대규모 네트워크 상의 다중공격에 대비한 분산 침입탐지시스템의 설계 및 구현†

최주영, 최은정, 김명주
서울여자대학교 대학원 컴퓨터학 전공

요 약

인터넷의 급속한 발달은 대규모 네트워크를 형성하였고 그에 따른 정보를 위협하는 요소의 형태 역시 점차 다양화되고 분산화된 형태를 나타낸다. 이러한 공격 형태의 변화에 대응하기 위해서는 단일 탐지 엔진을 갖는 침입탐지시스템만으로는 탐지가 어렵게 된다. 본 논문에서는 기존의 침입 탐지 기능 이외에 대규모 네트워크 상에서 이루어지는 다중 동시 공격을 효과적으로 감지할 수 있는 분산 침입탐지시스템을 설계 구현하였다. 이를 위해 감시 대상 호스트들에 침입탐지엔진을 독립된 에이전트로 설치한 후, 이들이 생성하는 감사 데이터를 토대로 하여 침입 판정을 내린다. 독립적으로 운영되었던 이전 버전에 비하여 새로운 버전은 이처럼 분산화되어 있을 뿐만 아니라 탐지 규칙에 대한 세계 표준화 동향인 CVE를 따르도록 개선되었다.

The Design and Implementation of A Distributed Intrusion Detection System for Multiple Attacks

Ju-Young Choi, Eun-Jung Choi and Myuhng-Joo Kim

ABSTRACT

For multiple attacks through large networks e.g., internet, IDS had better be installed over several hosts and collect all the audit data from them with appropriate synthesis. We propose a new distributed intrusion detection system called SPIDER II which is the upgraded version of the previous standalone IDS - SPIDER I. As like the previous version, SPIDER II has been implemented on Linux Accel 6.1 in GNU C. After planting intrusion detection engines over several target hosts as active agents, the administration module of SPIDER II receives all the logs from agents and analyzes them. For the world-wide standardization on IDS, SPIDER II is compatible with MITRE's CVE(Common Vulnerabilities and Exposures).

1. 서 론

네트워크의 발전과 인터넷의 대중화로 인해, 오늘날 정보 인프라는 대규모 네트워크를 형성하게 되었다. 이러한 네트워크를 통해 방대하고 다양한 정보들의 전송이 이루어지게 되었고, 그 안에서 기밀, 프라이버시 관련 정보가 차지하는 비율도 높아지게 되었다. 부가가치를 갖는 정보의 위조 및 변조에 대한 불안도 높아지면서 이에 대응할 수 있는 다양한 정보보안의 방법들이 제시되고 있다. 단순히, 시스템이 저장하고 있는 데이터 보안뿐만 아니라 네트워크를 통해 전송되는 데이터 보안 역시 중요하게 다루어야 한다. 최근 정보보안 침해사고의 추세는 특정 시스템에 한정해서 일어나는 것이 아니라 조직, 기관의 네트워크 상에서 분산되어 다중적으로 행해지고 있다[1].

대규모 네트워크로 발전되면서, 이에 대한 공격 형태는 분산화 되고 에이전트화 되면서, 은닉성 등의 형태를 갖게 되었다. 이러한 공격을 탐지하기 위해서는 침입탐지시스템을 활용할 수 있다. 그러나, 기존의 중앙 집중적인 침입탐지시스템은 해당 네트워크에 대한 국소적인 탐지를 주로 하기 때문에 대규모 네트워크 상의 다중공격을 탐지하기에는 다소 어려움이 있다. 이를 위해 탐지, 최종적으로 통합하여 판단할 수 있는 상호 보완적인 분산 침입탐지시스템이 필요하다[2].

따라서, 본 논문에서는 대규모 네트워크 상의 다중공격에 대비한 분산 침입탐지시스템을 설계 및 구현한다. 더불어, 보안 정책에 맞는 체계적인 탐지 결과를 제공하고 다른 기종의 보안 시스템과의 호환성을 높여주기 위하여 보안 취약성 및 기타 정보 보안에 대한 결과 내용을 MITRE의 CVE(Common Vulnerabilities and Exposures)로 보여준다.

대규모 네트워크 상의 다중공격에 대비한 분산 침입탐지시스템(SPIDERⅡ) 설계 및 구현을 위하여 제 2 장에서 중앙 침입탐지시스템과 분산 침입탐지시스템을 비교 분석하고, 제 3 장에서 분산 침입탐지시스템 관련 기존 연구에 대하여 분석한다. 제 4 장에서는 SPIDERⅡ 설계에 대한 기능 설명과 구현에 대하여 설명하고, 로그의 표준화 CVE의 적용에 대하여 설명한 후, 마지막으로 향후 연구방향 제시와 함께 결론을 맺는다.

2. 중앙 집중 침입탐지시스템과 분산 침입탐지시스템 비교

침입탐지시스템은 탐지 엔진의 위치에 따라 중앙 집중 침입탐지시스템과 분산 침입탐지시스템으로 분류할 수 있다. 중앙 집중 침입탐지시스템은 하나의 고정된 탐지엔진으로부터 모니터링한 결과를 전달받아 분석하는 것을 말한다. 분산 침입탐지시스템은 균일하게 배치되어 있는 탐지엔진의 모니터링 결과를 전달받는다. 이때 탐지엔진 결과 값에 대한 가중치를 결정하고 통합 관리를 통해 최종적인 탐지를 하는 부분이 필요하다.

기존의 침입탐지시스템은 중앙 집중 침입탐지시스템의 형태를 가진다. 이 경우에는 설치 및 운영의 편리성을 제공해 주지만, 중앙 시스템 자체가 침입의 대상이 될 수 있고, 하나의 호스트에서만 감사(audit)를 수행하므로 감사 대상이 제한적일 수 밖에 없다. 또한 수행중인 침입탐지시스템을 설정내용을 변경하거나 기능을 추가·변경하기 위해서 시스템 전체를 재시작 해야 하는 등의 번거로움 있다. 또한, 침입자가 침입탐지시스템의 잘못된 데이터를 수집하도록 조작하거나 침입탐지시스템을 공격할 경우, 올바른 탐지가 되지 않고 침입탐지시스템의 서비스

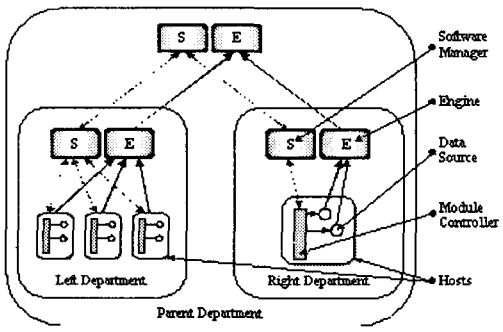
가 중단되거나 오류가 일어나는 등의 문제점이 발생할 수 있다.

분산 침입탐지시스템은 다수의 침입탐지시스템을 설치, 운영해야하는 어려움은 있지만, 침입탐지시스템은 계층적이고 분산된 에이전트의 구조를 가짐으로써 하나의 에이전트가 서비스를 중지해도 다른 에이전트들이 수행을 계속 할 수 있도록 한다. 각 에이전트들이 독립적으로 수행되므로 전체의 시스템을 다시 시작해야 하는 번거로움을 해결할 수 있다. 또한, 각 계층에 있는 에이전트들은 수집한 정보를 간단하게 정리하여 상위 계층으로 전달하므로 특정 부분에서 침입자에 의해 잘못된 데이터가 수집되더라도 다른 에이전트의 감사 데이터들과의 통계를 통해 문제를 발견할 수 있다[3].

3. 기존 분산 침입탐지시스템 분석

(1) GrIDS(Graph Based Intrusion Detection System for Large Networks)

GrIDS는 호스트들의 행위와 호스트들 사이의 트래픽에 대한 정보를 수집하며, 이러한 정보를 행위 그래프로 수집한다.



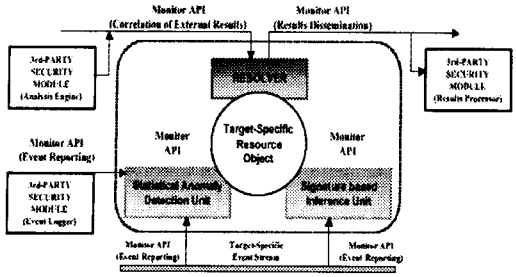
(그림 1) GrIDS 구조

GrIDS의 디자인 목적은 수천 개의 호스트들을 가진 TCP/IP 네트워크상에서의 행위를 분석하는 것이며, GrIDS의 기본 기능은 개인 호스트 위의 침입탐지보다는 대규모의 침입을 탐지하고 분석하는 것이다[4]. (그림 1)은 GrIDS 구조이다.

모든 GrIDS 소프트웨어는 표준화된 인터페이스를 지닌 모듈들로 형성되며, 이러한 모듈들은 각 호스트에 위치한 모듈 제어 프로세스(Module Controller Process)에 의해 수행되거나 멈추고 제어된다. 각 부분은 소프트웨어 관리자(Software Manager)와 그래프 엔진(Graph Engine)으로 구성된다. 소프트웨어 관리자는 계층적이고 분산적인 모듈들을 관리하며, 계층적인 구성은 사용자 인터페이스를 통해 동적으로 재배열될 수 있다. 기타 여러 모듈들 또한 이와 비슷하게 자동적으로 동작 유무가 결정된다. GrIDS 데이터 소스들은 호스트와 네트워크 위의 행위를 모니터링 하는 모듈들이며, 탐지된 행위의 보고를 엔진으로 보낸다. 그래프 엔진은 이러한 데이터 소스 모듈들로부터의 입력을 그래프로 표현하며, 상위 부분으로 이러한 그래프의 묶음을 보낸다. 이를 바탕으로 상위 엔진은 동시적인 결과 그래프를 생성한다. 이러한 구성요소 이외에도 사용자와의 상호작용을 위한 사용자 인터페이스 모듈이 존재하며, 이는 관리 기능 및 경보 디스플레이 기능 등을 수행한다.

(2) EMERALD(Event Monitoring Enabling Responses to Anomalous Live Disturbances)

EMERALD는 대규모의 분산 네트워크에 대한 감시 및 대응에 적합한 구조를 지닌다. 이는 네트워크를 우회하거나 파괴하고자 시도하는 외부 침입 및 자원에 대한 비인가 된 접근을 갖는 행위를 탐지한다[5]. (그림 2)는 EMERALD 구조이다.



(그림 2) EMERALD 구조

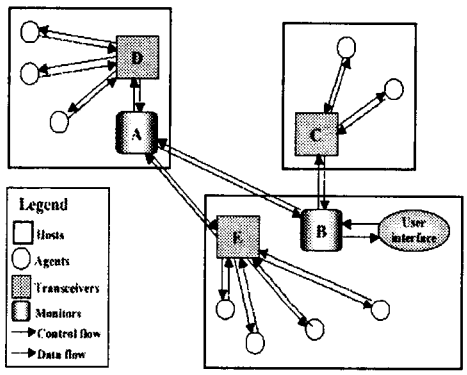
EMERALD의 구조는 모니터라 불리는 분석과 응답 단위로 구성되며, 이는 분석과 응답에 대한 지연을 최소화 할 수 있는 배치 전략을 지닌다.

EMERALD의 분석 범위는 개인 영역의 네트워크 수준에서 시작하여 각 도메인의 네트워크 서비스 및 접근 가능 도메인의 구성 요소들을 분석하며, 이의 수행은 서비스 레이어 모니터(service-layer monitor)와 엔터프라이즈 레이어 모니터(enterprise-layer monitor)로 구분된다. 서비스 레이어 모니터는 도메인 내의 네트워크 서비스 상에서 유발되는 위험 요소를 분석하며, 엔터프라이즈 레이어 모니터는 인터넷 웜과 도메인들 사이의 네트워크 서비스에 대한 반복적인 공격, 단일 도메인에 대한 다중 도메인으로부터의 상호 협력적 공격 등과 같은 침입의 탐지에 초점을 맞춘다.

EMERALD 모니터의 구조는 계층적 분석 형태를 지닐 수 있으며, 이를 위해 매우 작고, 빠르며 일반화된 형태를 갖도록 디자인되었다. (그림 2)는 이러한 EMERALD 모니터의 구조를 보이고 있으며, 시그니쳐 기반 엔진(signature -based engine), 통계 엔진(statistical profiling engine), 대응 단위인 결정자(resolver)인 세 가지 수행 요소로 구성된다. 이 외에도 모니터는 다른 침입탐지 도구와 상호 동작할 수 있는 능력을 높일 수 있도록 다양한 어플리케이션 인터페이스를 통합한다.

(3) AAFID(Autonomous Agent for Intrusion Detection)

AAFID는 하나의 호스트에서 특정 모니터링을 수행하는 소프트웨어 에이전트(agent)를 갖고 있으며 다른 구성 요소와 상관없이 독립적으로 수행한다[6]. 감사 흔적(Audit trail)으로부터 데이터 수집을 한다. 에이전트라는 독립적인 특징으로 간단한 추가 삭제가 가능하여 확장성이 용이하다. 다음 (그림 3)은 AAFID 구조이다.



(그림 3) AAFID 구조

(그림 3)에서 표현된 것같이 에이전트(agent)와 송수신기(transceiver), 모니터(monitor)의 세 가지 구성요소와 사용자 인터페이스로 구성된다. AAFID 시스템은 하나의 네트워크 내에 있는 여러 호스트에 분산될 수 있으며, 각 호스트는 침입과 관련된 이벤트들을 모니터링 하는 에이전트들을 포함한다. 하나의 호스트 내에 있는 모든 에이전트들은 하나의 송수신기로 정보를 전송하며, 송수신기는 각 호스트마다 위치하여 에이전트들을 제어한다. 송수신기는 에이전트에서 수집된 정보에 대한 축약(reduction)을 거치며, 이의 결과를 하나 이상의 모니터로 전송한다. 각 모니터는 여러 송수신기들을 감독하며, 침입탐지 결과를 사용자 인터페이스에 제공

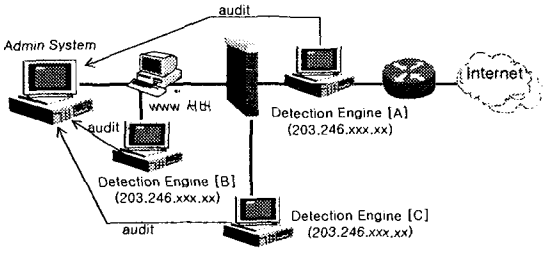
하고 이에 대한 제어 명령을 받는다.

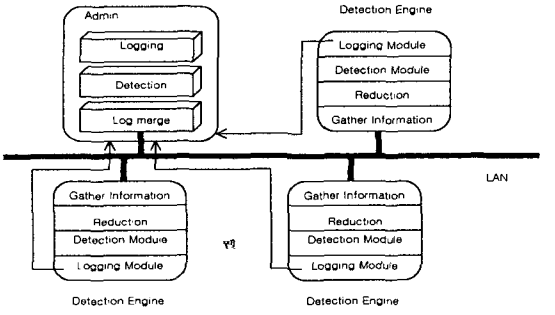
4. 분산 침입탐지시스템(SPIDER II) 모델 설계 및 구현

4.1 SPIDER II 모델 설계

SPIDER II는 Seoul Women's University Project on Intrusion Detection System for Everlasting Renovation의 머리글자이다. SPIDER II는 네트워크 영역을 탐지하여 수집된 패킷을 데이터 감사 자료로 사용한다. 탐지 방법은 알려진 침입 행위나 감사 자료를 이용하여 탐지하는 오용 탐지 모델이다. 대규모 네트워크 상의 다중공격에 대비하여 분산된 탐지엔진에서 전달된 탐지 정보를 통합적으로 분석하는 기능과 이 기종의 보안 시스템과의 호환성을 높이기 위하여 CVE 표준화에 근거한 탐지 결과를 제공한다.

(그림 4)는 SPIDER II 구성 및 전체 흐름 구조를 보여주고 있다.



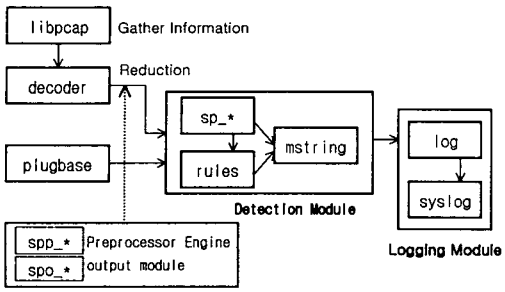


(그림 4) SPIDER II 구성 및 흐름도

SPIDER II는 크게 탐지엔진과 통합 분석 및 관리하는 Admin으로 나뉜다. 대규모 네트워크의 보안이 필요한 요소에 배치된 탐지엔진은 일반적인 침입탐지시스템을 축소한 모듈로써 네트워크 상의 정보를 수집하여 분석한 후, 탐지 내용을 Admin에게 전달한다. Admin은 각 탐지엔진으로부터 전달된 정보들을 각각 분석하는데 이때 일반 공격뿐만 아니라 IP Spoofing, SYN Flooding, SMURF 등의 다중 공격 내용을 탐지하여 사용자에게 결과를 전달한다.

(1) 탐지엔진(Detection Engine)

탐지엔진은 리눅스 기반의 공개 침입탐지시스템인 Snort를 모델로 사용하였다. 탐지엔진은 실시간 트래픽 분석과 IP 네트워크 상에서 패킷 수집이 가능한 장점을 활용하였다. Snort는 패킷 수집 라이브러리인 libpcap[7]에 기반하여 네트워크 스니퍼(sniffer)의 기능을 갖고, 쉽게 정의할 수 있는 침입탐지 룰(rule)들에 일치되는 네트워크 트래픽을 감시, 기록, 경고 할 수 있는 도구이다[8]. (그림 5)는 Snort 흐름도이다.



(그림 5) Snort 흐름도

(2) Admin

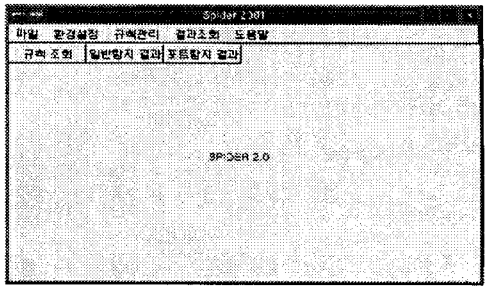
앞서 설명한 탐지엔진이 네트워크 상에 분산 배치되어 이로부터 발생된 탐지 결과를 받으면, 로그에 대한 통합적인 분석이 이루어진다. 따라서, Admin은 직접적인 네트워크 패킷을 수집하지는 않고 메시지 제어 모듈을 통해 탐지엔진의 결과는 전달받는다. 또한, 각각의 탐지엔진의 환경설정과 보안 정책에 따른 규칙 등을 설정하고 조회할 수 있다.

Admin을 통해 각 탐지엔진으로부터 통합, 판정된 탐지 결과에 대한 조회가 가능하다.

4.2 SPIDERII 구현

SPIDERII는 리눅스 Accel 6.1에서 개발하였으며, Admin의 메시지 제어 모듈은 GNU C를 이용하였다. Admin의 사용자 인터페이스는 리눅스 상에서 원활한 GUI 환경을 제작할 수 있는 Gtk+ [9]를 사용하였다.

SPIDERII의 Admin 실행 화면은 다음 (그림 6)과 같다.

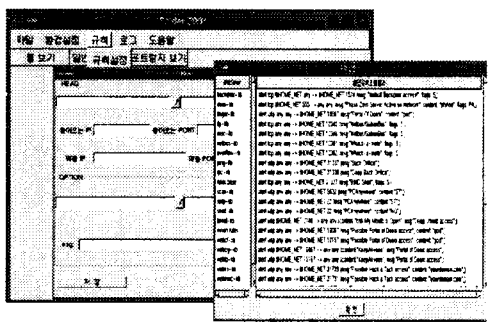


(그림 6) SPIDERII-Admin 실행 화면

Admin은 원격지의 탐지엔진을 실행·종료할 수 있도록 제어가 가능하다. 이를 위해 구현된 모듈은 다음과 같다.

```
void ids_stop( GtkButton *was_clicked, gpointer
user_data )
{
   /* 엔진 종료 */
}
void ids_start( GtkButton *was_clicked, gpointer
user_data )
{
   /* 엔진 실행 */
}
```

분산되어 있는 탐지엔진으로부터 전달된 감사(audit) 데이터들은 Admin에서 통합, 분석하게 된다. 이에 따라 관리자는 보안 정책에 규칙들을 설정할 수 있다. 다음의 (그림 7)은 규칙 관리 기능 및 규칙 조회 기능 화면이다.



(그림 7) 규칙 관리 및 규칙 조회 기능

이처럼 규칙 조회로 선택되어 있는 값을 화면에 출력하기 위하여 구현 내용은 다음과 같다.

```
/* 리스트 상에서 선택된 룰의 내용을 보여준다 */
void combo(GtkEditable *was_changed, gpointer
user_data)
{
  gchar  *srule0  =  gtk_editable_get_chars(
was_changed, 0, -1);
  gchar searchbuf[20] ;
  gchar buffer22[200];
  gchar *spider2[1];
.....
    fgets(searchbuf,sizeof(searchbuf),fp);
    gtk_clist_clear(GTK_CLIST(clist1));

 /* 선택되어 있는 규칙을 불러온다. */
    while(!feof(fq))
    {
    fgets(buffer22, sizeof(buffer22), fq);
    if(!strncmp("alert",  buffer22,  5)  ||
!strncmp("pass",  buffer22,  4)  ||  !strncmp("log",
buffer22, 3))
    {
    spider2[0] = buffer22;
    gtk_clist_append(GTK_CLIST(clist1), spider2);
    }
    }
    fclose(fp);
    fclose(fq);
}
```

Admin은 대규모 네트워크에서 동시 다발적으로 이루어지는 공격에 대한 탐지를 한다. 이를 위해 각 탐지엔진의 감사(audit) 데이터들을 수집하여 보안 정책에 따라 통합, 가공한다.

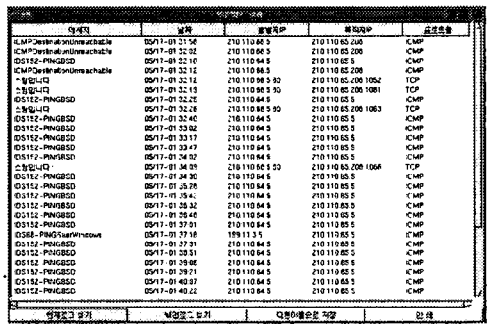
탐지엔지의 감사(audit) 데이터는 다음과 같은 순서로 통합, 분석된다.

① 탐지엔진 A(예, 라우터 후면 배치), 탐지엔진 B(예, 일반 WWW서버), 탐지엔진 C(예, 방화벽 후면 배치)에서 탐지모듈에 의해 탐지된 결과는 Admin에게 전달한다. 각 탐지엔진의 감사(audit) 데이터는 전체 네트워크를 탐지하는 자료로 사용된다.

② Admin은 관리자의 보안 정책이 결정된 규칙을 적용시켜 패턴매칭을 한다.

③ 패턴매칭 결과를 CVE에 적용시켜 관리자에게 네트워크 전반적인 취약점이 있는 곳이나 보안이 요구되는 부분에 대한 리스트를 생성한다.

다음 (그림8)은 위의 과정에 따라 보여지는 통합된 감사(audit) 데이터 결과인 로그를 보여준다.



(그림 8) Admin 통합 분석된 로그

4.3 SPIDER II의 CVE 적용

CVE는 MITRE에서 제공하는 보안 취약성 및 기타 정보 보안 노출에 대하여 표준화된 명칭 목록으로써, 지금까지 공개된 모든 보안 취약성과 보안 노출에 대한 관련 명칭들의 표준을 말한다[10]. CVE는 정보 저장소(보안 도구, 웹 사이트, 데이터베이스, 서비스)에서 CVE 명칭을 사용한 결과물은 다른 정보 저장소와 상호 연결할 수 있도록 한다.

침입탐지시스템과 CVE 호환성에 대한 요구조건은 다음과 같다.

- CVE-Searchable : 사용자로 하여금 CVE 명칭을 사용하여 보안 도구의 관련 작업들을 다룰 수 있도록 해야 한다.

- CVE-Output : 개별 보안 취약성 또는 보안 노출에 대한 내역을 보고할 때, 해당

보고 안에 CVE 명칭이 포함되도록 해야 한다. 보안 도구의 작업들과 관련된 CVE 항목들 모두에 대한 목록을 제공해야 한다.

- CVE 명칭 목록이 포함된 파일을 제공함으로써 사용자가 작업을 명시할 수 있도록 해야 한다.

- 보안 도구의 인터페이스는 사용자로 하여금 개개의 CVE 명칭을 사용하여 작업들을 검색, 선택, 해제 할 수 있도록 해야 한다.

SPIDERⅡ의 탐지엔진 모델인 Snort는 CVE 표준화에 맞게 사용하여 표준화하였다[11]. 따라서 탐지엔진에서 전달되는 결과는 CVE 표준화에 매칭되는 부분을 적용할 수 있다. SPIDERⅡ는 이를 활용하여 로그 결과에 CVE 표준을 적용한 정보를 사용자에게 제공하는 보고 기능을 강화하여 공격 형태 및 취약점 분석이 용이하도록 하였다. 또한 다른 기종간의 탐지 결과의 내용을 표준화하는 것으로 호환성에 이점을 준다.

5. 결 론

본 논문에서는 침입탐지시스템의 탐지엔진을 에이전트화 시킴으로써 대규모 네트워크 상의 다중 공격에 대비한 분산 침입탐지시스템을 설계 및 구현하였다. 여러 플랫폼에 적용할 수 있는 탐지엔진을 이용함으로써 이 기종간의 호환성을 갖는다.

SPIDERⅡ는 네트워크 기반의 침입탐지시스템을 분산 형태로 통합 처리하였다. 그러나, 최근의 공격형태 중, 내부 사용자에 의한 정보보안 침해 사고가 증가되고 있고 네트워크 기반에서 탐지하지 못하는 공격이 증가한다. 이를 위해, 차후에는 호스트 기반의 침입탐지시스템과의 연동이 필요하다. 또한, 침입탐지 뿐만 아니라 효과적인 보안을 위한 실시간 대응책도 함께 연구되어야 한다.

참고문헌

[1] http://www.certcc.or.kr/right11.htm

[2] 이현우, "네트워크 공격기법의 패러다임 변화와 대응방안, Part I : 네트워크 공격기법의 패러다임 변화 v1.0", 정보보호진흥원, November 2000

[3] Eugene H. Spafford, Diego Zamboni, Intrusion detection using autonomous agents, Computer Networks, 547-570, 2000.

[4] S.Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip and D. Zerkle, "GrIDS-A Graph based intrusion detection system for large networks," In proceedings of the 19th National Information Systems Security Conference, 1996.

[5] P. A. Porras and P. G. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," In National Information Systems Security Conference, pp. 353-365, Baltimore, MD, Oct., 1997.

[6] J.S. Balasubramaniyan, J.O. Garcia-Fernandez, E. Spafford, D.Zamboni, An architecture for intrusion detection using autonomous agets, Technical Report 98-05, COAST Laboratory, Purdue University, May 1998.

[7] ftp://ftp.ee.lbl.gov/libpcap.tar.Z

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

[9] http://www.gtk.org

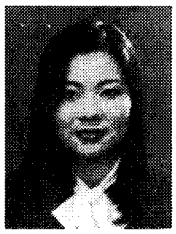
[10] http://www.cve.mitre.org

[11] http://whitehats.com/ids/

최 주 영

1999년 서울여자대학교 컴퓨터학과 학사
2001년 서울여자대학교 컴퓨터학과 석사 과정

최 은 정



1997년 서울여자대학교 컴퓨터학과 학사
2000년 서울여자대학교 컴퓨터학과 석사
2001년 서울여자대학교 정보통신공학부 박사 과정

김 명 주

1986년 서울대학교 컴퓨터공학 공학사
1988년 서울대학교 컴퓨터공학 공학석사
1991년 일본 추쿠바대학 정보전자공학계 연구원
1993년 서울대학교 컴퓨터공학 공학박사
1993-1995년 컴퓨터 신기술 공동연구소 자료실장/특별연구원
1995년- 현재 서울여자대학교 정보통신대학 교수/전산교육원장