연관성을 이용한 침입탐지 정보 분석 시스템의 설계 및 구현

(Design and Implementation of Alert Analysis System using Correlation)

이 수 진 † 정 병 천 † 김 희 열 † 이 윤 호 †
(Soojin Lee) (Byungchun Chung) (Heeyoul Kim) (Younho Lee)

윤 현 수 †† 김 도 환 ††† 이 은 영 †††† 박 응 기 †††
(Hyunsoo Yoon) (Dohwan Kim) (Eunyoung Lee) (Eungki Park)

요 약 조직이 운영하는 네트워크의 규모가 방대해지고, 인터넷 사용이 활성화되면서 보안의 중요성도 함께 증가하였다. 그러나 최근 보안의 핵심으로 부각되고 있는 침입탐지 시스템들은 인터넷상의 공격들에 대한 적절한 분석이나 효율적인 대응책을 제공해 주기 보다는, 대량의 침입탐지 정보를 생성시켜 관리자의 부담을 가중시키고 있다. 본 논문에서는 침입탐지 시스템이 생성하는 대량의 침입탐지 정보들간에 존재하는 연관성을 분석하여 대응에 필요한 고 수준의 정보를 실시간으로 생성해 냄으로써 관리 및 분석의 효율성을 증진시키고, 나아가서는 분산 서비스 거부 공격(DDoS) 같은 대규모의 공격까지도 조기에 탐지해 낼 수 있는 능력을 갖춘 침입탐지 정보 분석 시스템을 제안한다. 그리고 제안된 시스템의 성능 분석을 위해 각 모듈의 처리 효율을 측정하고 알려진 공격 시나리오 기반의 시험 평가를 실시한다.

키워드 : 보안, 침입탐지 시스템, 연관성 분석

Abstract *With the growing deployment of network and internet, the importance of security is also increased. But, recent intrusion detection systems which have an important position in security countermeasure can't provide proper analysis and effective defence mechanism. Instead, they have overwhelmed human operator by large volume of intrusion detection alerts. In this paper, we propose an efficient alert analysis system that can produce high level information by analyzing and processing the large volume of alerts and can detect large-scale attacks such as DDoS in early stage. And we have measured processing rate of each elementary module and carried out a scenario-based test in order to analyzing efficiency of our proposed system.*

Key words : Security, Intrusion Detection System, Alert Correlation

· 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았고 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음
† 비 회 원 : 한국과학기술원 전자전산학과
sjlee@calab.kaist.ac.kr
bcchung@calab.kaist.ac.kr
hykim@calab.kaist.ac.kr
yhlee@calab.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학과 교수
hyoon@cs.kaist.ac.kr
††† 비 회 원 : 국가보안기술연구소 연구원
dkim@etri.re.kr
ekpark@etri.re.kr
†††† 정 회 원 : 국가보안기술연구소 연구원
eylee@etri.re.kr
논문접수 : 2003년 11월 20일
심사완료 : 2004년 6월 1일

1. 서 론

최근 몇 년 동안 침입탐지 시스템은 정보자산을 안전하게 보호하고 정보시스템의 보안성을 강화시키기 위한 핵심요소로 인식되면서, 네트워크와 정보시스템을 보유한 대부분의 조직들이 침입탐지 시스템을 도입하여 운영하고 있다. 그러나 광범위하게 사용되고 있는 침입탐지 시스템들은 다음과 같은 몇 가지 문제점으로 인해, 점점 더 정밀화, 분산화, 대규모화 되어 가고 있는 인터넷상의 공격들에 대한 적절한 분석이나 효율적인 대응책을 제공해 주지 못하고 있다.

첫째, 네트워크의 규모가 방대해지고 인터넷이 활성화되면서 트래픽량도 증가하였고 상대적으로 침입탐지 시

스템이 생성해내는 정보도 기하급수적으로 증가하였다. 그러나 대부분의 침입탐지 시스템들은 그러한 대량의 탐지 정보들을 가공이나 분석 없이 있는 그대로 관리자에게 전달함으로써 관리자의 부담만 가중시킨다.

둘째, 네트워크의 현재 상황이나 공격의 변화 추세에 대한 판단은 전적으로 관리자의 수작업에 의존할 수밖에 없다.

셋째, 현재의 공격들은 그 기법이 점점 다양해지면서, 하나의 공격이 서로 연관된 다수의 침입정보를 발생시킨다. 그러나 대부분의 침입탐지 시스템들은 그러한 연관된 침입탐지 정보들을 분석하여 연관성을 가진 하나의 공격이 발생한 것으로 판단하는 것이 아니라, 다수의 공격이 개별적으로 발생하였다고 판단한다. 때문에 DDoS(분산 서비스 거부 공격)나 Worm 같은 대형화된 공격의 조기 탐지는 불가능하다.

이러한 이유들로 인해 침입탐지 시스템이 생성해 내는 다수의 침입탐지 정보들 간에 존재하는 연관성들을 적절한 방법으로 분석하고 가공하여 고수준의 새로운 정보를 생성해 냄으로써, 관리자로 하여금 적시적 판단과 효율적 대응이 가능하도록 해 주고 DDoS 같은 대규모의 공격을 조기에 탐지해 낼 수 있는 능력을 갖춘 시스템의 개발은 필수적이라고 할 수 있으며, 향후 침해사고 대응과 종합적인 보안관리를 위한 기반이 될 것이다.

2. 관련 연구

침입탐지 정보의 연관성 분석을 위한 기존 연구들을 5가지 범주로 나누어 살펴보고자 한다.

2.1 Data Mining Approach

데이타마이닝 기법은 대량의 저장된 데이타로부터 의미있는 패턴을 찾아내는 과정으로서, 연관성 분석을 위해 사용 가능한 다양한 방법론들이 존재한다. 그 중에서 대량의 데이타 내에서 자주 발견되는 연속적 패턴(Sequential pattern)들을 찾아내는데 사용되는 Frequent Episodes 알고리즘은 특정 네트워크에서 탐지된 침입탐지 정보를 대상으로 적용했을 경우, 그 네트워크에서 비교적 자주 발생되는 침입 패턴을 찾아낼 수 있게 된다. 또한 Association Rules 알고리즘은 다양한 요소들 간에 존재하는 모든 연관성을 분석하여 룰의 형식으로 표현해 주기 때문에, 네트워크 도메인에서 중요도가 높은 요소들을 찾아내는데 적용할 수 있다[1]. 그러나 이러한 데이타마이닝 접근방법은 온라인상에서 대량의 데이타를 실시간으로 처리하는 것이 거의 불가능하다는 문제점을 안고 있다.

2.2 Rule-based Approach

사전에 정의된 룰에 의해 연관성 분석을 수행하는 형태로는 Stanford 대학에서 개발한 CIDF Correlator와 Planning Process Model을 들 수 있다.

Stanford CIDF Correlator[2]는 침입탐지 정보들간의 연관성 분석을 통해 거짓 침입경보를 줄이고, DDoS같은 대형 공격을 탐지하고자 하는 의도에서 개발되었으며, CIDF와 CEP(Complex Event Processing)를 통합한 구조를 취하고 있다. 연관성 분석은 이미 정의된 시나리오에 기반을 두고, 탐지되는 침입정보의 패턴이 정의된 시나리오와 일치하는가, 시간적인 관계가 시나리오에 정의된 순서와 일치하는가 등의 여부를 통해 연관성을 가지는 침입탐지 정보들을 찾아낸다.

[3]에서 제안된 Planning Process Model 모델은 AI의 planning process 기법을 적용한 호스트 기반의 IDS 모델로서, 공격이 일어나기 전후의 호스트 시스템에 대한 상황과 공격의 각 단계를 포함하는 공격 시나리오를 모델링하고 이에 기반하여 침입탐지 정보들간의 연관성을 찾아낸다. 그러나 CIDF Correlator와 달리 현재 상태에서 수행 불가능한 공격순서를 정의하고 그러한 경우가 발생했을 때에는 연관성 분석을 중단하는 Anti-Correlation 메커니즘을 포함하고 있으며, 네트워크에 대한 추가적인 정보가 가용하다면 연관성 분석을 위한 룰을 자동으로 생성해 주는 기능도 포함하고 있다. 또한 시나리오와 완전 일치가 되지 않는 경우에는 추론을 통해 시나리오를 완성하는 Abduction 기능을 제공한다.

이러한 룰 기반의 접근방법은 연관성 분석을 위해 사용되는 시나리오만 완벽하다면, 다른 방법론들에 비해 향상된 성능을 보장할 수 있겠지만, 모든 공격에 대한 시나리오를 정의하는 것이 쉽지 않기 때문에 현재로선 제한적인 공격에 대한 연관성 분석만 가능하다.

2.3 Situation-aware Approach

Herve와 Andreas는 [4]에서 객체지향 설계기법이 제공하는 상속관계를 적용하여 침입탐지 정보를 계층구조로 표현하였다. 그리고 근원지 정보(Source), 공격 목표(Target), 공격 클래스(Attack class) 등의 요소에 대해 공통적인 값을 갖는 침입탐지 정보들의 집합을 7가지의 상황(Situation)으로 분류하였다. 센서에서 탐지되는 침입정보들은 판별기준에 따라 해당되는 상황으로 통합되어 나타남으로써 현 네트워크에서 발생되는 상황을 신속하게 파악할 수 있다는 장점을 가진다.

2.4 Probabilistic Approach

[5]에서 제안된 확률론적 접근방법은 공격의 근원지 정보, 공격 목표, 공격 클래스 및 시간정보 등의 요소들 간에 존재하는 유사도를 0과 1사이의 값으로 부여하여, 침입탐지 정보들의 연관성 분석을 수행한다. 즉, 특정한 상황하에서는 완전 일치가 되지 않더라도 최소한의 유사도만 보장되면 2개의 침입탐지 정보간에는 연관성이

존재하는 것으로 판단한다. 공격 클래스간의 유사도 분석에서는 사전에 정의된 유사도 값을 사용하며, 특정 상황 하에서는 반드시 일치되어야만 하는 요소들을 고려하여 각 요소들의 유사도에 대한 기대치를 0과 1사이의 값으로 부여하고, 침입탐지 정보들간의 종합적인 유사도를 분석하는 데 사용한다.

확률론적 접근방법의 또 다른 특징은 센서들간의 연관성 분석을 계층적으로 수행한다는 점이다. 먼저 단일 센서로부터 생성된 침입탐지 정보들은 모든 요소들에 대해 최소한으로 보장되어야만 하는 유사도 기대치를 높은 기준으로 적용하고, 유사도가 존재할 경우 스레드(Thread)로 통합한다. 이기종 센서들 간에는 공격 클래스에만 높은 수준의 유사도 기대치를 적용하여 유사도를 분석하고 Security Incident로 통합한다. 다단계로 수행되는 공격들에 대해서는 공격 클래스에 낮은 유사도 기준을 적용함으로써, 예상되는 공격 시나리오를 찾아내고 Correlated Attack Report를 생성한다.

이러한 확률론적 접근방법은 요소들 간의 완전 일치가 아니더라도 확률적인 방법으로 유사도를 분석하기 때문에 다른 방법론들에 비해 유연성이 있다고 볼 수 있으나, 침입탐지 정보를 대상으로 한 실제 실험에서는 성능상의 한계를 드러내고 있다.

2.5 Prioritizing Approach

[6]에서 제안된 M-Correlator는 침입탐지 정보들을 해석하는 과정에서, 네트워크 사용자들에게 가장 중요한 자원이나 서비스, 자료는 무엇이며 어떤 종류의 공격에 가장 많은 관심을 가져야 하는지를 효율적으로 식별해 내기 위해 각각의 요소에 우선순위를 부여하여 조건부 우선순위 테이블(CPT : Conditional Priority Table)을 만든다. 그리고 운영체제 및 버전 정보, 하드웨어, 운영중인 네트워크 서비스 및 응용 프로그램 등의 토폴로지 정보를 이용하여, 현재 탐지된 침입이 토폴로지에 얼마나 관련성을 가지는지를 판별하고 0부터 255까지의 관련성 점수(Relevance Score)를 부여한다. 최종적으로 각각의 침입사고들에 대한 우선순위 산출에는 "Bayes Calculation" 기법이 적용되며, 조건부 우선순위 테이블 값과 관련성 점수 및 센서로부터의 출력값(공격의 성공여부) 등이 입력으로 사용된다.

3. 시스템 설계

3.1 설계 고려사항

본 논문에서 제안하고 있는 시스템은 정형화된 룰에 의한 연관성 분석보다는 네트워크 형태나 특정 상황에 따라 유연성을 발휘할 수 있는 시스템을 목표로 한다. 따라서 확률론적 접근방법에서와 유사하게 침입탐지 정보에 포함된 각 요소들 간의 유사도에 기반한 연관성 분석을 실시한다.

그리고 각 접근방법에서 무시되거나 혹은 의미가 적은 정보로 취급되었던 시간 정보에 중요한 의미를 부여하고, 과거의 연구를 참고하여 좀 더 체계적인 접근방법을 통하여 시간 정보를 시스템에 적용하고자 한다.

또한, 현 네트워크의 상황과 공격 성향을 신속하게 파악하고 나아가서는 DDoS 나 Worm 등의 대형화된 공격을 조기에 탐지해 낼 수 있는 시스템을 목표로 하여, Situation-aware approach에서의 'situation' 개념을 발전시켜 시스템에 적용한다.

3.2 시스템 구조

다수의 분산된 단위 네트워크에 설치된 침입탐지 센서들로부터 침입탐지 정보를 모아 중앙의 관제센터에서 연관성 분석을 수행하는 통합시스템의 구성은 그림 1과 같으며, 각 구성 모듈의 기능은 다음과 같다.

1) Filter

단위 네트워크에 설치된 센서로부터 생성된 침입탐지 정보들 중에서 중복된 정보들을 통합하여 상위 수준의 Thread Event로 변환, 중앙의 관제센터로 보내는 역할을 수행한다.

2) Aggregator

여러 센서들로부터 전송받은 Thread Event들을 모아서 상위 수준의 Aggregation Event로 변환함으로써 관리되고 있는 전체 네트워크의 상황을 파악하는데 도움을 준다.

3) Correlator

Aggregation Event들 간의 시간적, 인과적 관계를 파악하여 Correlation Event를 생성함으로써 다단계로 수행되는 공격 시나리오를 파악하고, 새로운 공격 패턴에 대한 지식을 축적할 수 있도록 돕는 역할을 수행한다.

4) Situator

Thread Event의 분석을 통해 현 네트워크에 대한 공격 성향을 파악한다. 즉 여러 공격자가 하나의 목표를 집중적으로 공격하는 DDoS나 Worm으로 인한 무작위

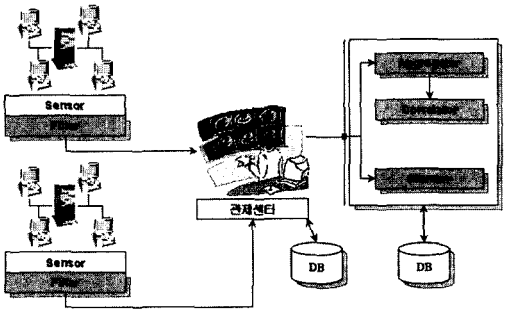


그림 1 전체 시스템 구성도

적 공격을 조기에 파악함으로써 대응시간을 줄이는데 사용할 수 있다.

3.3 이벤트 구조

시스템에서 사용되는 이벤트는 Thread Event, Aggregation Event, Correlation Event, Situation Event의 4가지 종류가 있으며, 각 이벤트 간의 계층 관계는 그림 2와 같다.

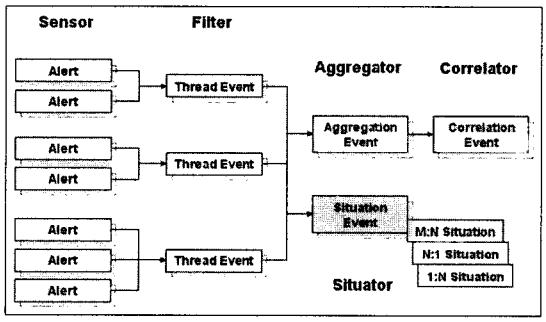


그림 2 이벤트 계층 구조

3.4 모듈별 세부 기능 및 구조

1) 관제센터

Filter로부터 Thread Event를 받아 이를 다른 모듈에 전달하며 관리하는 역할을 수행한다. 센터는 처음 시작할 때, Initialize() 함수를 호출하여 초기화 과정을 수행하며, 이 과정에서 DB 접속 및 각 파라미터의 값을 설정하게 된다.

그리고 관제센터 모듈에는 각 계층의 이벤트를 보여주는 뷰어가 존재하며, 시스템에서 사용될 이벤트들에 대한 자료 구조가 정의되어 있다.

2) Filter

센서로부터 오는 침입탐지 정보를 취합, 중복된 정보를 제거하고 Thread Event를 생성하여 관제센터로 전송하며, Alert Receiver, ThreadEvent Maker, ThreadEvent Sender 모듈로 구성된다. 전체적인 구조 및 처리흐름은 그림 3과 같다.

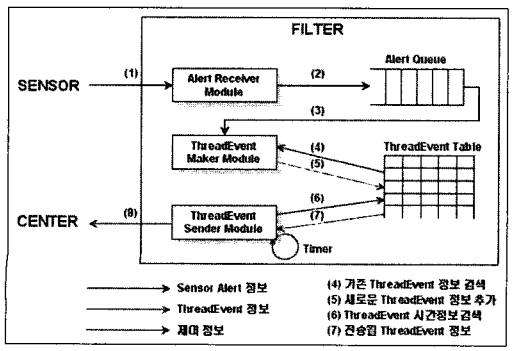


그림 3 Filter 구조 및 처리 흐름

• Alert Receiver

단위 네트워크에 설치된 센서로부터 오는 침입탐지 정보를 수신하는 모듈로서, 본 연구에서 적용 대상으로 하는 네트워크용 침입탐지 시스템 Snort의 로그 정보 구조체인 Alertpkt 구조체의 형식으로 전송 받는다.

• ThreadEvent Maker

Alert Queue에서 침입탐지 정보를 받아 현재의 ThreadEvent Table에 있는 정보와 비교 후 중복된 정보가 있으면 기존 Thread Event와의 통합 과정을 수행하고, 그렇지 않은 경우 새로운 Thread Event를 생성한다.

• ThreadEvent Sender

일정 주기 동안 수집된 Thread Event 정보를 관제센터로 전송한다. 즉, Timer 모듈에 정의되어 있는 이벤트 통합 주기가 지나게 되면 Thread Event 테이블 정보의 갱신을 중지하고 관제센터로 전송한다.

3) Aggregator

Aggregator는 단위 네트워크에 설치된 센서들로부터 전송된 Thread Event들간의 유사도 분석을 통해 공통적인 특성을 가지는 이벤트들을 상위 수준의 메타 이벤트(Aggregation Event)로 통합하는 역할을 수행한다. Filter에 비해 이벤트 통합 주기가 길기 때문에 짧은 통합 주기로 인해 Filter에서 통합되지 못한 Thread Event 들이 Aggregator를 통해 하나의 메타 이벤트로 통합된다.

Aggregator의 구조 및 처리 흐름은 그림 4에서 보는 바와 같다. 관제센터로 새로운 Thread Event가 전송된 경우, 관제센터 내에서 단위 네트워크 및 센서들과의 통신을 담당하는 Network 모듈은 Aggregator 모듈을 호출하게 된다. 호출된 Aggregator 모듈은 DB에서 기존에 저장되어 있던 메타 이벤트 중 특정 기간내의 이벤트들만을 선택하여 전송된 Thread Event와의 유사도 비교 및 통합 과정을 수행한다.

이 때 조건을 만족하는 메타 이벤트가 존재하는 경우

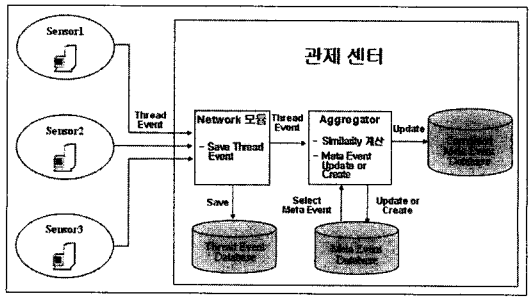


그림 4 Aggregator 구조 및 처리 흐름

에는 해당되는 기존 메타 이벤트를 갱신하고, 조건을 만족하는 메타 이벤트가 존재하지 않는 경우에는 새로운 메타 이벤트를 생성하게 된다.

4) Correlator

Correlator는 Aggregator에서 통합된 메타 이벤트들 간의 유사도 분석을 통해 동일한 근원지에서 동일한 목표에 대해 다단계로 수행되는 1:1 형태의 공격 시나리오를 찾아내는 역할을 수행하며, 시간적 관계를 가지는 이벤트들의 통합을 통해 새로운 다단계 공격 형태에 대한 분석을 가능하게 해 준다.

Correlator의 처리 흐름은 그림 5에서 보는 바와 같다. Correlator는 Aggregator를 통해 처리된 메타 이벤트들만을 처리 대상으로 하며, 유사도를 분석함에 있어서는 Source IP와 Destination IP에 높은 최소 기대치(minimum expectation)를 적용하여 동일인의 특정 목표에 대한 다단계 공격을 파악하는데 중점을 둔다. 전달받은 메타 이벤트가 기존의 이벤트를 갱신한 경우라면, 단순히 해당되는 Correlated Meta Event를 같은 방식으로 갱신하면 된다. Aggregator로부터 새로운 메타 이벤트를 전달받은 경우에는 일정 시간 간격 내에서 생성된 과거 이벤트들만을 대상으로 유사도를 분석하고, 그 결과에 따라 기존의 Correlated Meta Event에 시간 정보와 함께 통합하거나 새로운 Correlated Meta Event를 생성한다.

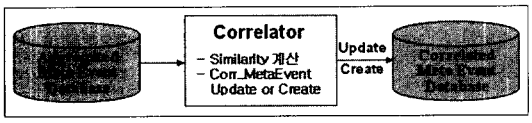


그림 5 Correlator 구조 및 처리 흐름

5) Situator

Situator는 Filter로부터 오는 Thread Event의 공격 근원지와 공격 목표간의 관계 분석을 통하여 현 네트워크에 대한 공격성향을 파악하는 기능을 수행하며, 1:N, N:1, M:N과 같은 3가지의 공격형태로 분류하게 된다.

1:N 유형의 공격은 네트워크 스캔이나 서비스 스캔 공격과 같이 하나의 근원지로부터 여러 목표로의 공격을 의미하며, N:1 유형의 공격은 여러 근원지로부터 하나의 목표로의 공격을 의미하는 것으로서 DDoS 공격이 이에 해당된다. M:N 유형의 공격은 Worm이나 바이러스와 같이 네트워크 전역을 대상으로 하는 공격을 의미한다.

본 논문에서 제안하고자 하는 Situator의 구조 및 처리 흐름은 그림 6에서 보는 바와 같고, 개별적인 공격이벤트들이 후보 리스트에 저장되어 있다가 사전에 정의

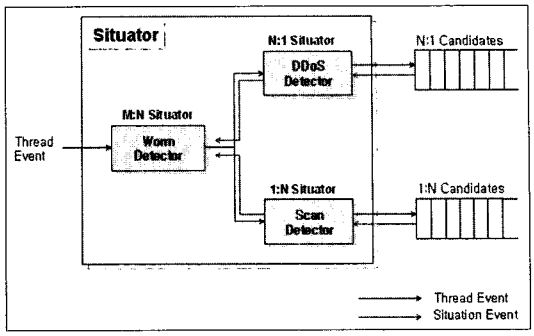


그림 6 Situator 구조 및 처리 흐름

된 한계값을 초과하여 발생된 경우에는 각각의 공격 유형별로 분류된다. 이러한 한계값은 네트워크의 상황이나 공격 트랜드에 따라 관리자가 조정 가능하다.

3.5 유사도 분석 알고리즘

본 논문에서 제안된 시스템은 확률론적 접근방법과 동일하게 침입탐지 정보 내에 포함된 각 요소들간의 유사도를 이용해 연관성 분석을 실시하며, 이 때 사용되는 요소들은 IP 주소, 포트(Port), 공격 클래스(Attack class), 시간 정보(Time information)이다.

1) IP 주소 유사도(IP address similarity)

이벤트가 포함하고 있는 IP 주소에 대한 유사도를 측정한다. 만약 두 이벤트의 근원지 주소가 같은 서브 네트워크에 포함된다면 두 공격자는 동일인일 가능성이 매우 높아지므로, 이를 고려해서 계산한다. 유사도 분석 대상이 되는 두 IP 주소의 불일치하는 길이가 길어질수록 동일인일 확률은 급격하게 감소하므로, 유사도는 로그 스케일(log scale)을 따름을 유추할 수 있다. 두 IP 주소의 유사도에 대한 상세한 기준은 아래와 같으며, 실제적인 기준값은 추후 실험을 통해 계속적으로 변동시켜 나갈 것이다.

- 완전한 일치 : 1
- C 클래스까지 일치 : 0.8
- B 클래스까지 일치 : 0.4
- A 클래스까지 일치 : 0.2
- 완전한 불일치 : 0

2) 포트 유사도(Port similarity)

각 공격 이벤트는 주로 포트의 리스트를 가지게 되고, 포트간의 유사도를 위해서는 비교 대상이 되는 두 공격 이벤트에 포함된 포트 리스트간의 유사도를 계산하는 방법이 요구된다. 입력되는 공격 이벤트는 메타 이벤트에 비해 하위 수준의 이벤트이므로 입력 이벤트의 리스트가 메타 이벤트의 서브셋이 되어야 할 확률이 높다. 그러므로 두 리스트의 유사도는 입력 이벤트 각 원소의 메타 이벤트에 대한 유사도의 평균으로 정의한다. 즉,

입력 이벤트가 L1={x1, x2, ... , xn}을 가지고 메타 이벤트가 L2={y1, y2, ... , ym}을 가진다면 L1과 L2의 유사도 S는 다음과 같이 정의된다.

$$S_i = \max_{1 \le j \le m} \{Similarity(xi, yj)\}$$

$$S = \frac{1}{n} \sum S_i$$

3) 공격 클래스 유사도(Attack class similarity)

이기종의 센서들은 동일한 공격을 서로 다른 클래스의 공격을 판별할 가능성이 존재한다. 때문에 이러한 문제를 방지하기 위해서 다양한 공격 이벤트들을 일정 수준의 클래스로 분류하고, 유사 유형의 공격은 같은 클래스에 속하도록 설정할 필요가 있다.

본 논문에서 제안하는 시스템은 Snort만을 대상으로 하기 때문에 앞서 언급한 문제점이 발생할 가능성은 없지만, 차후 다양한 이기종의 IDS를 수용할 수 있는 시스템으로 확장하기 위해서는 클래스의 분류가 필요하기에 Snort에서 정의된 34개의 클래스를 시스템에 적용하고자 한다.

두 공격 클래스간의 유사도는 사전에 정의된 유사도 행렬 S를 사용한다. 이 S는 34x34의 행렬로 각 값은 0~1 사이의 값이 된다. 메타 이벤트의 공격 클래스가 i이고 입력된 이벤트의 공격 클래스가 j라면 유사도는 S[i, j]가 되는 것이다.

성능분석을 위해 사용된 초기 S값은 DARPA에서 제공하는 침입탐지 데이타에 대한 통계적 분석을 통해 설정하였다.

4) 시간정보 유사도(Time similarity)

두 이벤트의 유사도를 계산하는 데 있어 시간정보 유사도는 상당히 큰 의미를 차지하며, 전체적인 유사도를 계산하는 데에도 큰 영향을 준다. 예를 들어, 다른 요소가 모두 일치하는 이벤트라 하더라도 메타 이벤트가 오래 전에 생성된 것이라면 연관성이 없는 것으로 여겨져야 한다.

공격 성향이나 공격 실행 코드(exploit)의 트랜드에 관한 연구 중 Browne의 연구는 특히 의미가 크다. 그에 따르면 한 공격에 대해 누적되는 이벤트의 수 C는 다음과 같다.

$C = I + S \times \sqrt{M}$ (I, S : 상수, M : 공격 시작 시간)

즉, 공격이 시작된 후 초반에 많은 이벤트가 발생하고 시간이 지날수록 발생 빈도가 떨어지는 것을 알 수 있다. 그러므로 입력 이벤트와 메타 이벤트의 유사도는 두 이벤트의 생성 시간이 가까울수록 훨씬 높아지고 시간 차이가 멀수록 급격히 낮아지게 된다. 즉, 입력 이벤트의 시간을 t2, 메타 이벤트의 생성 시간을 t1라고 하면, 두 이벤트의 시간 정보에 대한 유사도 S는 다음과 같이 정의된다.

$$S = I - V \times \sqrt{t2 - t1}, \quad I, V: \text{상수}$$

5) 전체 유사도(Overall similarity)

앞서 설명한 각 요소별 유사도 분석이 완료되면 두 이벤트간의 전체적인 유사도를 계산하게 된다. 이 과정에서 기대치(expectation) 값과 최소 기대치(minimum expectation) 값은 각각 가중치와 필요조건이 된다. 즉, 기대치 값을 통해서 각 요소 중 중요한 요소의 의미를 부각시킬 수 있고, 최소 기대치 값보다 작은 유사도를 가지는 요소가 있으면 이를 통해서 두 이벤트를 별개의 이벤트로 생각할 수 있게 된다.

입력된 이벤트 X와 메타 이벤트 Y에 대한 유사도 SIM(X,Y)는 확률론적 접근방법에서와 동일하게 다음과 같이 계산된다[5].

$$SIM(X, Y) = \frac{\sum_j E_j S(X_j, Y_j)}{\sum_j E_j}$$

j : 각 *feature*의 *index*

E_j : *feature* j의 *expectation*

X_j, Y_j : *feature* j에 대한 X와 Y의 similarity

이렇게 입력 이벤트와 각 메타 이벤트간의 유사도를 계산한 후, 이 중 가장 큰 값이 연관성 분석을 위한 후보값이 된다. 만약 SIM(X,Y)가 설정된 한계치(threshold) 값보다 크다면 두 이벤트가 통합되어 하나의 메타 이벤트가 되고, 그렇지 않은 경우에는 입력 이벤트가 새로운 메타 이벤트로 생성된다.

4. 시스템 구현

본 논문에서 제안된 연관성 분석 시스템의 센서는 NIDS인 Snort를 기반으로 하고 있으며, 관제 센터는 Unix에서 C 언어를 사용하여 구현하였다. 센서와 Filter는 Linux 기반으로 운용되며, 관제센터에서 사용된 데이타베이스는 Unix용 Oracle이다.

5. 시스템 분석 및 성능 평가

5.1 기존 연관성 분석 시스템들과의 비교

성능평가에 앞서 본 논문에서 제안하는 연관성 분석 시스템의 특징을 기존 연관성 분석 시스템들과의 비교를 통해 살펴보고자 한다.

제안된 시스템은 3장의 설계 고려사항에서 기술한 바와 같이 확률론적 접근방법에 기반하고 있다. 따라서 확률론적 기법에 의한 유사도 산출을 통해 연관성을 분석한다는 기본적인 전략은 동일하지만, 기존의 확률론적 접근방법과는 차별화된 특징들을 가지고 있다. 우선 제안된 시스템은 대량의 침입탐지 정보에서 중복되는 요소를 제거하고 필수적인 정보들만 추출하기 위해 1차적

으로 침입탐지 시스템과 연동되는 필터에서 필터링을 실시하고 2차적으로는 관제센터 내에서 Aggregator에 의한 통합이 이루어진다. 즉 모든 침입탐지 정보를 대상으로 하는 확률론적 접근방법과는 달리 중복 요소를 제거한 정보들만을 대상으로 연관성 분석을 실시함으로써 시스템의 부하를 줄이고 있다. 또한 확률론적 접근방법뿐만 아니라 대부분의 기존 연구에서 등한시하거나 체계적이지 못한 방법으로 시스템에 적용되었던 시간 정보에 중요한 의미를 부과하고, Browne의 연구를 참고하여 체계적으로 접근 및 적용하였기에 기존의 시스템들에 비해서는 정확한 결과를 얻을 수 있다. 그리고 제안된 시스템은 공격 클래스의 유사도 분석에 사용되는 행렬의 초기치를 구하기 위해 DARPA에서 제공하는 자료에 대해 데이타마이닝 기법을 적용하였다.

Stanford CIDF Correlator나 Planning Process Model 같은 룰 기반의 접근방법과 제안된 시스템은 기본적인 접근전략에서 차이가 있으며, 정형화된 룰에 의해 일부 공격에 대해서만 연관성 분석이 가능한 룰 기반의 접근방법과 달리 제안된 시스템은 새로운 공격 유형에 대해서도 탐지가 가능하다는 장점을 가진다.

현 네트워크의 상황과 공격 성향을 신속하게 파악하고 나아가서는 DDoS 나 Worm 등의 대형화된 공격을 조기에 탐지해 내기 위한 'Situator' 모듈은 Situation-aware approach에서의 'situation' 개념을 확장 발전시킨 형태이다. 그러나 Situation-aware approach에서의 situation은 단순히 동일한 특성을 가지는 침입탐지 정보들을 7가지의 집합으로 분류하는데 그치고 있을 뿐, 제안된 시스템에서와 같이 현 네트워크의 상황 분석이나 대형 공격에 대한 탐지는 불가능하다.

본 논문에서 제안하는 시스템과 기존의 연관성 분석 시스템들과의 비교 결과는 표 1에서 보는 바와 같다. 요컨대 제안된 시스템은 기존의 시스템들에 비해 연관성 분석 과정에서 유연성을 가지며, 체계적인 방식에 의한 시간정보 적용으로 비교적 정확한 연관성 분석 결과를 도출해 낸다. 또한, 자동화된 실시간 정보처리로 관리자의 오버헤드를 감소시킴은 물론 기존의 시스템에서는 볼 수 없었던 대형화된 공격의 조기 탐지 또한 가능하다는 장점을 가진다.

5.2 성능 평가

제안된 연관성 분석 시스템의 처리 능력 및 동작성능을 평가하기 위해 2단계에 걸쳐 대량의 침입탐지 정보를 줄여주는 역할을 수행하는 Filter와 Aggregator 모듈의 압축률 및 각 단위 모듈별 수행시간을 측정하였다.

그리고 제안된 시스템의 성능 평가를 위해 테스트베드 상에서 알려진 공격 기법과 도구들을 이용하여 다음과 같은 공격 시나리오들을 구성하여 시험 평가를 실시하였다.

- Scenario #1 : 특정 호스트에 대한 스텔스 스캔(Stealth Scan)
- Scenario #2 : FTP 서버에 대한 버퍼 오버플로우(Buffer Overflow) 공격
- Scenario #3 : Web 서버에 대한 CGI 공격
- Scenario #4 : RPC 서비스에 대한 버퍼 오버플로우 공격
- Scenario #5 : 다수의 호스트를 대상으로 하는 네트워크 스캔(Network Scan)[1:N 공격]
- Scenario #6 : DDoS 공격[N:1 공격]
- Scenario #7 : Worm/Virus에 의한 공격[N:M 공격]

DARPA에서 제공되는 침입탐지 자료는 IDS의 성능 평가를 위해서는 유용하나 본 연구에서 제안된 유형의 시스템 평가에는 부적절하며, 현재까지는 사이버공격 침입정보의 연관성 분석 능력을 평가할 수 있는 기준이나 자료가 제공되지 않는 실정이기 때문에 우선적으로 알려진 공격 시나리오에 기반한 시험 평가만을 실시하였다. 본 논문에서는 상기 시나리오 중 Scenario #4, Scenario #6에 대한 결과를 중심으로 Correlator와 Situator에 성능 분석 결과를 제시하고자 한다.

1) Filter/Aggregator 모듈의 압축률

Filter 및 Aggregator 모듈의 압축률은 시험기간 동안 발생한 총 공격 이벤트(Snort Alert) 수와 Filter 및 Aggregator에서 중복 요소를 제거하고 통합하여 생성해 낸 이벤트의 수를 이용하여 측정하였으며, 그 결과는 표 2에서 보는 바와 같다.

표 1 제안된 시스템과 기존 연관성 분석 시스템과의 비교 결과
(○ : 우수 또는 해당항목 구비, X : 해당 사항 없음, △ : 중간 수준 또는 비체계적 접근)

평가요소 / 접근방법	실시간 정보처리	시간정보 적용	시스템 유연성	대형공격 조기탐지	관리상의 오버헤드 발생여부
데이타마이닝	X	○	△	X	○
Rule-based Approach	○	△	X	△	○
Situation-aware Approach	○	△	X	△	X
Probabilistic Approach	○	△	○	X	X
Prioritizing Approach	○	△	△	X	○
제안된 시스템	○	○	○	○	X

표 2 Filter/Aggregator 모듈의 압축률

모듈	시험 기간 중 발생한 총 Alert 수	599403
Filter	Thread Event 수	66775
	평균압축률(# of Thread Event / # of Alert)	11.1%
Aggregator	Aggregation Event 수	33173
	평균압축률(# of Agggation Event / # of Alert)	5.5%

관제 센서로 Thread Event를 전송하는 주기를 1분으로 설정했을 경우 Filter의 압축률은 평균 11% 수준이었으며, ICMP Ping CyberKit에 의한 Nachi Worm의 경우에는 Thread Event 하나당 평균 20개 정도의 침입탐지 정보가 통합되어 최대 5%의 압축률을 보였다.

2) 각 모듈별 수행 시간

각 모듈별 처리 효율을 평가하기 위해 Filter에서 생성된 하나의 Thread Event가 각 모듈에서 처리되는데 걸리는 수행 시간을 측정하였으며, 그 결과는 표 3에서 보는 바와 같다.

표 3에서 알 수 있는 바와 같이 하나의 Thread Event가 제안된 시스템에서 처리 완료되는데 걸리는 시간은 0.8초로서 실시간에 가까운 처리 성능을 가진다. 또한, 처리의 전 과정이 자동으로 이루어지기 때문에 관리 및 분석 측면에서도 관리자의 수작업에 의존해 처리해야만 했던 기존의 방식에 비해 관리의 효율성을 향상시킴은 물론 대응시간과 관리자에게 부과되는 오버헤드 감소에도 크게 기여한다.

표 3 모듈별 수행 시간

모 듈	수행 시간(sec)
Thread event 저장	0.0097
Aggregator	0.5398
Correlator	0.0874
UpdateCorrelator	0.0013
Situator	0.1691
Total	0.8103

3) Correlator 성능 평가(Scenario #4)

네트워크 상에서 서비스 호출을 위해 사용되는 RPC 서비스의 취약점을 이용한 공격으로서, 다음과 같은 단계로 공격을 진행하였다.

- Step 1 : 호스트 스캔
- Step 2 : RPC 서비스의 취약점을 공격하기 위한 Exploit 실행
- Step 3 : Root Shell 상태에서 중요 시스템 파일 획득

호스트 스캔을 위해 NMAP을 이용하였으며, RPC 서비스의 취약점을 이용하여 버퍼 오버플로우 공격을 실행한 결과 그림 7에서 보는 바와 같이 공격자는 관리자 권한을 획득하였다.

```
[root@caatm3 sjlee]# ./rpc_exploit -t 143.248.137.84 -d 0
buffer: 0xbffff314 length: 999 (+str/+nul)
target: 0xbffff718 new: 0xbffff56c (offset: 600)
wiping 9 dwords
clnt_call(): RPC: Timed out
A timeout was expected. Attempting connection to shell..
OMG! You now have rpc.statd technique!@#$!
total 99
drwxr-xr-x   17 root     root         4096 Oct 10 08:45 ./
drwxr-xr-x   17 root     root         4096 Oct 10 08:45 ../
drwxr-xr-x    2 root     root         4096 Oct 13 22:13 bin/
drwxr-xr-x    3 root     root         4096 Oct 20 21:26 boot/
drwxr-xr-x    6 root     root        36864 Oct 20 21:27 dev/
drwxr-xr-x   29 root     root         4096 Oct 20 23:51 etc/
drwxr-xr-x    9 root     root         1024 Oct 20 20:08 home/
drwxr-xr-x    4 root     root         4096 Oct 10 08:53 lib/
drwxr-xr-x    2 root     root        16384 Oct 10 08:44 lost+found/
drwxr-xr-x    4 root     root         4096 Oct 10 08:45 mnt/
drwxr-xr-x    3 root     root         1024 Aug 24  1999 opt/
dr-xr-xr-x   45 root     root            0 Oct 21 06:25 proc/
drwxr-x---    2 root     root         4096 Oct 10 00:47 root/
drwxr-xr-x    3 root     root         4096 Oct 10 08:55 sbin/
drwxrwxrwt    4 root     root         4096 Oct 22 04:02 tmp/
drwxr-xr-x   21 root     root         4096 Oct 10 08:50 usr/
drwxr-xr-x   20 root     root         1024 Oct 10 08:55 var/
uid=0(root) gid=0(root)
```

그림 7 RPC 버퍼 오버플로우 공격 실행 결과

그림 8은 RPC 버퍼 오버플로우 공격에 대한 탐지 정보를 Filter가 처리하여 관제 센서로 전송한 Thread Event 로서, 연관성을 예측하기 어려운 다수의 이벤트가 생성됨을 알 수 있다.

이러한 다수의 침입탐지 정보들이 연관성 분석을 통해 처리된 결과는 그림 9와 10에서 보는 바와 같다.

그림 10의 상단에서 보는 바와 같이 개발된 시스템은 일련의 공격들이 시간적, 인과적 관계를 가지고 한명의 공격자에 의해 시도되었음을 명확하게 보여주고 있으며, 하단에서와 같은 세부적인 공격 시그너쳐 까지도 확인 가능하다.

4) Situator 성능 평가(Scenario #6)

Situator 모듈의 정상 작동 여부를 검증하기 위해 1:N, N:1, N:M의 형태로 공격 시나리오를 분류하여 성능 분석을 실시하였으나, 본 논문에서는 Scenario #6에 대한 결과만을 보이고자 한다.

DDoS 공격은 다수의 호스트가 특정 호스트의 서비스 제공이나 정상작동을 방해할 목적으로 실시하는 공격으로서, 대부분의 상용 IDS들은 탐지가 불가능한 실정이다. 그러나 본 논문에서 제안된 연관성 분석 시스템의 Situator 모듈은 실시간에 가까운 수준으로 현 네트워크에서 DDoS 유형의 공격이 행해지고 있음을 탐지하였다.

DDoS 공격의 조기 탐지 능력 평가를 위해서 다수의 공격 호스트에서 ICMP Flooder를 이용하여 그림 11에

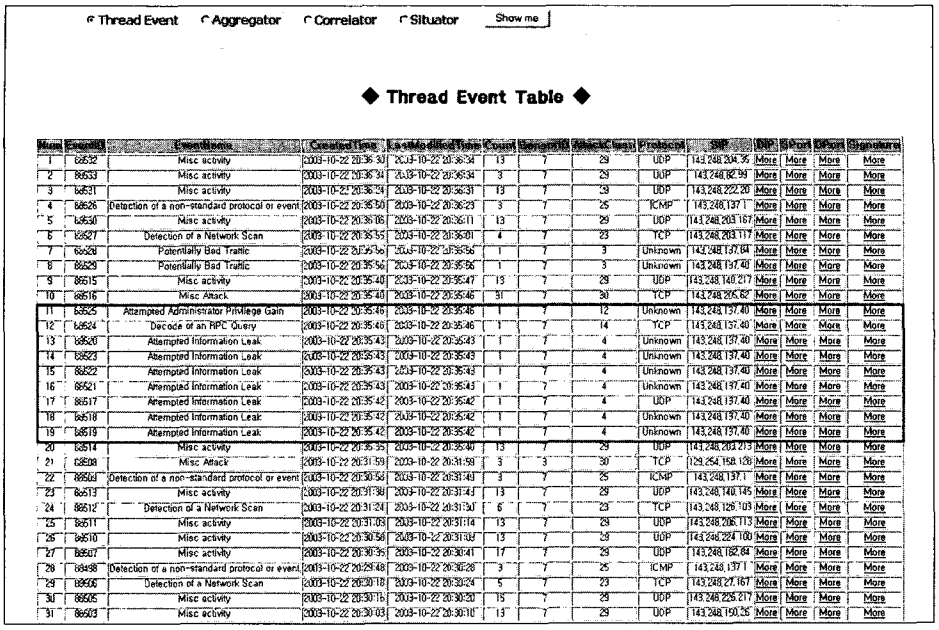


그림 8 RPC 공격 탐지 결과 : Thread Event

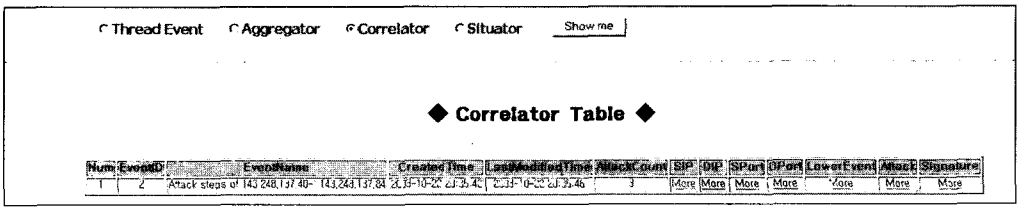


그림 9 RPC 버퍼 오버플로우 공격에 대한 연관성 분석 결과 : Correlation Event

Correlator Lower Events with ID = 2

EventID	EventName	CreatedTime	LastModifiedTime	Count	SIP	DIP	SPort	DPort	LowerEvent	Signature
50004	Attempted Information Leak	2003-10-22 20:35:42	2003-10-22 20:35:43	7	More	More	More	More	More	More
50005	Decode of an RPC Query	2003-10-22 20:35:46	2003-10-22 20:35:46	1	More	More	More	More	More	More
50006	Attempted Administrator Privilege Gain	2003-10-22 20:35:46	2003-10-22 20:35:46	1	More	More	More	More	More	More

Correlator Signature Lists with ID = 2

EventID	AttackClass	Signature	Count
2	4	ICMP PING NMAP	1
2	12	RPC STATD TCP stat mon_name format string exploit attempt	1
2	14	RPC portmap status request UDP	1
2	4	SCAN Proxy (8080) attempt	1
2	4	SCAN SOCKS Proxy attempt	1
2	4	SCAN Squid Proxy attempt	1
2	4	SNMP AgentX/tcp request	1
2	4	SNMP request tcp	1
2	4	SNMP trap tcp	1

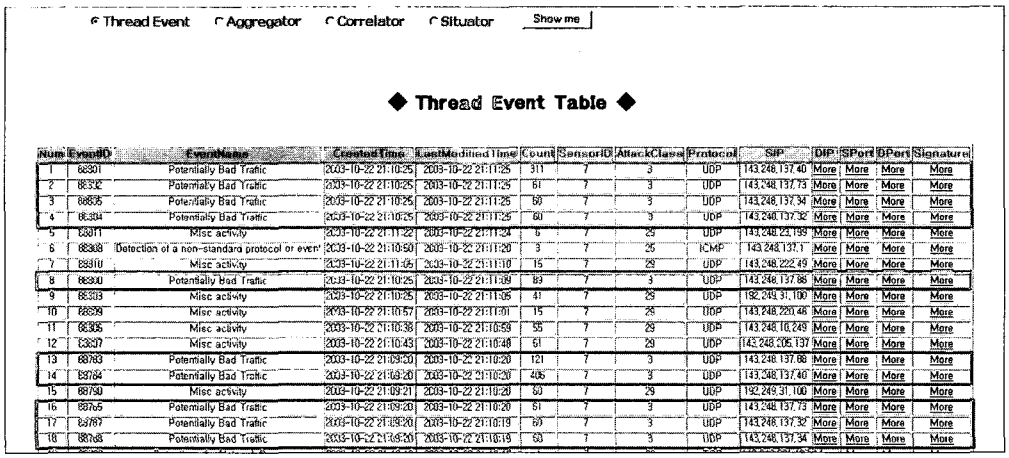
그림 10 Correlation Event : 세부 정보 확인

```
ICMP Flooder v0.2 By: Code

Notes:
- It Is Normal For All Ping Requests To Time-Out
- This Prog Will Do A Direct Ping To The IP Specified
- To get the persons IP from mIRC Type /dns nick

IP Address To Flood?
? 143.248.137.48

Bytes Per Ping (2500-3000 rec. for 28.8):
? 3000

Number Of Pings(or 0 to keep going until you press Ctrl-Break):
? 0
```

그림 11 DDoS 공격 실행

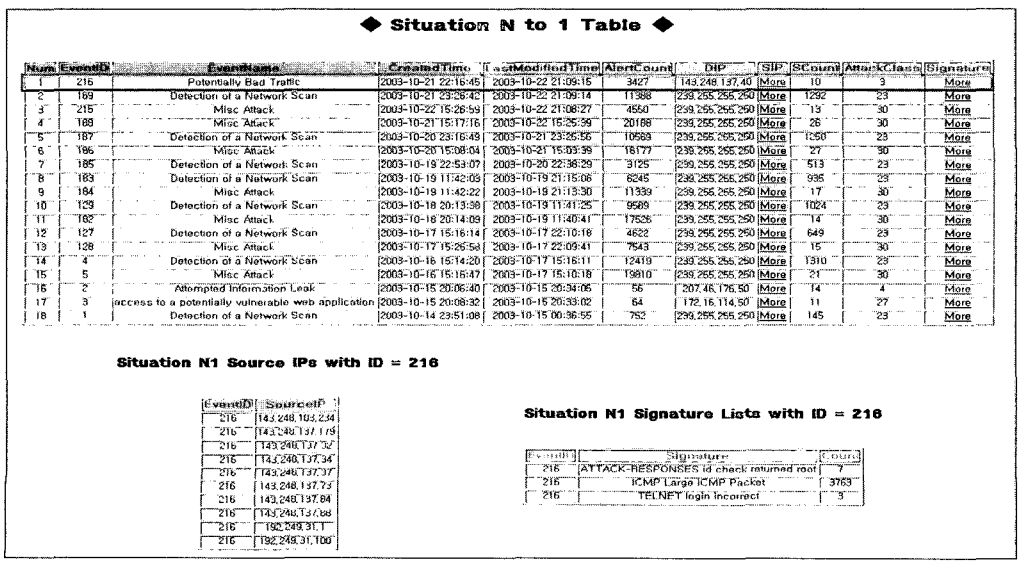
서 보는 바와 같이 공격 대상 호스트에 대해 3000Byte의 대용량 패킷을 계속 전송하는 서비스 거부 공격을 실행하였다. 이러한 DDoS 공격이 제안된 시스템에 의해 탐지되고 분석된 결과는 그림 12와 13에서 보는 바와 같다.

그림 12에서 보는 바와 같이 현재 관리 대상 네트워크에서는 다양한 공격들이 발생하고 있으며, 그 중 10명의 공격자가 동일한 유형의 공격을 실시하고 있는 것이 탐지되었다. 이러한 기초적인 탐지정보는 연관성 분석과정을 거쳐 그림 13에서 보는 바와 같이 하나의 공격 이벤트로 통합되어 나타난다. 즉 다수의 공격 이벤트가 동일한 시그너쳐와 공격 대상을 가지기 때문에 Situator에



Num	EventID	EventName	CreatedTime	LastModifiedTime	Count	SensorID	AttackClass	Protocol	SIP	DIP	SPort	DPort	Signature
1	88301	Potentially Bad Traffic	2003-10-22 21:10:25	2003-10-22 21:11:25	311	7	3	UDP	143.248.137.40	More	More	More	More
2	88312	Potentially Bad Traffic	2003-10-22 21:10:25	2003-10-22 21:11:25	61	7	3	UDP	143.248.137.73	More	More	More	More
3	88305	Potentially Bad Traffic	2003-10-22 21:10:25	2003-10-22 21:11:25	60	7	3	UDP	143.248.137.34	More	More	More	More
4	88304	Potentially Bad Traffic	2003-10-22 21:10:25	2003-10-22 21:11:25	60	7	3	UDP	143.248.137.32	More	More	More	More
5	88311	Misc activity	2003-10-22 21:11:22	2003-10-22 21:11:24	6	7	29	UDP	143.248.23.199	More	More	More	More
6	88308	Detection of a non-standard protocol or event	2003-10-22 21:10:50	2003-10-22 21:11:20	3	7	26	ICMP	143.248.137.1	More	More	More	More
7	88310	Misc activity	2003-10-22 21:11:05	2003-10-22 21:11:10	15	7	29	UDP	143.248.222.49	More	More	More	More
8	88300	Potentially Bad Traffic	2003-10-22 21:10:25	2003-10-22 21:11:09	89	7	3	UDP	143.248.137.86	More	More	More	More
9	88309	Misc activity	2003-10-22 21:10:25	2003-10-22 21:11:05	41	7	29	UDP	192.249.31.100	More	More	More	More
10	88299	Misc activity	2003-10-22 21:10:57	2003-10-22 21:11:01	15	7	29	UDP	143.248.220.46	More	More	More	More
11	88306	Misc activity	2003-10-22 21:10:38	2003-10-22 21:10:59	55	7	29	UDP	143.248.10.249	More	More	More	More
12	88307	Misc activity	2003-10-22 21:10:43	2003-10-22 21:10:48	61	7	29	UDP	143.248.205.137	More	More	More	More
13	88283	Potentially Bad Traffic	2003-10-22 21:09:20	2003-10-22 21:10:20	121	7	3	UDP	143.248.137.86	More	More	More	More
14	88284	Potentially Bad Traffic	2003-10-22 21:09:20	2003-10-22 21:10:20	405	7	3	UDP	143.248.137.40	More	More	More	More
15	88290	Misc activity	2003-10-22 21:09:21	2003-10-22 21:10:20	60	7	29	UDP	192.249.31.100	More	More	More	More
16	88285	Potentially Bad Traffic	2003-10-22 21:09:20	2003-10-22 21:10:20	61	7	3	UDP	143.248.137.73	More	More	More	More
17	88287	Potentially Bad Traffic	2003-10-22 21:09:20	2003-10-22 21:10:19	60	7	3	UDP	143.248.137.32	More	More	More	More
18	88288	Potentially Bad Traffic	2003-10-22 21:09:20	2003-10-22 21:10:19	60	7	3	UDP	143.248.137.34	More	More	More	More

그림 12 DDoS 공격 탐지 결과 : Thread Event



Num	EventID	EventName	CreatedTime	LastModifiedTime	AlertCount	DIP	SIP	SCount	AttackClass	Signature
1	216	Potentially Bad Traffic	2003-10-21 22:16:45	2003-10-22 21:09:15	3427	143.248.137.40	More	10	3	More
2	169	Detection of a Network Scan	2003-10-21 23:26:42	2003-10-22 21:09:14	11388	239.255.255.250	More	1292	23	More
3	215	Misc Attack	2003-10-22 15:26:59	2003-10-22 21:08:27	4550	239.255.255.250	More	15	30	More
4	188	Misc Attack	2003-10-21 15:17:16	2003-10-22 15:25:39	20188	239.255.255.250	More	26	30	More
5	187	Detection of a Network Scan	2003-10-20 23:16:49	2003-10-21 23:25:55	10589	239.255.255.250	More	1250	23	More
6	186	Misc Attack	2003-10-20 15:08:04	2003-10-21 15:03:39	18177	239.255.255.250	More	27	30	More
7	185	Detection of a Network Scan	2003-10-19 22:53:07	2003-10-20 22:36:29	3125	239.255.255.250	More	513	23	More
8	183	Detection of a Network Scan	2003-10-19 11:42:05	2003-10-19 21:15:06	6245	239.255.255.250	More	935	23	More
9	184	Misc Attack	2003-10-19 11:42:22	2003-10-19 21:13:30	11339	239.255.255.250	More	17	30	More
10	129	Detection of a Network Scan	2003-10-18 20:13:36	2003-10-19 11:41:25	9589	239.255.255.250	More	1024	23	More
11	182	Misc Attack	2003-10-16 20:14:09	2003-10-19 11:40:41	17526	239.255.255.250	More	14	30	More
12	127	Detection of a Network Scan	2003-10-17 15:16:14	2003-10-17 22:10:18	4622	239.255.255.250	More	649	23	More
13	128	Misc Attack	2003-10-17 15:26:54	2003-10-17 22:09:41	7543	239.255.255.250	More	15	30	More
14	4	Detection of a Network Scan	2003-10-16 15:14:20	2003-10-17 15:16:11	12419	239.255.255.250	More	1310	23	More
15	5	Misc Attack	2003-10-16 15:16:47	2003-10-17 15:16:18	19810	239.255.255.250	More	21	30	More
16	2	Attempted Information Leak	2003-10-15 20:06:40	2003-10-15 20:34:06	56	207.46.176.50	More	14	4	More
17	3	access to a potentially vulnerable web application	2003-10-15 20:08:32	2003-10-15 20:33:02	64	172.16.114.50	More	11	27	More
18	1	Detection of a Network Scan	2003-10-14 23:51:08	2003-10-15 00:36:55	752	239.255.255.250	More	145	23	More

Situation N1 Source IPs with ID = 216

EventID	SourceIP
216	143.248.103.234
216	143.248.137.119
216	143.248.137.32
216	143.248.137.34
216	143.248.137.37
216	143.248.137.73
216	143.248.137.86
216	143.248.137.88
216	192.249.31.1
216	192.249.31.100

Situation N1 Signature Lists with ID = 216

EventID	Signature	Count
216	ATTACK-RESPONSES id check returned root	7
216	ICMP Large ICMP Packet	3763
216	TELNET login incorrect	3

그림 13 DDoS 공격 탐지 결과 : Situation Event

서 하나의 추상화된 이벤트로 표현되는 것이다.

또한, 동일한 공격을 실시하고 있는 공격 호스트들을 쉽게 파악할 수 있고 세부적인 공격 시그너쳐도 확인 가능하기 때문에, 기존의 IDS들은 불가능했던 DDoS 공격의 조기 탐지가 가능하다.

6. 결론 및 향후 연구

본 논문에서는 대량의 침입탐지 정보를 적절히 분석 및 가공하여 관리자에게 필요한 고수준의 정보를 생성해내고, DDoS나 Worm 등의 대규모 공격을 조기에 탐지해 낼 수 있는 침입탐지 정보 연관성 분석 시스템을 제안하였다. 그리고 기존 연관성 분석 시스템들과의 비교를 통해 제안된 시스템이 가지는 특징들을 살펴보았다.

성능 분석 결과에서 알 수 있는 바와 같이, 제안된 시스템은 연관성 분석을 통해 실시간으로 관리 및 분석에 필요한 고수준의 정보를 제공해 주며, 현 네트워크에서 발생되고 있는 다단계 공격이나 기타 공격 유형들을 조기에 파악할 수 있도록 해 준다.

그러나 제안된 시스템의 성능 평가에 필요한 객관적인 기준이나 평가용 자료가 없어 이미 알려진 공격 시나리오만으로 평가를 진행하였는데, 차후에는 좀 더 다양한 공격 시나리오를 구성하여 실험을 진행하고자 한다. 그리고 현재 NIDS인 Snort만을 대상으로 하고 있는 시스템을 확장하여 호스트 기반의 IDS와 다른 종류의 NIDS를 수용할 수 있는 시스템으로 개선해 나갈 예정이다.

참 고 문 헌

[1] Wenke Lee,. "A Framework for Constructing Features and Models for Intrusion Detection System," *PhD thesis*, Columbia University, June 1999.

[2] L. Perrochon, E. Jang, and D.C. Luckham, "Enlisting Event Patterns for Cyber Battlefield Awareness," *DARPA Information Survivability Conference & Exposition (DISCEX'00)*, Hilton Head, South Carolina, January 2000.

[3] F. Cuppens, "Correlation in an intrusion detection process," *Internet Security Communication Workshop(SECI02)*, Tunis- Tunisia, September 2002.

[4] H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts," *Proceedings of 2001 International Workshop on Recent Advances in Intrusion Detection*, Davis, CA, October 2001.

[5] A. Valdes and K. Skinne, "Probabilistic Alert Correlation," *Fourth International Workshop on the Recent Advances in Intrusion Detection*, Davis, USA, October 2001.

[6] Phillip A. Porras, et al, "A Mission impact-Based Approach to INFOSEC Alarm Correlation," *Fifth International Workshop on the Recent Advances in Intrusion Detection*, Zurich, Switzerland, October 2002.

[7] P. Porras and P. Neumann, "Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances," *National Security Conference*, 1997.

[8] E. Bloedorn, et al, "Data Mining for Network Intrusion Detection: How to Get Started," *MITRE Technical Report*, August 2001.

[9] F. Cuppens, "Cooperative Intrusion Detection," *International Symposium "Information Superiority: Tools for Crisis & Conflict-Management,"* Paris, France, September, 2001.

[10] F. Cuppens, "Managing alerts in a multi intrusion detection environment," *17th Annual Computer Security Applications Conference (ACSAC)*, New Orleans, December 2001.

[11] Bugtraq. Security Focus Online. http://online.securityfocus.com/archive/1

[12] CERT Coordination Center. Cert/CC Advisories Carnegie Mellon, Software Engineering Institute. Online. http://www.cert.org/advisories/

[13] C. Kahn, P.A. Porras, S. Staniford-Chen, and B. Tung, *"A Common Intrusion Detection Framework,"* http://www.gidos.org.

[14] K. Kendall, *"A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems,"* Master's Thesis, Massachusetts Institute of Technology, June 1999.

[15] W. Lee, R.A. Nimbalkar, K.K. Yee, S.B. Patil, P.H. Desai, T.T. Tran, and S.J. Stolfo, *"A Data Mining and CIDF-Based Approach for Detecting Novel and Distributed Intrusions,"* Proceedings 2000 International Workshop on Recent Advances in Intrusion Detection (RAID), Toulouse, France, October 2000.

[16] NMAP Network Mapping tool. http://www.insecure.org/nmap/

이 수 진

1992년 3월 육군사관학교 전산과 학사 1996년 2월 연세대학교 컴퓨터과학과 석사. 1996년~1999년 육군교육사령부 전산개발처. 1999년~2001년 육군사관학교 전산소. 2002년 3월~현재 한국과학기술원 전자전산학과 박사과정 재학중. 관심분야는 네트워크 보안, 침입탐지, Ad-hoc 네트워크 및 센서 네트워크 보안

정 병 천

1998년 2월 성균관대 정보공학과 학사 2001년 2월 한국과학기술원 전자전산학과 석사. 2001년 3월~현재 한국과학기술원 전자전산학과 박사과정 재학중. 관심분야는 암호학, 네트워크 보안

김 희 열

2000년 2월 한국과학기술원 전산학과 학사. 2002년 2월 한국과학기술원 전자전산학과 석사. 2002년 3월~현재 한국과학기술원 전자전산학과 박사과정 재학중 관심분야는 암호학, 네트워크 보안

이 윤 호

2000년 2월 한국과학기술원 전산학과 학사. 2002년 2월 한국과학기술원 전자전산학과 석사. 2002년 3월~현재 한국과학기술원 전자전산학과 박사과정 재학중 관심분야는 암호학, 네트워크 보안

윤 현 수

1979년 서울대학교 전자공학과 학사 1981년 한국과학기술원 전산학과 석사 1981년~1984년 삼성전자 연구원. 1988년 오하이오 주립대학 전산학 박사. 1988년~1989년 AT&T Bell Labs. 연구원 1989년~현재 한국과학기술원 전산학과 교수. 관심분야는 Ad-hoc망, 네트워크 보안, 암호학, 상호연결 네트워크

김 도 환

2000년 2월 서울대학교 전산과 학사 2002년 2월 서울대학교 컴퓨터공학과 석사. 2002년~현재 국가보안기술연구소 연구원. 관심분야는 네트워크 보안, 침입탐지, 대규모 네트워크 트래픽 분석

이 은 영

2001년 2월 아주대학교 정보및컴퓨터공학부 학사. 2003년 2월 한국과학기술원 전자전산학과 석사. 2003년~현재 국가보안기술연구소 연구원

박 웅 기

1986년 2월 중앙대학교 전산학과 학사 1988년 2월 중앙대학교 전산학과 석사 1988년 2월~1999년 12월 한국전자통신연구원 선임연구원. 2000년 1월~현재 국가보안기술연구소 책임연구원