Snort 침입탐지 구조를 활용한 디지털 Forensic 응용모델 설계방법

노 시 춘*

요 약

Snort는 Sourcefire. Combining에 의해 개발된 signature, protocol and anomaly-based 탐지방식의 공개 침입탐지 및 침입방지 소프트웨어이다. Snort는 30만의 등록 가입자와 백만의 다운로드를 통해 세계에서 가장 널리 알려진 IDS/IPS 기술이다. Snort는 네트워크 상에서 패킷의 전송과정의 패킷을 검사하여 침입여부를 판별한다. 본 논문 에서는윈도우 환경에서 Snort를 활용한 Forensic 기법을 이용하여 디지털 문서 및 증거 자료에 분석방안을 제안한다. 순서는 Snort를 활용할 경우 Snort 기법과 Forensic 기법에 대해 알아보고 정보보호를 위한 윈도우 환경의 Snort 기법을 활용한 디지털 Forensic 기법 적용한 시스템을 설계해 보고자 한다. 이를위해 IDS가 어떻게 작동하는지, Snort를 어디에 설치하는지, Snort의 요구사항, Snort의 설치방법, Snort의 사양을 위도우 환경에서 적용 하므로서 침입탐지 방법을 제안하고 이를 Forensic 기법에 적용하는 모델을 제시하였다.

A Designing Method of Digital Forensic Snort Application Model

SiChoon Noh*

ABSTRACT

Snort is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. Combining the benefits of signature, protocol and anomaly-based inspection, Snort is the most widely deployed IDS/IPS technology worldwide. With millions of downloads and approximately 300,000 registered users. Snort identifies network indicators by inspecting network packets in transmission. A process on a host's machine usually generates these network indicators. This means whatever the snort signature matches the packet, that same signature must be in memory for some period (possibly micro seconds) of time.Finally, investigate some security issues that you should consider when running a Snort system. Paper coverage includes : How an IDS Works, Where Snort fits, Snort system requirements, Exploring Snort's features, Using Snort on your network, Snort and your network architecture, security considerations with snort under digital forensic windows environment.

Key words : Digital Forensic, Snort Application, Windows Environment

접수일 : 2010년 5월 11일; 채택일 : 2010년 6월 11일

* 남서울대학교 컴퓨터학과

1. 서 론

증가하고 있는 컴퓨터 범죄는 성격상 증거물을 쉽게 확보할 수 없다는 특성을 가지고 있다. 따라서 컴퓨터 범죄는 일반 범죄와 달리 무형의 정보를 증거물로 확보하고, 법적인 효력을 갖는 증거로 채택될 수 있도록 과학적인 수사 기법 및 체계가 제시되어야 한다. Snort는 signature, protocol and anomaly-based 탐지방식의 공개 침입탐지 및 침입방지 소프트웨어로서 가장 잘 알려진 IDS/IPS 기술이다. Snort는 네트워크 상에서 패킷의 전송과정의 패킷을 검사하여 침입여부를 판별한다. 침입탐지 시스템 또는 침입방지 시스템의 패턴탐지의 signature는 대부분 snort를 사용하는데 snort의 패킷 조사는 이미 알려진 공격패턴을 패킷의 데이터에서 패턴매칭을 통해 찾는다. snort는 Streem Matching 방법으로서 패킷이 들어오면 snort의 Rule 첫 번째부터 순차적으로 마지막 Rule까지 패턴매칭을 시도한 후에 빠져 나간다. 따라서 snort 패턴이 많아질수록 탐지 수행 속도는 떨어지기 마련이다. 이러한 문제점에서 좀 더 향상된 시스템 연구가 활발히 진행되고 있다. 이미 보편화 되어있는 컴퓨터 범죄에 대해 수사기법에 대해 Forensic이라는 기법을 적용하여 컴퓨터 범죄에 대해 증거물 처리에 대한 절차와 방법을 확립하고, 이러한 과정을 통해 얻은 합리적인 결과 및 결론은 법정에서 동의 할 수 있도록 해야한다. 본 논문에서는 발달된 기술에 의해 악용 되는 범죄로부터 정보보호 목적의 윈도우 환경의 Snort 기법을 활용 디지털 Forensic 기법을 적용한 시스템을 설계 해 보고자 한다.

2. 관련 연구

2.1 침입탐지 시스템의 처리과정

침입이란 네트워크나 컴퓨터 시스템에 대하여 불법 접근, 불법 정보조작, 시스템 무력화 등을 목적으로 행하는 모든 행위를 말한다. 의도된 행위이든 실수이든 시스템의 보안정책에 위배되는 모든 행위는 침입으로 취급된다. 즉, 시스템 자원에 대한 무결성(Integrity), 기밀성(Confidentiality), 또는 가용성(Availability)을 침해하는 모든 행위를 침입이라고 정의한다. 침입탐지 시스템은 이러한 침입행위를 신속히 탐지하고 대응하는 소프트웨어 또는 하드웨어를 말한다. 침입탐지 시스템의 기술적 처리과정은 다음과 같다[1].

2.1.1 데이터 수집단계

데이터 수집 단계는 침입탐지 시스템 대상 시스템에서 제공하는 시스템 사용 내역, 컴퓨터 통신에 사용되는 패킷 등과 같은 탐지대상으로부터 생성되는 데이터를 수집하는 감사 데이터 수집 단계이다.

2.2.2 데이터 가공 및 축약

데이터 가공 및 축약단계는 수집된 감가 데이터가 침입판정이 가능할 수 있도록 의미 있는 정보를 전환시킨다.

2.2.3 분석 및 침입탐지단계

분석 및 침입탐지 단계에서는 이를 분석하여 침입여부를 판정하는데, 이 단계는 침입탐지 단계의 핵심 단계이며, 시스템의 비정상적인 사용에 대한 탐지를 목적으로 하는지, 시스템의 취약점이나 응용프로그램의 버그를 이용한 침입에 대한 탐지를 목적으로 하는지에 따라 행위탐지기술과 오용탐지기술로 나누어진다.

2.1.4 보고 및 대응

보고 및 대응단계에서는 침입탐지 시스템이 시스템 침입여부를 판정한 뒤 침입으로 판단될 경우 이에 대한 적절한 대응을 자동으로 취하거나, 보안관리자에게 침입사실을 보고하여 보안 관리자에 의한 조치를 취하게 한다[2].

2.2 Snort 기본패턴 분석방법

snort의 패턴매칭 방법에 있어서 snort의 내부 호스트의 운영체제 정보와 IP정보를 이용하여 패턴매칭을 시도하지 않아도 될 부분을 배재함으로서 snort의 처리성능을 보다 빠르게 하기 위하는데 연구의 목적을 두고 있다. Snort 기본패턴 분석방법으로서 대표적인 두가지 방법을 살펴보면 다음과 같다[3].

2.2.1 Boyer-Moore 알고리즘을 이용한 패턴분석

Boyer-Moore 알고리즘은 유명하고 실제로 매우 빠르다. 이 알고리즘은 주어진 문자열이 특정 패턴에 맞는지를 찾기 위한 비교횟수를 줄이기 위해서 휴리스틱(인간의 경험적 지식)을 사용한다. 즉, 문자검색 시 불필요한 비교를 없애기 위해서 키워드에 대한 지식을 사용한다. 이 알고리즘은 탐색을 위해서 문자열과 키워드를 결합하여 사용한다. 키워드는 문자열을 따라서 왼쪽에서 오른쪽으로 체크하게 되고 키워드 마지막 문자에서 시작하여 끝 문자로 체크하게 된다. Boyer-Moore 알고리즘은 규칙들을 규칙별로 구성하고 규칙 내에서 추가 옵션들을 옵션 리스트로 구성하는 방법으로 snort v1.6에서 기본으로 사용하는 방법이다. 규칙의 구성은 RTN에 표시되며, 옵션의 구성은 OTN에 표시된다. Boyer-Moore 알고리즘은 단일의 키워드에 대하여 많은 문자열의 정확한 스트링 일치를 위하여 설계되었다. 이 알고리즘은 이러한 동작의 수행 시 매우 효율적인 반면에, 다중의 키워드의 유사성에 있어서는 이득이 없다. 따라서 실패할 것이 분명하더라도 패킷의 내용거사를 함으로써 처리성능이 떨어지게 된다.

2.2.2 Aho-corasick 알고리즘을 이용한 패턴분석

Snort v2.0 이하에서 사용되던 Boyer-Moore 알고리즘의 구조를 변경하여 패킷 매칭 속도를 크게 향상시킨 방법이다. 이 방법은 패킷 내용 검사를 비교해야 할 키워드 별로 매번 진행해야 하는 번거로움이 있는 점을 개선한 방법이다. 이 방법은 한 RTN에 소속된 OTN들 내에 있는 검사해야 할 키워드들에 대하여 패킷 내용 검사를 한꺼번에 진행한다. 이 때 사용되는 문자열 비교 알고리즘은 Boyer-Moore 알고리즘을 다중 키워드 검색으로 확장시킨 AC-BM (Aho-corasick Boyer-Moore)방법이다. 이 방법에서는 패턴이 일치하지 않는 옵션을 갖는 모든 OTN을 조사에서 배제함으로서 snort 성능에 큰 영향을 미쳤다.

2.3 snort 시스템 구성요소

snort는 스니퍼(Sniffer), 전처리기(preprocessor), 탐지엔진(detection engine), 출력모듈(outprocessor)의 4가지 구성요소로 이뤄져 있다[4].

2.3.1 스니퍼(sniffer)

스니퍼 또는 패킷 스니퍼는 네트워크에서 데이터를 수집하는 소프트웨어나 하드웨어를 의미한다. snort는 원래 스니퍼 용도로 만든 프로그램이기 Eons에 기본적으로 네트워크를 돌아다니는 패킷을 잡아낼 수 있다.

snort는 libcap 라이브러리를 통해 네트워크 트래픽을 캡쳐한다. 디코드 엔진은 패킷을 분석하여 링크계층 프로토콜용 패킷을 채운다. 그 다음 디코드 엔진은 TCP와 UDP 포트와 같은 상위계층 프로토콜 패킷구조를 채운다. 따라서 전처리기의 전단계로서 패킷을 재조합한다. 대표적인 도구로서 tcpdump를 들 수 있다.

2.3.2 전처리기(preprocessor)

스니퍼에서 캡처한 네트워크 패킷은 전처리기로 이동한다. snort의 전처리기는 패킷을 탐지 엔진에서 비교하기 전에 사전 처리 작업을 해 주는 역할을 하며, Plug-In 방식으로 되어 있어 추가와 삭제가 쉽기 때문에 훨씬 유연한 소프트웨어를 운

영할 수 있다.

2.3.3 탐지 엔진(Detection Engine)

탐지 엔진은 snort의 핵심 모듈로서 패킷과 규칙을 비교하여, 패킷에 해당하는 규칙이 있을 경우 경고를 발생한다. snort의 규칙은 snort에서 가장 중요한 부분으로 snort에는 나름대로 규칙 문법이 있다. 규칙 문법은 프로토콜의 종류, 컨텐트, 길이, 헤더, 기타 여러 요소(버퍼 오버플로우를 정의하기 위한 쓰레기 문자 등)을 포함하고 있다. 규칙을 잘 설정하면 snort를 자신의 환경에 맞게 커스터마이징 하는 것이 가능하다.

2.3.4 출력 모듈(outprocessor)

snort가 발생시킨 경고는 출력 모듈로 전송된다. 출력 모듈도 Plug-In 구조로 되어 있으며, 현재 snort는 로그 파일, 네트워크 연결, UNIX 소켓 또는 윈도우 팝업(SMB), 또는 SNMP 트랩, MySQL과 Postgres와 같은 SQL 데이터베이스로 경고를 보낼 수 있다. 그리고 snort의 경고를 분석하기 위한 snort Snarf와 Swatch, ACID등의 여러 툴이 나와 있다.

3. Snort의 침입탐지 구조설계

3.1 Snort에서 제공되는 룰

Snort는 Chain이라는 구조를 가진다. 이 구조는 여러곳에서 발견할 수 있는데, 먼저, 탐지할 룰 그룹이 그렇고, 각종 플러그인들을 함수 포인터로 관리하는 구조체의 메모리상의 구조도 같지는 않지만, 이와 매우 유사하다.

먼저 탐지할 룰을 받게 되면, Chain Header를 통해서, Source IP, Port, Destination IP, Port를 확인한다. 탐지하고자 하는 내용과 일치할 경우, Chain Option으로 가게 되고, 아니면, 다음 헤더로 넘어가게 된다. Chain Option에서 나머지 정보들을 확인을 하고, 일치할 경우, 침입탐지로 간주하고, 로그를 생성한다[3].

```
alert tcp $EXTERNAL_NET any → $HOME_NET 80 (msg : "WEB-CGI-Websendmail CGI access attempt" ; flags : PA; content : "webs …
alert tcp $EXTERNAL_NET any → $HOME_NET 80(msg : "IDS226-CVE-1999-0172-CGI-formmail"; flags : PA; content : "/for …….
```

〈그림 1〉 Snort에서 사용하는 룰 형태

기본적으로 Snort에서 제공되는 룰은 *.rules이란 이름을 가지며, 소스와 같은 디렉토리에 존재한다. 위의 룰에서 보면, alert는 로그를 남길 형태를 말하며, tcp는 프로토콜의 형태를 말한다. "$EXTERNAL_NET any → $HOME_NET 80"이 값은 Chain Header의 값이 된다. 즉, 탐지하고자 하는 IP 주소일 경우에만 탐지를 하고, 아니라면, 탐지를 하지 않겠다는 말이 된다. 마지막으로 "(msg : "WEB-CGI-Websendmail CGI access attempt"; flags : PA; content : " webs …" 부분이 바로 Chain Option이 된다. 두 번째 룰을 설명하면, "msg"는 룰에 대한 제목쯤으로 해석할 수 있다. "flags"는 헤더에 있는 플래그를 말하는 것으로, PSH, ACK 플래그가 켜져 있다는 말이다. 마지막으로 "content"는 패킷의 내용을 말한다. 그러므로, 이 패킷은 TCP 프로토콜을 사용하고, $EXTERNAL_NET에서 $HOME_NET의 80번 포트로 접근하는 패킷 중에서, PSH, ACK 플래그를 사용하고, 패킷의 내용이 " : "/for ……"와 같다면, 탐지가 되는 것이다.

$EXTERNAL_NET, $HOME_NET의 값은 snort.conf에서 설정하는 값이다.

3.2 디지털 Forensic 기법

디지털 Forensic은 정보기기에 내장된 디지털 자료를 근거로 삼아 그 정보기기를 매개체로 하여 발생한 어떤 행위의 사실 관계를 규명하고 증명하

는 것을 말한다. Forensic은 크게 디스크 Forensic, 네트워크 Forensic, 인터넷 Forensic으로 3가지로 나누어 볼 수 있다. 위에서 정의한 Forensic 시스템에 따라 본 논문에서는 아래의 구성 시스템을 이용하여 Forensic 시스템을 설계 해 보고자 한다. 네트워크 Forensic에 쓰이는 IDS 시스템과 Firewall 시스템, 그리고 Internet Forensic에 쓰이는 Web Server를 활용하여 설계한다. 각 시스템에 따라 로그의 형태가 다르게 나오므로, 하나의 데이터로 통합 시키는 과정이 필요하다. 3가지의 로그의 형태 중에 공통 속성을 적용하여 하나로 수집하는 시스템을 구현하며, 이를 이용하여 조건에 의해 필요한 데이터를 볼 수 있게 된다. 디지털 Forensic의 의의를 보면 “법정의”, “공개토론이나 변론에 사용되는”, “수사와 법정에서의 증거 또는 사실관계를 확정하기 위하여 사용하는 과학이나 기술에 관한”, “범죄와 관련된 증거물을 과학적으로 조사하여 정보를 찾아내기 위한” 이라는 의미를 말한다. 또한 수사의 근본적인 특성에 따라 법적으로 유효한 전자적 증거의 확보를 목표로 과학적 지식과 기술을 활용하여 전자적 증거를 수집하고 분석하는 제반 행위를 의미한다[5].

3.3 디지털 Forensic 증거자료 수집 및 로그

위와 같은 특성으로 디지털 범죄의 증거 자료에 대해서는 다음과 같이 증거를 수집하고 분석하는 도구를 사용한다.

디스크 이미징과 디스크 복제도구, 데이터 복구 및 분석도구, 데이터 무결성 도구, 암호 복구 도구, 데이터 조사 도구, 증거 수집 도구, 데이터베이스 분석 도구, 네트워크 및 인터넷 분석도구. Forensic 시스템에서 사용하는 Log Data는 일반적인 Log의 집합으로 만든다. 일반적으로 각각의 시스템에서 나오는 로그 데이터를 하나의 로그 데이터에 모은다. 이렇게 모인 데이터를 Log Data라고 한다. 이때, 각각의 로그 데이터에 포함되는 형식에 따라 동일한 포맷의 DB를 구축하여 하나의 DB를 만든다. 이렇게 만들어진 Log Data_DB에서 Forensic 기법에 의해 필요한 데이터들은 최종 Forensic 시스템에서 입력된 조건에 의해 구현된 UI를 통해 보이게 된다. 조건이 적용되어 보이게 되는 Data를 UI Data라고 정의하며, 전체 시스템에서 모아진 전체 데이터 중에 UI Data는 일부이다[4].

4. 시스템 설계

4.1 Snort_log

시스템 설계는 Windows XP 환경에서 Snort, MySQL, Honeynet Security Console을 연동하여 Snort경고를 실시간 모니터링 한다. 이때 채집된 Snort_log 파일을 Forensic 기법에 응용하는 방법이다. 모니터링 중 파일을 전송 받으면 안전한 파일인지 확인하고 이를 Forensic 기법으로 활용한다[5].

4.1.1 관련 프로그램 설명

• WinPcap : http://www.wincap.org

WinPcap이란 libpcap(packat capture library : 네트워크 패킷을 저장하고 보내는데 널리 쓰이는 네트워크 프로그래밍 API)의 Win32 포트이다. WinPcap은 크게 두가지 요소로 구성되어 있다. 대부분의 유닉스 커널에서 제공하는 BPF 기능에 기반한 커널 수준 패킷 필터 드라이브, 윈도우용 고수준 프로그래밍 라이브러리, libpcap 패킷 캡쳐 드라이버는 유닉스 커널의 BPF와 유사하게, 윈도우 9x, 윈도우 NT, 그리고 윈도우 2000에 로(raw) 네트워크 패킷을 캡쳐하고 보내는 기능을 추가하는 장치 드라이버이다. Packet.dll은 WinPcap의 컴포넌트로, BPF 드라이버 함수로 바로 접근할 수 있는 API를 제공한다. WinPcap은 또한 libpcap과 호환 가능한 일련의 함수를 엑스포트(export)하며, 이에 따라 네트워크 하드웨어 및 운영 체제와 독립적으로 패킷을 캡쳐하는 고수준 함수를 제공한

다. libpcap API가 Win32에서 사용할 수 있다는 것은 수많은 유닉스 네트워크 유틸리티가 윈도우에 이식되었다는 뜻이다. 여기에 소개된 모든 툴은 WinPcap을 사용한다[6].

• MySQL : http://www.mysql.org

Mysql 데이터베이스는 다른 상용 데이터베이스와 비교해 안정성과 기타 여러 문제에 크게 떨어지지 않으면서 공개되어진 데이터베이스이다. 다양한 함수와 여러 기능면에서 상용 데이터베이스가 가지지 못한 점을 가진 것도 많이 있다. 그리고 공개된 다른 데이터베이스와 비교 할 때 사용자 또한 넓다. Mysql은 C, C++, Java, python, tck/tk, ODBC도 제공하며 다양한 웹서버와의 연결도 용이하다. ODBC는 Ms 윈도우(win95, win98, NT) 등을 제공한다. 현존하는 윈도우의 대부분을 지원 하며 설치 또한 대부분의 OS를 지원 한다. Mysql을 개인적인 용도로 사용할 경우 무료이며 상업적인 목적으로 사용할 경우 mysql 개발팀에 조금의 후원을 하면 가능하다.

• Snort win32 : http://snort.org

Snort의 개발자인 Marty Roesch에 의하면 “Snort는 실시간 트래픽 분석과 IP 네트워크 상에서 패킷 로깅이 가능한 가벼운(lightweight) 네트워크 침입탐지 시스템”이라고 한다. Snort는 패킷 수집 라이브러리인 libpcap에 기반한 네트워크 스니퍼인데, 쉽게 정의할 수 있는 침입탐지 rule들에 일치되는 네트워크 트래픽을 감시하고 기록하고 경고할 수 있는 도구이다. Snort는 프로토콜 분석, 내용 검색/매칭을 수행할 수 있으며 오버플로우, Stealth 포트스캔, CGI 공격, SMB 탐색, OS 확인 시도 등의 다양한 공격과 스캔을 탐지할 수 있다. 또한 이러한 탐지 rule들은 보안 Community를 통해 지속적으로 업데이트되고, 본인이 쉽게 rule을 작성하여 추가할 수 있으므로 최신 공격에 적응이 쉽다.

• Net Framework : http://www.microsoft.com

• Honeynet security Console : http://www.activeworx. org

Honeynet security Console은 개인용 네트워크나 허니넷(honeynet)을 감시하고 분석하는 툴로써 snort, TCPDump, 방화벽, syslog, sebek로그등과 연동되는 강력한 기능을 제공한다. 이벤트를 분석하여 해킹공격에 대응하기 위한 도구이다.

4.1.2 관련 프로그램 설치

- WinPcap : 범용 패킷 캡쳐 라이브러리인 libpcap의 window 버전으로 snort가 패킷 캡쳐를 수행하기 위하여 필요하다(WinPcap 4.0.1설치).
- MySQL Win 32설치 : DBMS로 Snort의 로그나 이벤트 저장하기 위해 연동시키다(MySQL Server 4.1로 설치).
- Snort :
- Honeynet Security Console : 2.6버전 설치

4.1.3 Snort 기능 설정

- Snort설정 :

파일 : snort.conf

위치 : 설치디렉토리/etc

설정 내용 :

1. # like this :
   var HOME_NET 10.1.2.0/24
2. # Path to your rules files(this can be a relative path)
   # Note for Windows users : You are advised to make this an absolute path,
   # such as : c:\snort\rules
   var RULE_PATH c:\snort\rules
   var PREPROC_RULE_PATH \preproc_rules
3. # Include classification and priority settings
   # Note for Windows users : You are advised to make this an absolute path,
   # such as : c:\snort\etc\classification.config

include c:\snort\etc\classification.config
#
Include reference systems
Note for Windows users : You are advised to make this an absolute path,
such as : c:\snort\etc\reference.config
include c:\snort\etc\reference.config

4. dynamicpreprocessor file c:\snort\lib\snort_dynamicpreprocessor\sf_dcerpc.dll
dynamicpreprocessor file c:\snort\lib\snort_dynamicpreprocessor\sf_dns.dll
dynamicpreprocessor file c:\snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll dynamicpreprocessor file c:\snort\lib\snort_dynamicpreprocessor\sf_smtp.dll
dynamicpreprocessor file c:\snort\lib\snort_dynamicpreprocessor\sf_ssh.dll
Load a dynamic engine from the install path
(same as command line option -- dynamic-engine-lib)
dynamicengine c:\snort\lib\snort_dynamicengine\sf_engine.dll

5. # Include all relevant rulesets here
The following rulesets are disabled by default :
web-attacks, backdoor, shellcode, policy, porn, info, icmp-info, virus,
chat, multimedia, and p2p
These rules are either site policy specific or require tuning in order to not
generate false positive alerts in most enviornments.
Please read the specific include file for more information and
README.alert_order for how rule ordering affects how alerts are triggered.

4.1.4 Snort 설정 확인

● Snort룰 설치

www.snort.org에서 Snort탐지 룰을 받아 압축을 푼다. Rules 디렉터리 이하의 파일은 Snort 설치 디렉터리의 rules디렉터리에 복사하고, doc\signature 디렉터리 이하의 파일은 snort설치 디렉터리의 signature 디렉터리에 복사한다.

- Snort와 MySQL연동

실행 명령 창에서 snort가 사용할 DB생성.

명령 : mysqladmin-u root-p create snort

비밀번호 입력.

내부적으로 사용하는 DB테이블 등의 스키마 생성

명령 : mysql-D snort-u root-p < create_mysql

비밀번호 입력

● Snort설정 확인.

본 시스템 설계에 사용되는 Snort 시스템을 구동하여 로그 데이터를 얻는다. 공격이 발생 하였을 때, 공격이 탐지된 포트에 대해 그래프로 보여주는 시스템으로 되어있다. (그림 2)와 같은 로그를 남겨 침입에 대한 분석을 상세히 할 수 있도록 한다.

```
[**] TCP SYN flooding [**] 05-09 11 : 16 : 34, 327160
211.106.28.203 : 780 → 255.255.255.255 : 780 TCP TTL : 226 TOS : 0X0, ID : 51324 ***sc* Seq : 0xA1325410 Ack : 0x0 Win : 0xA01
[**] SNMP request tcp [**] 05-14 11 : 54 : 36, 245185
211.106.28.84 : 0 → 211.106.28.82 : 161 TCP TTL : 152 TOS : 0x0, ID : 249612 **d **se Seq : 0x42891 Ack : 0x0 Win : 0x4920
```

(그림 2) Snort에서 나오는 로그의 형태

(그림 2)에서 보여주는 Snort 로그에서는 발생한 공격의 유형과 시간정보, 그리고 버전정보를 제공한다. 탐지 정책을 준 것에 의거하여 정책을 기준으로 정책에 맞지 않는 공격 발생시 기록을 남기게 된다. IP주소를 확인 할 수 있으며, 확인된 IP주소를 기반으로 즉각적인 패킷 분석이 가능하다. 위와같

은 로그의 정보는 침해에 대한 직접적인 정보를 제공 할 수 있다.

4.2 Forensic 응용방법

4.2.1 IPcop_log

(그림 3)에서 보여주는 IPcop 로그 데이터에는 시간정보와, 버전정보, 그리고 서비스 번호와 포트 번호 및 IP주소가 나온다. 일반적으로 Firewall에서는 탐지정책과 통제 정책을 동시에 정하기 때문에, 탐지 뿐 아니라 통제된 정책 위반 시도에 대해서도 로그 기록을 남긴다.

```
time = "29Mar2007 23 : 10 : 01" action = ctl orig = netfw2 i/f_dir = inbound i/f_name = daemon sys_msgs = "started sending log to localhost"
time = "29Mar2007 23 : 19 : 03" action = accept orig = netfw2 i/f_dir = inbound i/f_name = hme1 proto = tcp src = 192.168.210.67 dst = 211.117.62.98 service = 3000 s_port = 4198 len = 48 rule = 13 has_accounting = 0 xlatesrc = 210.108.137.146 xlatedst = 211.117.62.98 xlatesport = 27023 xlatedport = 3000
time = "29Mar2007 0 : 07 : 27" action = drop orig = 210.108.137.12 i/f_dir = inbound i/f_name = hme0 proto = tcp src = 211.202.12.9 dst = 210.108.137.132 service = 23312
```

(그림 3) IPcop에서 나오는 로그 데이터

4.2.2 Web Server_log

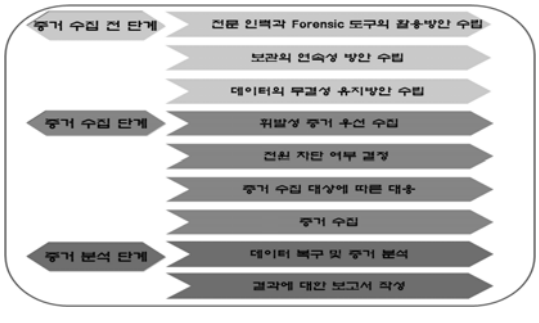
(그림 4)에서는 Web Server에서 남기는 로그를 확인 할 수 있다. 시간과 소스 IP, 목적지 IP, 포트 번호 등의 정보가 남는다. 또한 GET/POST 메시지를 통해 가져온 것인지, 전송한 것인지 알 수 있으며, 송수신 한 파일의 이름이 남게 된다.

```
211.178.31.139, -, 2005-09-21, 0 : 00 : 06, W3SVC186, NURIWEB31, 211.39.149.31, 0, 457, 180, 304, 0, GET, /inner_img/b_qbn02.gif, -,
211.178.31.139, -, 2005-09-21, 0 : 00 : 06, W3SVC186, NURIWEB31, 211.39.149.31, 0, 457, 180, 304, 0, GET, /inner_img/b_qbn03.gif, -,
211.178.31.139, -, 2005-09-21, 0 : 00 : 06, W3SVC186, NURIWEB31, 211.39.149.31, 0, 457, 180, 304, 0, GET, /inner_img/b_qbn04.gif, -,
```

〈그림 4〉 Web Server에서 나오는 로그 데이터

4.2.3 Forensic 기법 적용순서

위에서 알아본 시스템 및 로그 데이터를 기준으로 Forensic 기법을 적용하는 시스템을 설계 제안이다. 일반적인 네트워크 장치를 활용하여 Forensic 시스템을 설계 할때, Forensic 수사를 통한 증거 확보를 위해 갖추는 절차를 정리 해 보면 (그림 5)와 같다. 크게 3개의 절차로 나누어, 증거 수집 전과 증거 수집, 그리고 수집된 증거를 분석하는 단계로 구분이 가능하다. (그림 5)에서 제시하는 순서에 따라 설계된 시스템을 통해 증거를 확보할 수 있다. 각각의 시스템에서 발생하는 로그 데이터를 LogData_DB에 수집하여 하나의 DB화하여 Forensic 로그를 만들고. Forensic UI를 통해 원하는 어떤 조건에 의해 Forensic 로그에 있는 내용들 중에서 필요한 부분만 확인 할 수 있는 시스템이다. IDS와 Firewall인 Snort와 IPcop에서 나오는 로그를 하나의 통합된 Forensic DB에 수집하고 Forensic 시스템을 수행 할 때 이 DB를 참조하여 침해 상황 및 경로를 파악할 수 있다. Forensic_UI에는 사용자가 조건을 입력할 수 있는 기능도 명시해야 한다. 최종적으로 사용자는 Forensic UI를 통해 모아진 로그 데이터의 조합으로 이루어진 정보를 보게 될 것이다.



(그림 5) Forensic 기법을 적용 증거 확보 절차

5. 결 론

본 논문에서는 디지털 환경에서 정보보호를 위

해 정보침해가 발생 하였을때 적용할 수 있는 시스템으로서 Snort 시스템을 구동한 Forensic 기법 적용 시스템을 설계해 보았다. 네트워크 기술에 사용되는 모듈을 활용하여 Forensic 기법을 적용한 사례이며 방화벽과 침입탐지 시스템, 그리고 우리 가장 가까이 접할 수 있는 웹 서비스를 통해 남게 되는 로그 데이터를 사용하여, 정보 침해사고 발생시 디지털 수사를 진행할 수 있는 시스템을 설계한 것이다. 윈도우 환경의 Snort 기법을 활용하여 디지털 Forensic 기법을 적용한 시스템 설계 방법론을 활용할 경우 향후 정보보호 운용 현장에서 응용할 수 있으리라 기대한다.

참 고 문 헌

[1] Dieter Gollmann, “Computer Security”, John Wiley and Sons, 2005.

[2] 고병수, 박영신, 최용락, “컴퓨터 포렌식스를 지원하는 보안 감사/추적 모듈설계”, 한국 컴퓨터정보학회 논문지, 제9권, 제1호, pp. 79-86, 2004.

[3] 이형우, 이상진, 임종인, “컴퓨터 포렌식스 기술”, 한국정보보호학회지, 제12권, 제5호, pp. 8-16, 2002.

[4] 박종성, 최운호, 문종섭, 손태식, “자동화된 침해사고 대응시스템에서의 네트웍 포렌식 정보에 대한 정의”, 정보보호학회 논문지, 제14권, 제4호, pp. 149-162, 2004.

[5] 정정기, 박대우, “로그 히스토리 분석을 사용한 웹 포렌식 알고리즘 연구”, 한국 컴퓨터정보학회 논문지, 제12권 제1호, pp. 123-132, 2007.

[6] 정익래, 홍도원, 정교일, “디지털 포렌식 기술 및 동향”, 전자통신동향분석 제11권, 제1호, pp. 97-104, 2007.

[7] Information Technology Laboratory-Computer Forensics Tool Testing Program, http://www.cftt.nist.gov 2008.

[8] Sichoon, Noh, “MSPI(Multi-Spectral Protection Infrastructure) System for Optimal Network Security”, IEEE Computer Society, 2008.

[9] Paul Woosnam, “10 hottest Technologies”, Teleco mmunications, 2003.

노 시 춘

1987년 고려대학교 경영정보학(석사)

2005년 경기대학교 정보보호기술(박사)

2002년 KT 시스템보안부장

2004년 KT 충청전산국장

2005년~현재 남서울대학교 컴퓨터학과 교수

관심분야 : 차세대통신, 컴퓨터네트워크, 정보보호