논문 2009-4-15

웹서비스-해킹패턴 인지시스템에 관한 연구

A Study on the Web Service-Hacking Pattern Recognition System

임채균*, 안다선*, 김규호**, 이기영***

Chae-Gyun Lim, Da-Seon Ahn, Kyu-Ho Kim, Ki-Young Lee

요 약 최근 Web 2.0이 이슈가 되면서 웹 기반으로 제공할 수 있는 서비스에 대한 기술이 빠르게 발전하고 있다. 또한, 웹상의 다양한 플랫폼과 SDK(Software Development Kit)도 여러 분야에서 제공되고 있어서 웹 서비스의 사용자가 급격히 증가하는 추세를 보이고 있다. 그러나 보안 시스템의 발전은 그 속도를 따라가지 못하고 있으며, 이러한 약점을 노려서 웹 서비스에서 제공되는 수많은 정보를 불법적인 방법으로 취득하려는 시도가 다량 발생하고 있는 실정이다. 따라서, 본 논문에서는 웹 서비스에 대한 다양한 불법적인 접근을 확인하기 위하여 접근정보를 분석하였다. 또한, 분석한 결과와 기존의 해킹방법으로부터 분류한 해킹패턴과의 매칭을 통해 해킹을 인지함으로써 위험 사실을 경고하는 시스템을 설계하고 구현하였다.

Abstract Recently, web 2.0 is becoming issue can provide a web based services and the technology is developing rapidly. In addition a variety of platforms on the web and SDK(Software Development Kit) is available from various aspects, web services, a trend that is rapidly growing user. But the development of security systems cannot follow the speed and These weaknesses are provided in the Web service by using illegal methods of information acquisition. Therefore, in this paper, we analyzed information for checking a variety of illegal access on Web services. Also, we designed and implemented Web Service-Hacking Pattern Recognition System for which warns the dangerous fact that as recognized hacking through comparisons the hacking patterns which have classified from the hacking method of existing system.

Key Words : Web 2.0, Hackicng Pattern, Web Service, Pattern Matching

Ⅰ. 서론

최근에 해킹사고 신고건수가 2008년 15,940건으로 2007년 21,732건에 비해 26.7% 감소하였다. 이 신고건수 감소는 2005년 이후 꾸준히 이어지고 있다. 해킹 신고가 줄어드는 주요 원인은 스팸 릴레이나 단순침입시도와 같이 불특정 다수를 대상으로 하는 공격이 감소한 결과로 볼 수 있다. 그러나 해킹사고 신고건수가 감소함에도 불구하고 특정 대상을 목표로 하는 해킹공격은 감소하지 않은 것으로 보인다.

특히 2008년에는 금품 요구와 같은 크고 작은 협박성 공격이 다수 발생하였으며 글로벌 경기 침체와 맞물려 2009년에도 지속될 것으로 전망되어 특히 보안이 취약하고, 인터넷 인프라 내성도 약한 중소 인터넷기업을 대상으로 지속적으로 발생할 것으로 예상된다[1]. 홈페이지 해킹 공격은 홈페이지 가입자의 개인정보와 기업의 보안을 필요로 하는 정보에도 밀접한 관계가 있어 금전적인 피해 뿐 아니라 각종 범죄에 이용되는 등 2차적 피해까지도 예상되고 있다. 하지만 개인과 기업에 피해를 주는 해킹을 미리 알고 예방할 수 있다면 그에 따른 금전적인 피해와 부가적인 피해까지도 막을 수 있다[2].

*준회원, 을지대학교 의료산업학부
**정회원, 을지대학교 의료산업학부
***종신회원, 을지대학교 의료산업학부 (교신저자)
접수일자 2009.6.30, 수정완료 2009.8.3

이에 본 논문에서는 웹 서비스 해킹의 패턴을 분류하고 서버에 접근하는 해킹 패턴을 인지하여 관리자에게 해킹위험을 경고함으로써 피해를 막을 수 있는 시스템을 제안한다. 본 논문은 2장에서 침입 탐지 시스템과 패턴 인식에 관해 다루고 3장과 4장에서 시스템 설계 및 구현에 대하여 설명한다. 5장에서는 시스템 성능평가에 대하여 기술하며 6장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1 침입 탐지 시스템(Intrusion Detection System, IDS)

침입 탐지 시스템은 탐지 대상, 침입 탐지 기술, 침입에 대한 대처 방법에 따라 분류 할 수 있다. 먼저 탐지 대상에 따른 분류를 사용할 경우 네트워크 기반 침입 탐지 시스템, 호스트 기반 침입 탐지 시스템, 하이브리드 기반 침입탐지시스템으로 분류할 수 있다. 네트워크 기반 침입 탐지 시스템은 탐지 대상이 네트워크에 존재하고 패킷 분석을 통해 탐지한다. 그리고 전체 네트워크에 대한 탐지가 가능하고 호스트 기반 침입 탐지 시스템에서 탐지 못하는 공격도 탐지가 가능하며 기존 네트워크 구조의 변경 없이 설치가 가능하다. 하지만 속도가 빠른 네트워크에 적용이 어려우며 실제 침입에 사용될 패킷인지 확실히 단정하기는 어려움이 있다.

호스트 기반 침입 탐지 시스템은 탐지 대상이 호스트 내부에 존재하고 사용자 활동 기록, 프로그램 로그 등의 감사 기록 분석을 통해 탐지를 한다. 또한 네트워크 기반 침입 탐지 시스템에서 탐지할 수 없는 공격도 탐지가능하며 속도가 빠른 네트워크에서도 적용가능하다. 하지만 개별 호스트 마다 설치해야 하는 비용 부담과 설치된 시스템에 성능 저하 유발과 침입 탐지 시스템 자체에 대한 공격이 가능하다.

하이브리드 기반 침입 탐지시스템은 네트워크와 호스트 모두에 대한 침입 탐지가 가능하고 네트워크 기반과 호스트 기반의 침입 탐지 시스템의 장점을 모두 수용한다[3]. 하지만 설치가 복잡하고 설치된 시스템의 성능 저하 유발 가능성이 있다. 본 시스템에는 그림1.과 같은 구조의 IDS 모습을 따른다.

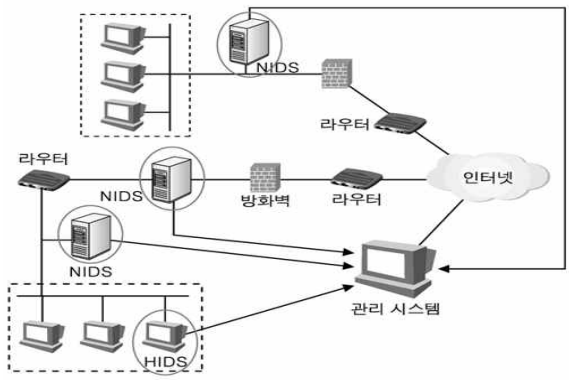

그림 1. IDS 구조
Fig. 1. IDS Architecture

탐지 방법에 따른 분류에는 비정상행위 탐지 침입 시스템과 오용 탐지 침입 탐지 시스템이 있다. 비정상행위 탐지 침입 탐지 시스템은 사용자 프로파일 기반 비정상행위 탐지를 하고 프로파일의 정상행위 패턴과 사용자 행위를 비교, 사용자의 향후 행위를 예측하여 공격을 탐지한다. 이 방법은 새로운 공격 기법의 탐지가 가능하지만 데이터 분석 및 프로파일 구축비용 부담이 크고 구현이 난해하다.

오용 탐지 침입 탐지 시스템은 알려진 취약성에 기반하여 공격을 탐지한다. 공격을 예측하는 비정상행위 탐지 기법과는 다르게 실제 발생하는 공격을 탐지하므로 패턴 매칭등의 기술이 적용된다. 구현 비용이 저렴하나 새로운 공격의 탐지가 어렵고 최신 공격에 대한 계속적인 패턴 구축을 필요로 한다[4].

침입 대응 방법에 따른 분류에는 침입 탐지 후 관리자에게 침해 사실과 대응 방법만 통보하는 수동적 대응 침입 탐지 시스템과 침입 탐지 후 관리자에게 침해 사실을 통보하고 직접 대응 절차를 수행하는 능동적 대응 침입 탐지 시스템이 있다.

2.2 패턴 인식(Pattern Matching)

패턴이란 개별 객체의 특색 혹은 특징을 의미하며, 특징을 모아놓은 집합으로 정의할 수 있다. 즉, 특징과 패턴은 유사한 개념이지만 특징들이 모여서 패턴을 이루게 된다고 볼 수 있다. 패턴인식에서 패턴은 관측된 특징 벡터 x와 이 특징 벡터가 속한 클래스 ω로 이루어진 변수쌍 {x,ω}으로 표현된다. 여기서 클래스(class)는 카테고리(category), 그룹(group), 라벨(label)이라고도 하는데 같은 소스로부터 발생된 공통된 속성 혹은 특징 집합을 공

유하는 패턴 집합을 의미한다[5].

패턴 인식은 데이터로부터 중요한 특징이나 속성을 추출하여 입력 데이터를 식별할 수 있는 부류로 분류(Classification) 으로 정의될 수 있다[6][7].

III. 시스템 설계

3.1 시스템 구조

본 시스템은 웹서비스를 제공하고 있는 서버에 대한 접근을 처리하는 시스템이므로, 클라이언트 - 서버 구조를 포함하고 있으며 서버 측에서 일련의 작업들이 수행된다. 어떠한 접근인지 판별하기 위해서 서버 측에 있는 모듈이 주기적으로 로그를 분석 및 감시한다.

기본적으로는 웹 서비스를 제공하는 서버에 클라이언트의 Request가 있을 경우에 로그분석 모듈로 클라이언트의 Log 정보가 전달된다. 그 후에 분석결과를 패턴인지 모듈로 전달하여 기존의 구조화 데이터와 패턴매칭을 진행하고, 그 결과를 경고 모듈에서 입력받아서 처리하는 구조이다. 이러한 작동과정에 대한 구조는 그림 2.에 나타나있다.

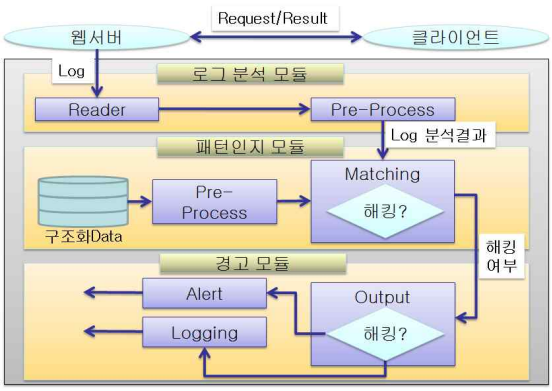

그림 2. 시스템 구조
Fig. 2. System Architecture

3.2 세부 모듈

3.2.1 로그 분석 모듈

먼저, 로그분석 모듈은 그림 3.과 같이 서버에서 로그를 입력받은 후에, 전처리를 통해 데이터를 가공하는 모듈이다. 이 모듈에서는 입력으로 로그 정보를 받아야하는데, 이는 서버로부터 제공받는다.

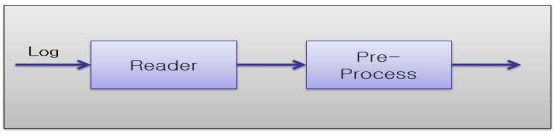

그림 3. 로그 분석 모듈
Fig. 3. Log Analysis Module

서버로부터 입력받는 로그는 기본적으로 아파치 서버의 Common Log File Format을 따라서, 다음의 구조를 가지고 있다.

```
LogFormat "%h %l %u %t \"%r\" %>s %b
        \"%{Referer}i\" \"%{User-agent}i\""
```

이에 대한 구조를 조금 더 상세히 설명해줄 수 있는 샘플 데이터이다.

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700]
   "GET /apache_pb.gif HTTP/1.0" 200 2326
   "http://www.example.com/start.html"
   "Mozilla/4.08 [en] (Win98; I ;Nav)"
```

“127.0.0.1”은 클라이언트의 접속 위치를 나타내고, “-”은 원격의 로그인 ID(지원하지 않는 경우에 “-”으로 표기)이다. "frank"는 사용자 아이디를 의미하며, “[10/Oct/2000:1

3:55:36 -0700]" 은 서버에 요청하는 시간이다. 그 후에는 순서대로 요청의 첫 번째 라인 내용, 상태 코드, 헤더를 포함한 전송량(bytes), 클라이언트의 이전 위치, 클라이언트의 브라우저 정보이다.

3.2.2 패턴 인지 모듈

다음으로 패턴인지 모듈은 그림 4.와 같이 로그분석 모듈에서 가공되어 출력된 결과를 입력으로 받아서, 가지고 있던 구조화 데이터와 매칭하는 모듈이다.

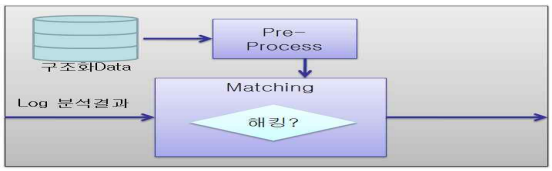

그림 4. 패턴 인지 모듈
Fig. 4. Pattern Recognition Module

패턴인지 모듈에서는 기존의 다양한 해킹패턴에 대한 데이터들을 구조화하여 저장하고 있다. 이 구조화 데이터의 포맷은 그림 5.와 같은 형태를 가지고 있다.

그림 5. 구조화 데이터 포맷
Fig. 5. Structed Data Format

이 구조화 데이터는 다음의 경고 모듈에 의해서, 지속적으로 수정, 보완되어 더욱 효율적으로 갱신된다.

3.2.3 경고 모듈

마지막으로 경고 모듈에서는 그림 6.과 같이 실제 해킹 여부를 전달받는다. 만약에 해킹이라면 관리자에게 경고를 출력하여 적절한 조치를 요구하고, 그렇지 않은 경우에는 정보를 저장하여 구조화 데이터를 보강한다.

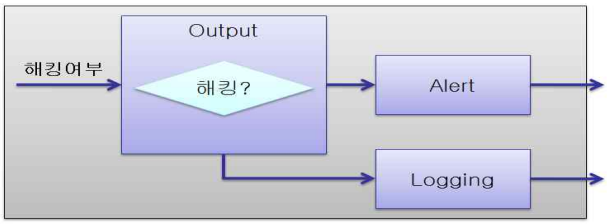

그림 6. 경고 모듈
Fig. 6. Alert Module

Ⅳ. 시스템 구현

본 시스템의 구현은 크게 로그분석 모듈, 패턴인지 모듈, 경고 모듈 세 가지의 구현으로 이루어진다. 각 모듈에 대한 개요는 앞에서 언급하였으며, 이 장에서는 각각의 모듈의 실질적인 구현에 대하여 기술한다.

4.1 시뮬레이션 환경 및 가상 시나리오

본 시스템은 기본적으로 Windows XP SP3를 운영체제의 환경에서 시뮬레이션을 하는데 서버, 클라이언트 모두 동일한 운영체제를 사용한다. 주요 구현 언어는 C++과 Python이고, 로컬 네트워크 환경에서 APM6(Apache, PHP ver 5.2.5, MySQL)를 구축하여 시뮬레이션 하고자 한다. 가상 시나리오를 구성해보면 그림 7.과 같다.

1. 기존에 운영되고 있는 웹서비스에 인증되지 않은 접근 시도 발생
2. 로그분석 모듈에서 분석된 로그결과에서 비인증 접근 및 행동 확인
3. 패턴인지 모듈에서 저장된 구조화데이터와 비교 및 분석
4. 만약 해킹이라고 분석된 결과를 경고 모듈이 입력받는 경우에는 관리자에게 경고를 발생
5. 그렇지 않으면 정보를 로그로 저장하고, 구조화데이터에 반영하여 성능 향상

그림 7. 가상 시나리오
Fig. 7. Assumption Scenario

4.2 로그 분석 알고리즘

로그분석 모듈에서는 서버의 로그를 입력받아서 전처리를 한 결과를 출력하는 기능을 구현한다. 기본적인 알고리즘은 그림 8.과 같다.

```
// 로그 정보 로딩
log ← Reading in common log file;
// 전처리
for in log do
  result ← Distribute log as CSV;
return result;  // 결과 리턴
```

그림 8. 로그 분석 알고리즘
Fig. 8. Log Analysis Algorithm

4.3 패턴 인지 알고리즘

패턴인지 모듈에서는 로그 분석 결과와 기존의 구조화 데이터를 매칭하여, 해킹여부를 출력하는 기능을 구현한다. 알고리즘은 그림 9.와 같다.

```
// 로그 분석결과 로딩
result_log ← Reading in data that log analysis module
returned CSV type;
// 구조화 데이터 로딩
struct ← Reading in data that structured database;
// 분석결과와 구조화 데이터 매칭
result ← match(result_log, struct);
return result;  // 결과 리턴
```

그림 9. 패턴 인지 알고리즘
Fig. 9. Pattern Recognition Algorithm

4.4 경고 알고리즘

경고 모듈은 해킹여부에 따라 관리자에게 경고하거나, 관련 정보를 저장하는 기능을 구현한다. 알고리즘은 그림10.과 같다.

```
// 해킹여부 입력
flag ← Reading in result that pattern matching module
matched log and structured data;
// 해킹이면
if(flag == 'hacking')
// 관리자에게 경고
  Alert to administrator;
// 그렇지 않으면
else
// 정보저장(구조화 데이터 보완)
  Logged to structured database;
```

그림 10. 경고 알고리즘
Fig. 10. Alert Algorithm

V. 성능 평가

이 장에서는 본 시스템을 구현한 환경에서 임의의 해킹시도의 인지율에 대한 성능 평가 결과를 살펴본다. 성능 평가를 위해 사용된 시스템의 하드웨어 사양은 Intel(R) Core(TM)2 Duo CPU T9300 2.50GHz, 4GB RAM이며, 운영체제는 Windows XP SP3를 사용하였고, 성능 평가를 위해 사용된 해킹시도의 수는 1000개, 2000개, 3000개, 4000개, 5000개 이었다.

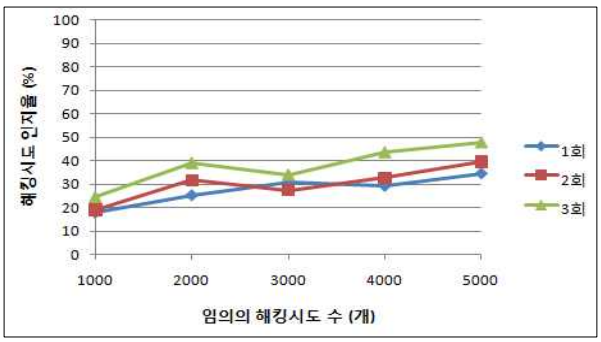

그림 11. 임의의 해킹시도에 대한 인지율
Fig. 11. Recognition Ratio on Hacking Attacks

그림 11.은 본 시스템에서 초기화 후 실험 횟수를 1~3회로 다르게 하여, 임의의 해킹시도의 인지율에 대한 성능 평가 결과를 보여준다. 실험을 진행하면서 구조화 데이터가 보완되는 것을 감안하여, 각 실험마다 초기화를 수행한 후에 1~3회의 실험을 진행하고 각 실험 횟수에 따라 측정결과를 구분하였다.

그림 11.의 실험 결과를 통해 실험 횟수가 1회인 경우의 평균 인지율은 약 28%, 2회인 경우는 약 30%, 3회인 경우는 약 38%로 전체 실험 결과의 평균 인지율은 약 32%를 보여준다. 이러한 해킹패턴 인지율의 성능이 떨어지는 이유는 해킹패턴에 대한 많은 형태를 구조화하지 못한 이유가 있다. 그러므로 추후 시스템의 해킹패턴들을 좀 더 보강하여 시스템을 구축한다면 해킹패턴 인지율이 높게 상승할 것으로 판단한다.

VI. 결론

Web 2.0의 등장 이후로 급격하게 변하고 있는 현황에서, 웹 애플리케이션에 대한 해킹대안은 부족한 실정이다. 이러한 상황에서 본 시스템은 더욱 다양하고 교묘해지는 해킹의 패턴들로부터 발생되는 문제점을 해결하는 방안으로 적용될 수 있다. 향후 본 시스템을 다른 장비들과 연동하여 물리적인 경로를 제어하는 방향으로도 발전시킬 수 있으며, 각종 응용 프로그램에 대한 논리적인 네트워크 경로를 제어하고 감시하는 방향으로도 활용할 예정이다.

또한, 본 시스템에 대한 해킹패턴 인지율을 향상시키기 위한 방안 연구와 기존 시스템과의 효율성 비교 및 분석의 경우도 향후에 진행할 예정이다.

참고문헌

[1] 해킹사고 신고 건수, http://www.index.go.kr/
[2] Gunter Ollmann, “Hacking as a service”, GuIBM Internet Security Systems, 2008.
[3] Ajith Abrahama, Ravi Jain, Johnson Thomas and Sang Yong Han, “D-SCIDS: Distributed Soft Computing Intrusion Detection System”, Journal of Network and Computer Applications Vol. 30, pp. 81 - 98, 2007.
[4] Rafeeq Ur Rehman, “Intrusion Detection

Systems with Snort : Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID", Prentice Hall PTR, 2003.
[5] 한학용, "패턴인식 개론: MATLAB 실습을 통한 입체적 학습", 한빛 미디어, 2005.
[6] Pattern Recognition, http://www.aistudy.com/
[7] Srivatsan Laxman and P.S.Sastry, "A survey of temporal data mining", Sādhanā Vol. 31, pp. 173 - 198, 2006.

저자 소개

임 채 균(준회원)

• 2007년~현재 을지대학교 의료산업학부 의료전산학전공 학생
<주관심분야 : u-Healthcare, 유비쿼터스, LBS, GIS, USN, 영상처리 등>

안 다 선(준회원)

• 2007년~현재 을지대학교 의료산업학부 의료전산학전공 학생
<주관심분야 : u-Healthcare, 인공지능, 모바일 DB 등>

김 규 호(정회원)

• 제 9 권 3호 참조
• 1992년~현재 : 을지대학교 의료산업학부 부교수
<관심분야 : USN, U-Healthcare, 임베디드시스템 등>

이 기 영(종신회원): 교신저자

• 제 9 권 3호 참조
• 2009년~현재 한국인터넷방송통신TV학회 협동이사
• 1991년~현재 을지대학교 의료산업학부 부교수
<관심분야 : 공간 데이타베이스, GIS, LBS, USN, 텔레매틱스 등>