 값에 의해 판단된다. 누적기여율은 차례대로 제1주성분에서 마지막인 제4주성분까지를 순서대로 더한 값이다. 포트스캔 공격탐지 요약 정보에 대한 4가지 주성분의 기여율이 있다고 할 때, 각각의 주성분의 기여율은 분석대상의 데이터가 가지고 있던 정보가 그 주성분에 어느 정도 집약되어 있고 설명력을 가지는 지를 나타낸다. 또한 누적기여율은 고유값의 기여율을 누적할 때 전체 누적에 참여한 주성분에 의해 포트스캔 공격탐지 요약 정보가 어느 정도 설명력을 가질 수 있는 지를 나타낼 수 있다. 본 논문에서는 누적기여율이 80%이상 되는 최초의 주성분까지를 채택하였다.

5. 실험과 평가

5.1 실험 방법

Snort는 공격자가 포트스캔을 행한 포트들의 수와 시간에 대해 이미 설정된 진행 임계 포트들의 수 N과 진행 임계 시간 T를 비교하여 포트스캔 공격의 유무를 판단한다. Snort의 T와 N은 개별 네트워크 마다 이미 정해진 고정된 값들이 아니다. 보안관리자가 T의 값을 높이고 N의 값을 낮춘다면 세세한 포트스캔까지도 탐지할 수 있지만 정상적인 패킷들도 포트스캔이라고 판단하는 거짓 긍정이 발생할 수 있는 가능성이 높다. 반대로 T의 값을 낮추고 N의 값을 높인다면 거짓 부정이 발생할 수 있다. 그러므로 T와 N 값들은 개별 네트워크에서 보안을 책임지는 보안관리자의 경험적인 면이나 개별 네트워크의 상황을

고려하여 설정되어져야 한다. Snort는 포트스캔 공격을 탐지하기 위한 기본 임계값을 T=3과 N=4로 설정하고 있다. 이 임계값들은 포트스캔 공격을 잘 탐지하기도 하지만 공격이 아닌 일반 패킷들도 공격으로 인식하는 문제를 발생시키기 때문에 관리자들은 이 값들을 수정해야 한다. 본 시뮬레이션에서는 Snort의 포트스캔 탐지를 위한 임계값을 여러 개 설정하여 결과가 더 좋은 것을 선택하였다.

본 연구는 제안한 방법이 Snort의 포트스캔 탐지 방법보다 포트스캔을 더 효과적으로 탐지한다는 것을 보이기 위해 시뮬레이션을 수행하였다. 시뮬레이션에서 이용한 포트 스캔 공격은 CON, SYN, FIN, XMAS, NULL, ACK, WIN, RPC, 그리고 UDP 등 9개를 대상으로 한다. 그리고 포트스캔으로 오인하기 쉬운 P2P(Peer to Peer) 및 DNS 응용 프로그램의 패킷들을 시뮬레이션에서 포트스캔 공격과 함께 혼합하여 포트스캔 탐지 효율성을 측정하였다. 시뮬레이션에서 포트스캔을 탐지했을 때, 기본적인 정보로는 소스 IP, 포트스캔을 행한 시간, 포트스캔을 행한 횟수, 그리고 포트스캔 공격 종류이고, 이들을 각각 확률적으로 발생시켜 구성하였다.

본 시뮬레이션에서 Snort의 포트스캔 탐지 방법은 포트스캔 공격 시그너처를 가진 패킷이 적용된 시간 임계값(Time Threshold; TT)과 계수 임계값(Count Threshold; CT) 이상 발생하면 포트스캔 공격이라는 가정 하에 포트스캔 공격 정보를 저장한다. 하지만 본 논문에서 제안한 PSRI의 포트스캔 공격탐지 방법을 시행하기 위해 포트스캔 시그너처를 가진 패킷이면 임계값에 관계없이 포트스캔 공격탐지 요약 정보의 구성 요소들을 새로이 삽입하거나 갱신한다. 포트스캔 공격탐지 요약 정보는 Table 1에서 보는 바와 같이 FC, IT, PC, 그리고 AT 등과 소스 IP, 마지막으로 포트스캔을 행한 시간, 그리고 포트스캔 공격 정보들로 구성된다. 본 논문에서 제안한 방법은 이 포트스캔 탐지 요약 정보를 이용하여 포트스캔 공격의 위험지수를 알아내게 된다. 본 시뮬레이션에서는 Snort의 포트스캔 탐지를 위한 임계값을 여러 개 설정하여 결과가 더 우수한 것을 고른 것처럼 제안한 PSRI의 임계값도 여러 개 설정하여 결과가 더 좋은 것을 선택하였다.

본 연구의 성과를 측정하기 위해 포트스캔 탐지로부터 참 긍정(TP; True Positive), 거짓 긍정(FP; False Positive), 거짓 부정(FN; False Negative)에 의해 계산하고, 식 (3), (4), (5)처럼 accuracy(A), precision(P), 그리고 recall(R)을 계산하였다. 또한 단일한 측정치를 위하여 P와 R을 하나로 합한 F-measure(F)를 식 (6)와 같이 구하였으며, P와 R사이의 중요도를 균등하게 생각하여 $\beta = 1$로 설정하였다. 일반적으로 성능을 측정하는 데에는 P, R, 그리고 F를 사용하면 크게 문제가 되지 않는다.

$$A = \frac{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} TN_i}{\sum_{i=1}^{n} TN_i + \sum_{i=1}^{n} FP_i + \sum_{i=1}^{n} FN_i + \sum_{i=1}^{n} TN_i} \quad (3)$$

$$P = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \quad (4)$$

$$R = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \quad (5)$$

$$F = \frac{(1+\beta^2) * R * P}{\beta^2 * (R+P)} \quad (6)$$

또한 명확한 검증을 위해 수신자 동작 특성(Receiver Operating Characteristic, ROC)을 사용한다. ROC는 세로축에 참 긍정률(True Positive Rate; TPR)을 놓고, 가로축에 거짓 긍정률(False Positive Rate; FPR)을 위치하게 그래픽적으로 모델들의 상대적 성능을 비교하는 데 사용된다. TPR은 R과 같고 FPR은 식 (7)과 같다. 가장 이상적인 것은 TPR=1이고 FPR=0일 때이며, 가장 부정적인 때는 TPR=0이고 FPR=0일 때이다(Provost, 1997).

$$FPR = \frac{\sum_{i=1}^{n} FP_i}{\sum_{i=1}^{n} TN_i + \sum_{i=1}^{n} FP_i} \quad (7)$$

5.2 포트스캔 공격탐지 정보의 실험 결과

포트스캔 공격탐지 정보를 바탕으로 주성분분석을 실시하여 누적기여율에 의해 몇 개의 주성분들을 선택할지 결정하였다. 주성분의 기여율은 그 주성분이 전체 주성분들에서 얼마만큼의 설명력을 가지는지를 나타내는 척도가 된다. 가장 큰 주성분의 기여율로부터 하나씩 차례로 다음으로 큰 주성분 기여율을 더해 나가면 누적 기여율에 대한 계산을 마칠 수 있다. 일반적으로 몇 퍼센트에서 주성분들을 선택해야 하는지에 대한 명확한 척도는 존재하지 않지만 80% 정도면 기여율이 높다고 결정하게 된다. Table 3은 실험에서 나타난 포트스캔 공격

탐지 정보의 주성분 누적기여율이다.

실험에서 제1주성분이 차지하는 기여율이 88.30%로 높은 수치를 보여 상대적으로 다른 주성분들은 기여율이 아주 낮게 나왔다. 그러므로 제1주성분만 선택하면 된다. 본 논문에서는 선택된 제1주성분에 해당하는 고유벡터를 가지고 포트스캔 공격에 대한 위험정도를 나타내는 위험지수를 계산하였다. Table 4는 포트스캔 공격탐지 정보의 위험지수 PSRI 계산 결과를 보여준다. 위험지수에 의해 포트스캔 공격을 판단하는 기준이 마련되지 않았기 때문에 5.1절에서 설명하였듯이 PSRI의 임계값을 여러 개로 변경하면서 결과가 더 좋은 임계값을 선택하여 포트스캔 공격을 탐지한다. Fig. 1은 제안한 알고리즘에 의한 포트스캔 공격의 Risk Index의 분포를 보여주고 있다. 포트스캔 공격의 Risk Index의 분포에 대해 1보다 작은 값에 대해 좀 더 세밀하게 표현하고자 Risk Index 값이 1보다 클 경우 1로 한정하여 보여주고 있다.

Table 3. Accumulated contribution rate of principal component about the information of port scan attack detection

	고유값	기여율(%)	누적기여율(%)
1st PC	3.53	88.30	88.30
2nd PC	0.34	8.64	99.94
3rd PC	0.12	2.99	99.93
4th PC	0	0.06	99.99

Table 4. Risk index about the information of port scan attack detection

Seq.	FC	IT	PC	AT	PSRI
1	229	7591	69199	9446	24.03
2	220	7140	65784	7221	21.56
3	2	5405	693	10	3.45
4	28	1105	6759	635	2.25
5	2	3371	507	0	2.06
6	15	725	4834	1325	1.97
7	2	950	1120	72	0.53
8	2	931	417	0	0.40
9	2	711	886	0	0.30
10	3	594	1215	2	0.28
11	3	586	1019	0	0.25
12	1	0	668	645	0.24
13	2	384	1116	115	0.18
14	1	0	655	416	0.07
15	2	165	294	169	0.00
16	2	30	645	242	-0.01
17	2	34	1200	137	-0.03
18	2	279	432	0	-0.04
19	3	60	498	110	-0.07
20	1	0	470	224	-0.08
21	2	180	695	0	-0.08
22	2	32	451	156	-0.09
23	1	0	255	200	-0.12
24	1	0	12	0	-0.29
25	1	0	1	0	-0.29

임계값을 0.0으로 하였을 때, Table 4에서 20번의 위험지수는 -0.08로 일반 포트 스캔에 해당한다. 그 이유는 포트스캔 세션의 빈도인 FC가 1이고 세션 사이에 포트스캔을 보내지 않은 시간 IT가 0초, 한 세션에서 포트에 대한 정보를 알아내기 위해 보낸 포트스캔 횟수 PC가 470, 그리고 한 세션에서 AT에 해당하는 포트스캔을 보낸 시간이 224초로 1초당 2개 정도를 보냈기에 위험성은 낮기 때문이다.

1번을 살펴보면 전체 포트스캔 횟수 69199를 보내기 위해 229번으로 나누어 보내고, 세션 사이에 쉰 누적된 시간이 7591초 된 것을 알 수 있다. 이를 보낼 때 소비된 누적된 시간도 9446초로 아주 느리게 전송되었다. 1번에 대해 세션 단위로 분석해 보면 하나의 세션 당 소비시간은 41초이고 전송한 패킷은 302개이며, 그리고 세션 사이에 쉰 평균 시간은 33초이다. 이 세션 단위의 결과는 Snort의 포트스캔 탐지 알고리즘을 사용해서 공격으로 판별될 경우 아주 치명적이지는 않다. 왜냐하면 Snort는 누적된 포트스캔 공격 탐지 정보를 이용하지 않고 단지 현재 시점에서 임계값을 기준으로 공격 정도를 판정하고 다음 시점에서는 모든 결과를 버리기 때문이다. Snort에 의해 공격 탐지를 할 경우 1번보다 20번이 더 치명적이라고

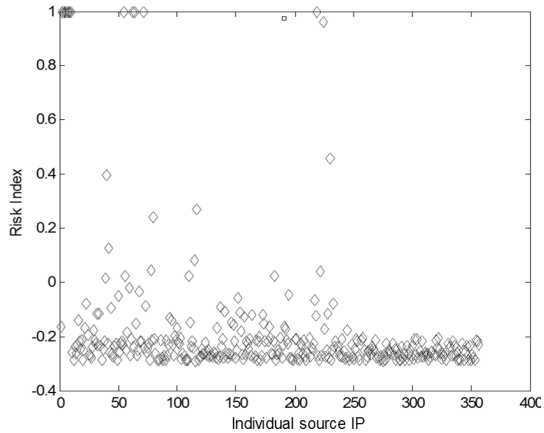

Fig. 1. Risk index distribution of port scan attack by proposed algorithm

판별하게 된다. 그러나 본 연구에서 제안한 방법으로 누적된 포트스캔 공격 탐지 정보 분석을 할 경우 훨씬 더 위험한 공격자로 판별하게 된다. 따라서 20번에 제시된 -0.08보다 1번의 위험지수가 24.03만큼 높게 제시되었다.

포트스캔 공격탐지 요약 정보들 중 3번은 FC=2, IT=5404, PC=693, AT=10이고 4번은 FC=28, IT=1105, PC=6759, AT=0이다. 3번의 IT=5405는 4번의 IT=1105보다 크지만 3번의 나머지는 FC만 조금 상대적으로 조금 차이가 날 뿐 다른 두 값 PC 그리고 AT는 둘 다 확연하게 작은 값을 가진다. 그러나 포트스캔 공격탐지 정보에 대한 위험지수 살펴보면 4번보다 3번의 IT값 하나만 큰 값을 갖더라도 3번은 PSRI=3.45이여서 4번의 PSRI=2.25보다 큰 값을 가짐을 볼 수 있다. 이것은 포트스캔 공격탐지 정보에 대한 위험지수를 위해 PC와 AT에 있어서 더 큰 차이가 나는 값을 갖는 것보다 IT의 값에 대해 더 큰 비중을 가진다는 것을 알 수 있다.

Snort의 포트스캔 공격탐지 방법과 PSRI의 포트스캔 공격탐지 방법을 비교하기 전, Snort의 포트스캔 공격탐지 방법에서 적절한 TT와 CT를 실험을 통해 결정하였다. 시간 임계값과 계수 임계값의 쌍을 (60, 90), (60, 150), (60, 180)로 하여 하나씩 차례대로 Snort의 포트스캔 공격탐지에 적용한 뒤 precision, recall, 그리고 F-measure 등을 측정하였다. precision는 TT=60와 CT=150가 일 때 가장 높았고, TT=60와 CT=90가 일 때 가장 낮았다. recall은 TT=60와 CT=90가 일 때 가장 높았고, TT=60와 CT=180가 일 때 가장 낮았다. TT=60와 CT=90일 때 거짓 긍정의 영향으로 precision가 가장 낮았지만, 거짓 부정이 줄어들어 recall이 가장 높게 나왔고, 결과적으로 F-measure가 가장 높았다.

PSRI를 이용한 포트스캔 공격탐지 방법에서 적절한 PSRI 임계값은 실험을 통해 결정하였다. 임계값을 -0.15, 0, 그리고, 1로 하여 하나씩 차례대로 PSRI의 포트스캔 공격탐지 방법에 적용한 뒤 precision, recall, 그리고 F-measure 등을 측정하였다. precision는 3개 모두 비슷한 결과를 보였고. recall은 거짓 부정율이 줄어든 PSRI 임계값이 -0.15일 때 가장 높았다. PSRI 임계값이 -0.15일 때 precision는 다른 임계값과 비슷했지만 recall에서 상대적으로 상당히 높게 나왔기 때문에 F-measure가 가장 높을 수 있었다. PSRI에 의한 포트스캔 공격탐지와 Snort의 포트스캔 공격탐지를 비교하기 위해 두 개에서 각각 결과가 가장 잘 나온 임계값 -0.15와 (60, 90)을 사용하였다.

포트스캔 공격탐지에 대한 정확도를 알기위해 Snort의 포트스캔 공격탐지 방법과 본 연구의 제안방법을 비교한 것이 Fig. 2이다. Fig. 2를 살펴보면 평균적으로 본 연구에서 목표로 했던 포트스캔 공격 탐지의 accuracy는 Snort보다 훨씬 높고 안정적으로 나왔다. precision을 나타낸 Fig. 3은 본 연구에서 제안한 알고리즘이 Snort보다 높게 나왔지만 약간은 불안정적인 면을 보였다. 그러나 Fig. 4를 보면 본 연구에서 제안한 알고리즘보다 Snort의 recall이 더 높게 나온 것을 알 수 있다. 마지막으로 Fig. 3의 precision과 Fig. 4의 recall에 대한 조화평균을 표현한 Fig. 5의 F-measure를 살펴보면 본 연구에서 제안한 알고리즘이 평균적으로 더 높음을 알 수 있다.

지금까지 본 연구에서 제안한 알고리즘과 Snort의 포트스캔 공격탐지에 대한 내용에 대해 TP, FN, FP, 그리고 TN 등에 의해 표현한 혼동 행렬(confusion matrix)을 가

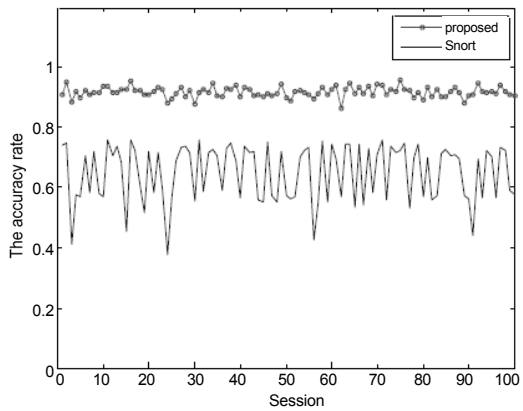

Fig. 2. Accuracy about proposed algorithm and Snort's port scan attack detection

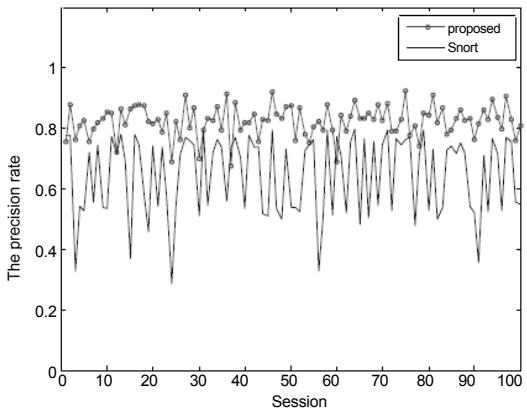

Fig. 3. Precision about proposed algorithm and Snort's port scan attack detection

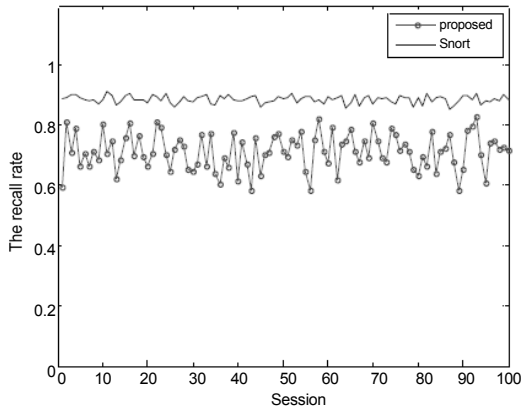

Fig. 4. Recall about proposed algorithm and Snort's port scan attack detection

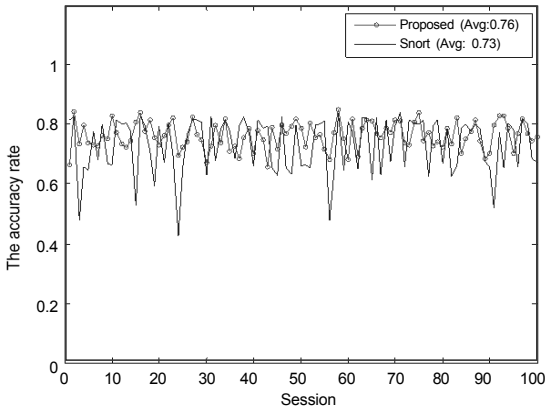

Fig. 5. F-measure about proposed algorithm and Snort's port scan attack detection

지고 precision, recall, 그리고 F-measure 등에 의해 비교 분석하였다. 결과적으로 본 연구에서 제안한 알고리즘이 더 우세하다고 할 수 있지만 극명한 결과를 주지는 못하였다. 둘 가운데 더 우수한 모델을 선택하기 위해 수신자 운영 특성(receiver operating characteristic, ROC) 곡선에 의한 비교 결과를 보여주는 것이 Fig. 6이다. ROC는 혼동 행렬보다 훨씬 더 직감적이고 견고한 형태를 제공하면서 시각적으로 같은 정보들을 표현하므로 모델 선택에 있어서 더 유리하다고 볼 수 있다. 또한 ROC는 모델을 비교하는 데 있어서 데이터에서 왜곡된 특성에 영향을 받지 않는다. Fig. 6은 본 연구에서 제안한 알고리즘과 Snort의 혼동 행렬로부터 거짓 긍정률과 참 긍정률을 계산하여 ROC 곡선을 그린 것이다. Fig. 6에서 보면 본 연구에서 제안한 알고리즘의 곡선 아래로 Snort의 곡선이

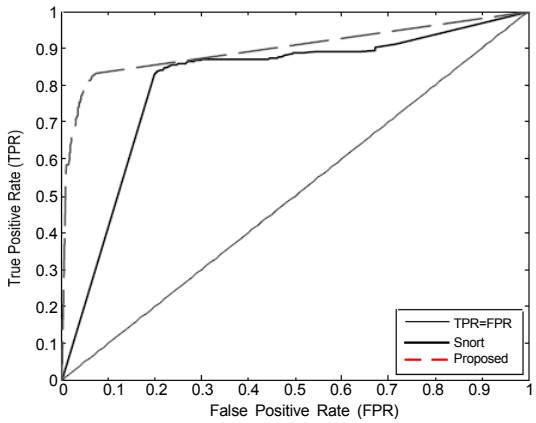

Fig. 6. ROC curve of proposed algorithm and Snort's port scan attack detection

포함된 것을 볼 수 있다. 이것은 Snort의 포트스캔 공격탐지 알고리즘보다 본 연구에서 제안한 알고리즘이 더 효과적이라는 것을 보인다.

6. 결론 및 향후 연구과제

본 연구는 포트스캔 공격탐지의 정확도를 증가시키기 위한 방법으로 포트스캔 공격탐지 정보를 주성분분석에 의하여 포트스캔 위험지수(PSRI)로 나타내는 것을 제안하였다. 포트스캔 공격탐지 정보는 포트스캔 공격탐지 시간(Active Time; AT), 포트스캔 횟수(Port Scan Count; PC), 포트스캔 빈도 횟수(Frequency Count; FC), 그리고 포트스캔 사이에 쉬는 시간(Idle Time; IT) 등으로 구성되며, 주성분분석을 이용해서 포트스캔 공격탐지 정보의 포트스캔 위험지수(port scan risk index)를 계산하였다.

본 실험 결과에서 제안한 포트스캔 공격 탐지의 accuracy는 Snort보다 훨씬 높게 나왔다. 그 밖에 accuracy 관련된 recall, precision, 및 F-measure의 결과는 Snort의 precision 및 F-measure보다 높음을 보였지만 거짓 부정에 관련된 recall에 있어서 Snort의 recall보다 결과 값이 좋지 못했으나, ROC 곡선을 이용하여 본 연구에서 제안한 알고리즘과 Snort를 비교한 결과 본 연구에서 제안한 알고리즘이 우수한 결과를 얻었다.

포트스캔 공격의 분석에 대한 향후 과제로는 위험지수 방정식을 기반으로 위험지수가 계산된 후 포트스캔 공격을 판별할 임계값을 얼마로 해야 할지를 결정하는데 있어서 네트워크의 상황에 따라 위험지수 임계값은 달라질 수 있기 때문에 최적화된 임계값에 대한 연구 및 동적인 임

계값 조정 방안에 대한 연구가 필요하며, 또한 거짓 부정에 관련된 재현율을 높이는 방안에 대한 연구도 필요하다.

또한 PSRI는 시간이 지남에 따라 변형될 수 있기 때문에 주기적인 갱신이 필요하다. 그러나 이러한 주기적 갱신은 현재 네트워크 상황의 패킷들에 시스템이 빠르게 적응하기 어렵게 만든다. 이러한 문제를 해결하기 위해 변화하는 상황에 따라 점진적으로 PSRI를 구하는 방법에 대한 연구가 필요하다.

참 고 문 헌

1. W. El-Hajj, F. Aloul, Z. Trabelsi, N. Zaki, “On Detecting Port Scanning using Fuzzy Based Intrusion Detection System”, *Coll. of Inf. Technol.*, UAE Univ., Al-Ain, 2008.
2. Fyodor, The Art of Port Scaning, *Phrack Magazine*, Vol. 7, Issue 52, 1997.
3. IANA, *http://www.iana.org/assignments/port-numbers*, 2010
4. H. Kikuchi, T. Kobori, “Orthogonal Expansion of Port-scanning Packets”, *Intl. Conf. on Network-Based Information Systems*, 2009.
5. S. K. Kim, S. H. Lee and S. W. Seo, “An Automatic Portscan Detection System with Adaptive Threshold Setting”, *Journal of Communications and Networks*, Vol. 12, No. 1, Feb. 2010.
6. J. Jung, V. Paxson, A. W. Berger, H. Balakrishnan, “Fast Portscan Detection Using Sequential Hypothesis Testing”, *Proc. of the 2004 IEEE Symposium on Security and Privacy*, 2004.
7. A. Lazarevic, V. Kummar, J. Srivastava, *Managing Cyber Threats : Issues, Approaches and challenges*, Springer, 2005
8. C. Leckie, R. Kotagiri, “A probabilistic approach to detecting network scans”, *Proc. of the Eighth IEEE Network Operations and Management Symposium (NOMS 2002)*, Florence, Italy, 2002.
9. C. B. Lee, C. Roedel, E. Silenok, “Detection and Characterization of Port Scan Attacks”, http://cseweb.ucsd.edu/users/clbailey/PortScans.pdf, Univeristy of California, Department of Computer Science & Engineering, San Diego, 2003.
10. G. Lyon, The Art of Port Scaning, *Phrack Magazine*, Vol. 7, Issue 52, 1997.
11. J. Mai, A. Sridharan, C. N. Chuah, H. Zang, T. Ye, “Impact of packet sampling on portscan detection”, *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 12, 2006.
12. S. Northcutt, J. Novak, *Network intrusion detection an analyst’s handbook, 2nd Edition*, New Riders, 2002.
13. V. Paxson, “Bro: A System for Detecting Network Intruders in Real-Time”, *Proc. of the 7th USENIX Security Symposium*, San Antonio, TX, pp. 2435-2463, 1999.
14. F. J. Provost, T. Fawcett, “Analysis and Visualizatio of Classifier Performance: Comparison under Imprecise Class and Cost Distributions”, *Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, Newport Beach, CA, pp. 43-48, 1997 .
15. Snort, *Intrusion Detection System*, http://www.snort.org, 2010.
16. S. Staniford, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagl, K. Levitt, C. Wee, R. Yip, and D. Zerkle, “GrIDS - a graph based intrusion detection system for large networks”, *Proc. of the 19th National Information Systems Security Conference (NISS ’96)*, 1996.
17. S. Staniford, J. A. Hoagland, J. M. Mcalerney, “Practical automated detection of stealthy portscans”, *Journal of Computer Security*, 2002.

박 성 철 (scpark@dongguk.edu)

1997 동국대학교 정보관리학 학사
2000～2001 윈스테크넷 부설보안연구소 선임연구원
2001～2003 넷시큐어테크놀러지 보안연구소 선임연구원
2005 고려대학교 전자컴퓨터공학 석사
2003～2007 서울호서전문학교 인터넷정보보안과 전임교수
2012 동국대학교 컴퓨터공학 박사
2012 SK인포섹 관제사업본부/SK보안관제팀 부장

관심분야 : 기계학습, 데이터마이닝, 침입탐지, 네트워크 보안, 보안 아키텍쳐

김 준 태 (jkim@dongguk.edu)

1986 서울대학교 제어계측공학과 학사
1990 미국 University of Southern California 전기공학 석사
1993 미국 University of Southern California 컴퓨터공학 박사
1994 미국 Southern Methodist University Postdoctoral Research Associate
1995～현재 동국대학교 컴퓨터공학과 교수
2003～2004 미국 Oregon State University 방문교수

관심분야 : 인공지능, 기계학습, 데이터마이닝, 추천시스템, 소셜네트워크 분석