베이지언을 이용한 웹 어플리케이션 공격의 실시간 탐지에 관한 연구

모정훈, 임종인, 문종섭
고려대학교 정보경영공학 전문대학원
{picomax, jilim, jsmoon}@kore.ac.kr

A Study of Realtime Detection of Web Application Attack using Bayesian

Jeong-hoon Mo, Jongin Lim, Jongsub Moon
Graduate School of Informaiton Management and Security

요약

인터넷 사용의 대중화에는 웹 서비스의 힘이 컸다고 할 수 있다. 지금까지도 웹 기반의 서비스가 점차 확대되고 있고 이에 따라 웹 공격과 웹 보안이 이슈가 되고 있다. 웹 서비스를 이용하는 어플리케이션은 기존 보안도구를 통한 분석 작업과 모니터링에 관리자의 개입이 많이 요구되었고, 자동화된 방법 중의 하나인 로그를 이용한 분석 방법들은 실시간으로 확인하고 대응할 수 없는 단점이 있다. 본 논문에서는 기존의 웹 공격 탐지 방법과 시각화 방법들의 개선사항들을 제안한다.

1. 서론

최근 DoS(DDoS)와 같이 네트워크를 마비시키는 공격은 크게 줄어들고 있는 추세인 반면, OWASP 에서 정의하는 웹 어플리케이션 공격의 10대 취약점을 이용한 상위 레벨의 공격이 급증하고 있다[1]. 이러한 웹 어플리케이션 공격의 탐지와 분석에 있어서는 관리자의 많은 개입이 요구되고, 실시간으로 공격의 탐지가 어렵다는 단점이 있다[2]. 전통적인 웹 어플리케이션의 침입탐지 시스템은 알려진 공격 시그너쳐를 탐지해내는 오용탐지 방식의 시스템을 이용함으로써 새로운 공격 패턴이나 이상 행위에 대한 즉각적인 대응을 할 수 없었다.

따라서 본 연구에서는 새로운 공격 패턴이나 사용자의 이상 행위에 대한 탐지 시스템을 제시하고 관리자에게 정보 전달의 효율성과 신속성 등의 측면에서 보안도구의 운용과 네트워크 상황을 인지하기 위한 시각화 방법을 제시한다.

본 논문의 2장에서는 웹 어플리케이션의 취약성과 기존의 침입 탐지 관련 시각화 연구를 소개한다. 3장에서는 웹 어플리케이션의 이상 행위의 실시간 탐지 기법과 관리자를 위한 효과적인 시각화에 대한 내용을 제안하고, 끝으로 4장에서 결론을 맺도록 한다.

2. 관련 연구

가. 웹 어플리케이션 취약점

본 절에서는 OWASP에서 제시한 대표적인 웹 어플리케이션 공격 기법 10가지를 살펴본다.

1) 입력 값 검증부재

웹 어플리케이션에서 주고받는 URL, 쿼리 문자열, HTTP 헤더, 쿠키, *HTML* 폼 인자, *HTML* 히든 필드 등 모든 HTTP 요청을 변조할 수 있으며 이를 통해 사이트의 보안 메커니즘을 우회할 수 있다.

2) 취약한 접근통제

웹상에서 사용자 인증은 전형적으로 사용자의 ID와 패스워드를 필요로 한다. 자신의 패스워드를 다른 사람이 기억 등 기타 관계된 기능을 포함하여 신용 관리 기능의 결함이 있을 수 있다.

3) 취약한 인증 및 세션관리

HTTP는 개별 사용자의 요청을 지속적으로 추적하기 위한 세션 기능을 제공하지 않는다. 웹 어플리케이션이 자체적으로 생성하여 사용하는 세션 토큰을 가로채 다른 사용자를 가장하여 접속할 수 있다.

4) 크로스사이트-스크립팅 취약점

웹 어플리케이션의 입력 값에 Javascript와 같은 스크립트를 삽입하여 비정상적인 조작이 일어나도록 유도할 수 있다. 이는 웹 어플리케이션 서버 측을 공격하는 것이 아니라 웹 서비스를 이용하는 사용자를 공격 대상으로 하고 있다는 점이 다른 공격과 구별되는 특징이다.

5) 버퍼 오버 플로우

일반적인 응용프로그램 보안취약점의 하나로써 프로그램에서 버퍼의 한계를 점검하지 않고 작성된 코드부분을 이용하여 악의적인 코드로 프로그램의 정상적인 동작을 변경하거나 프로그램이 다운되도록 하는 공격 방법이다. 웹 서버 또는 웹 어플리케이션에 존재할 수 있다.

6) 삽입 취약점

웹 응용프로그램을 통해 악의적인 코드를 다른 시스템으로 전달할 수 있다. OS의 시스템 콜 등을 쉘 명령어를 경유하여 후미의 데이터베이스에 SQL을 경유하여 호출될 수 있다..

7) 부적절한 에러처리

고의적으로 에러를 유발시켜 주요 정보가 feedback되어 돌아오는 것을 역이용 할 수 있다. 이는 공격자에게 웹 사이트의 데이터베이스 덤프, stack traces 등 다양한 보안 문제점을 알려줄 수 있다.

8) 취약한 정보저장 방식

웹 어플리케이션은 데이터베이스나 파일 시스템 상에 패스워드, 신용카드 정보, 계좌정보 등 민감한 정보를 저장할 수 있다.

9) 서비스 거부 공격

한 사용자가 시스템의 리소스를 독점하거나 모두 사용, 또는 파괴함으로써 시스템에 과도한 부하를 일으켜 데이터나 자원을 정당한 사용자가 적절한 대기 시간 내에 사용하는 것을 방해할 수 있다.

10) 부적절한 환경설정

불필요한 디폴트 파일이나 백업 및 예제 파일, 부적절한 파일 및 디렉터리 권한 설정, 디폴트 패스워드를 사용하는 계정 등 웹 어플리케이션 설치 당시의 기본 값을 그대로 사용할 경우 표적이 될 수 있다.

나. 기존의 시각화 기법 연구

이 장에서는 네트워크 및 웹 어플리케이션의 이상행위에 대한 기존 시각화 연구들에 대해서 알아본다. John.R.Goodall은 침입탐지의 절차를 감시(Monitoring), 분석(Analysis), 대응(Response)의 3단계로 구분하였다[7]. 감시과정은 네트워크 트래픽이나 감시 대상을 지속적으로 주시하고 이벤트 발생 시 다음 단계로 전달하는 과정이다. 분석단계는 감시과정으로부터 전달된 이벤트 기준을 정의하고, 처리하는 방법에 대한 정의 단계이다. 마지막 대응단계에서는 분석과정에서 발생한 정보를 사용자에게 어떻게 전달하고, 대응할 것인지 정의하는 단계이다. 분석과정에의 정보를 효과적으로 전달하기 위한 방법으로는 시각화를 통한 정보 전달 방법이 가장 효과적이고, 신속한 방법이 될 수 있다. 이는 생태학적 구조상 사람은 직관적으로 시각적인 패턴을 가장 잘 인식하기 때문이다[8]. 다음 [표 1]은 네트워크 및 웹상의 이상행위 시각화에 대한 기존 연구들의 특징이다.

[표 1] 기존의 시각화 도구들의 특징 비교

도 구	특 징	단 점
TNV [1]	시간에 기반한 이상행위의 감시	장기간의 트래픽 수집이 요구됨
VisAlert [2]	동심원을 이용한 오용탐지의 시각화	Snort 룰에 대한 시각화에 대한 국소적인 단점
PVF [3]	x, y 축을 이용한 Port, IP의 시각화	1차원 데이터만으로 시각화를 구성
WebViz [4]	웹 페이지들과 사용자들의 이동 경로를 시각화	효율적이지 못한 레이아웃 구조
SAD [5]	웹 로그 분석을 통한 웹 서버의 이상행위 감시	웹 접근 로그 기반의 시스템

TNV를 제외한 나머지 도구들은 모두 웹 로그와 같은 감사파일에 기반한 시각화 도구들이다. 이러한 로그파일에기반한 시스템은 로그파일의 보안상의 문제와 로그가 생성되는 시간 동안의 시간이 경과한 다음에 시각화가 갱신되는 단점이 있다[7]. 특히 TNV는 장기간 적어도 몇 시간 동안의 사용자들의 세션을 수집해야만 이상 세션을 분석 및 확인할 수 있다.

WebViZ는 웹 접근 로그 상에서 각 페이지 간의 연결 관계를 분석하여, 페이지들의 좌표를 화면에 보여주는 형태이다. 이는 정보의 단순 시각화로 인해, 일반적인 행위의 시각화는 하지 못할 뿐만 아니라 랜덤하게 좌표를 생성하고 비효율적인 레이아웃 구조를 가지고 있다.

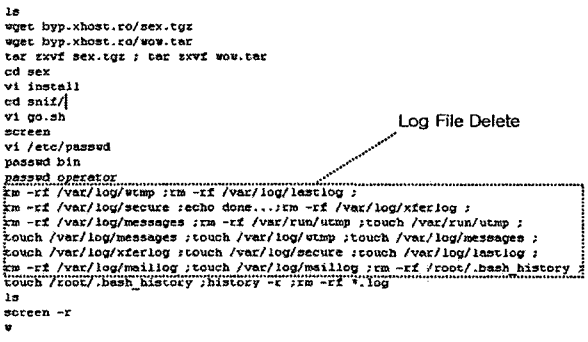
PNV는 0~65535 사이의 내부 네트워크와 외부 네트워크 포트번호를 시각화하고 있다. 이는 특정한 이상행위의 판단을 위한 데이터 마이닝과 같은 기법을 사용하지 않은 정보의 시각화 자체에 그친다는 단점이 있다.

SAD/SAD Viewer는 웹 서버 로그를 기반으로 일반적인 웹 페이지에 대한 시퀀스 프로파일을 만들고, 각 클라이언트의 행위를 생성 된 프로파일과 비교하여 클라이언트가 정상적인 사용자인지 비정상 사용자인지를 판단하여 시각화하는 이상 탐지 시각화 시스템이다. SAD의 경우 웹서버의 로그파일로부터 침입탐지와 시각화를 구현함에 따라 실시간 침입탐지 시스템이라고는 할 수 없다. 뿐만 아니라 시각화 레이아웃 측면에서 동심원의 빈도수에 대한 이벤트를 갱신할 때, 정적인 구조를 가지고 있다.

3. 웹 공격에 대한 실시간 이상 탐지 방법

가. 웹 로그 분석 방식의 대체

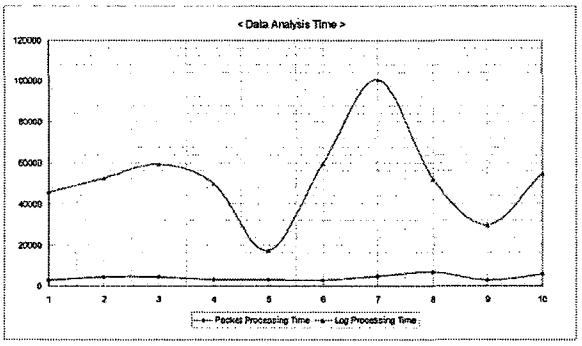
웹 어플리케이션에 대한 정보를 얻는 방법을 크게 4가지가 있다. 첫째, 웹 브라우저를 수정하여 개인의 정보를 얻는 방법과, 둘째, 웹 서버의 웹 로그를 통해서 얻는 방법, 셋째, 웹 프록시 서버의 로그를 통해 얻는 방법, 마지막으로 네트워크의 트래픽(패킷)을 모니터링 하는 방법이다[6]. 각각의 방법은 나름대로 장점이 있으나 기록되는 정보의 구체적인 내용에 많은 제한이 있다. 웹 브라우저를 수정하는 방식은 사용자의 샘플집단에게 동의를 얻어야 하며, 웹 브라우저의 소스를 수정해야 한다. 웹 서버의 로그를 이용하는 방식은 로그 정보가 해당 웹 서버에 한정된 로그형식으로 인해 필요한 정보를 얻지 못할 수도 있으며, 대상 웹 서버의 로그파일의 공격 및 삭제 등으로 인해 정확한 정보를 얻지 못할 수도 있다.



[그림 1] 공격자의 로그 삭제 행위

또한 분석 시간 측면에서도 많은 문제점이 존재한다. 웜과 Bot 등의 자가 복제 코드들이 수 초 만에 네트워크 전체로 퍼져나가는 것을 고려해볼 때, 길게는 수분까지 걸리는 웹 로그 방식의 처리 시스템은 문제가 있다고 할 수 있다. 이러한 속도차를 비교해보기 위해 다음과 같은 실험을 진행하였다. IIS의 웹서버 로그 업데이트시간과 로그로부터 시각화 프로그램의 갱신속도의 합과 패킷을 모니터링 하는 방법으

로부터의 시각화 갱신 속도를 비교하였다. 로그 분석은 클라이언트의 요청에 대한 웹 로그 갱신 시간과 300개의 로그로부터 파싱하는 과정의 시간 합이며, 패킷 분석은 300개의 패킷을 캡처하여 파싱하는 과정의 시간이다. [그림 2]에서 X축은 실험 회수이며, Y축은 수행속도의 시간(millisecond)이다. [그림 2]의 실험 결과 수행시간 면에서 시간차이는 최대 95637msec, 최소 14236msec를 보였다.



[그림 2] 웹 로그 기반 방식과 패킷 모니터링 기반 방식의 분석시간 비교

이러한 엄청난 양의 데이터와 지속적인 모니터링, 웹 서버 로그파일로 부터 필요한 데이터를 뽑아내고 처리하는 것은 쉽지 않다. 이러한 요구사항은 관련된 데이터를 실시간으로 처리하여 가공해야 할 필요성을 발생 시킨다.

나. HTTP 프로토콜의 구조와 헤더분석

HTTP 프로토콜은 요청/응답 방식을 이용하여 동작한다[9]. 즉, GET, HEAD, POST 와 같이 프로토콜 기능에 대해 서비스 요구를 하면 데이터 송수신을 위한 TCP 연결이 만들어지고, 서버가 응답을 보내어 데이터 전송을 끝내면 연결이 끊어지게 되는 것이다. 이러한 내용은 HTTP 패킷의 헤더에 내용을 포함하고 있다. 요청헤더 정보로부터는 Method, Version, Uri, Reffer 등을 얻을 수 있고, 응답헤더로부터는 요청에 대한 Status Code, Server Type 등을 얻을 수 있다.

HTTP 프로토콜의 실시간 분석을 위해서는 세 가지 문제점을 해결할 수 있어야 한다. 첫째, HTTP 헤더의 어떤 형태를 사용할 것인가와 둘째, TCP/IP 프로토콜 상의 지연으로 인한 결합문제, 마지막으로 복수 클라이언트 구조를 어떻게 처리할 것인가에 대한 문제이다[6]. 첫 번째 문제를 해결하기 위해, W3C의 웹서버 로그 포맷 구성 형태를 구성하였다[10]. 즉, 패킷의 아스키코드를 파싱하여, 헤더의 각 필드 구성 내용을 가지는 자료구조(Log) 형태를 만들어 분석에 이용하였다. 이를 통해 HTTP헤더의 정보를 로컬상의 일반적인 웹서버 로그 파일을 읽어오는 것과 동일한 결과를 가져오도록 하였다. 두 번째 문제를 해결하기 위해서 요청 패킷에 대해서 큐를 생성하여, 패킷을 버퍼에 저장하도록 하였다. 마지막으로 세 번째 문제를 해결하기 위해 해시테이블의 구조를 이용한다. 고유키 값을 클라이언트의 IP주소와 포트번호, 서버의 IP주소와 포트번호를 이용하여 생성하였다.

다. 이상행위 탐지 기법

이 절에서는 기존의 발생한 이벤트와 발생하지 않은 이벤트에 따라 새로운 이벤트의 발생 확률을 추정하는 베이지언 알고리즘을 이용한 알려지지 않은 이상행위 탐지기법을 사용한다[11]. 기존의 Uri필드의 정보와 Reffer필드 쌍을 이용하여, 새로운 HTTP 패킷 발생 시, 다음과 같은 방식으로 이상 확률 값을 계산한다. 이상 탐지 스코어는 에이전트의 요청발생시 마다 각 에이전트별로 IP와 포트 번호를 결합하여 디렉터리를 생성하여, 내부에 link.dat과 symbol.dat 2개의 파일DB를 사용한다.

[표 2] 이상 스코어 변수 정의

변수	정의
X	Reffer 값
N	Reffer 필드의 총 빈도 수
N_i	Symbol 의 빈도 수
K	Symbol 의 경우의 수
L	발생 가능한 전체 Reffer 수
α	의사 결정 값

(※ Symbol : {Reffer, Uri} 쌍을 지칭)

link.dat 파일은 Reffer, Uri, 필드로 구성되며, symbol.dat 파일은 Reffer, N, K 필드로 구성된다. 새로운 경로가 발생 시마다 추가되어, 이상 값을 수식 (1) 과 같이 계산하게 된다. 이상 탐지 스코어는 에이전트의 요청발생시 마다 각 에이전트별로 IP와 포트 번호를 결합하여 디렉터리를 생성하여, 내부에 2개의 파일DB를 생성한다. link.dat과 symbol.dat 2개의 파일DB를 사용하였다. link.dat 파일은 Reffer, Uri, 필드로 구성되며, symbol.dat 파일은 Reffer, N, K 필드로 구성된다. 새로운 경로가 발생 시마다 추가되어, 이상 값을 계산하게 된다.

$$P(X=i)=\frac{C(N_i+\alpha)}{(K\alpha+N)} \quad (C=\frac{N}{N+L-K}) \qquad \text{수식(1)}$$

이상 탐지 스코어는 에이전트의 요청발생시 마다 각 에이전트별로 IP와 포트 번호를 결합하여 디렉터리를 생성하여, 내부에 link.dat과 symbol.dat 2개의 파일DB를 사용하였다. link.dat 파일은 Reffer, Uri, 필드로 구성되며, symbol.dat 파일은 Reffer, N, K 필드로 구성된다. 새로운 경로가 발생 시마다 추가되어, 이상 값을 수식 (1)과 같이 계산하게 된다. 이상 탐지 스코어는 에이전트의 요청발생시 마다 각 에이전트별로 IP와 포트 번호를 결합하여 디렉터리를 생성하여, 내부에 2개의 파일DB를 생성한다. link.dat과 symbol.dat 2개의 파일DB를 사용하였다. link.dat 파일은 Reffer, Uri, 필드로 구성되며, symbol.dat 파일은 Reffer, N, K 필드로 구성된다. 새로운 경로가 발생 시마다 추가되어, 이상 값을 계산하게 된다.

라. 시각화 레이아웃

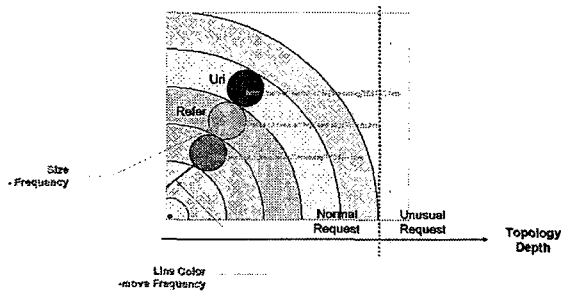
동심원 레이아웃 구조는 웹 페이지들의 디렉토리 구조를 효과적으로 표현할 수 있는 레이아웃 구조이다[12]. 동심원 레이아웃 구조는 SAD에서 시도하였던 구조였으나 정적인 갱신과 이벤트의 중복 시각화로 인하여 감시 시간이 길어질수록 효율성이 크게 떨어지게 되었다. 이러한 문제점을 해결하기 위해 본 논문에서 새로운 시각화 레이아웃 구조를 제시한다. [표 3]은 제안하는 동심원 레이아웃 구조의 시각화 기준이다. Division Circle 영역에서는 요청하는 웹 페이지의 해당 디

렉토리 구조의 깊이를 나타내는 레이아웃 역할을 한다. Request Circle 영역에서는 동심원의 색깔, 크기 등으로 이상행위를 시각화하는 역할을 한다. Link Line 영역은 각 페이지간의 연결 관계와 빈도수를 시각화 한다.

〔표 3〕 시각화 표준 기준 정의

	변 수	설 명
Division Circle	radius	웹 페이지 디렉토리의 깊이를 나타냄 (원의 외부는 존재하지 않은 요청)
Request Circle	color	정상과 비정상 여부와 최근의 요청을 표현 (이상탐지 알고리즘을 이용)
	size	HTTP Request에 대한 빈도수를 표현
	degree	시간에 따른 순서를 표현 (상대좌표를 이용해 이벤트의 중복을 회피)
Link Line	color	Reffer와 Uri 사이의 정상관계 여부를 표현
	width	Reffer와 Uri 쌍의 빈도수를 표현

다음의 [그림 3]는 위의 [표 3]의 정의 기준을 바탕으로 시각화를 표현한 인터페이스의 예시화면이다.



〔그림 3〕 동심원 인터페이스 시각화 구현 예시

4. 결론

본 논문에서는 실시간 탐지를 위해 TCP/IP 프로토콜 상의 지연 복수 클라이언트 구조의 처리 등과 같은 고려해야할 문제점에 대한 해결책을 제시하였다. 기존의 웹 접근 로그 분석을 통해 시각화로 표현한 시스템을 개선하여, 웹서버의 HTTP 패킷의 요청/응답 헤더를 이용한 탐지 기법을 제안하였다. 이상 행위에 대한 탐지는 베이지언 추론 기법을 사용하여, 확률에 따른 이상 스코어로써 공격을 탐지하고 실시간 시각화 방법에 대해 제안하였다. 또한, 제안한 시각화 도구에 대한 예시 화면을 보였다. 이러한 시스템은 보안 도구 관리자들에게 보다 빠르고, 이해하기 편리한 정보 전달 도구로써 구현될 수 있을 것이다. 하지만 포털사이트와 같은 많은 에이전트를 가질 수 있는 웹서버와 어플리케이션에 대한 적용 방안은 추가적인 실험과 연구가 필요하다.

참고문헌

[1] Eugene Lebanidze, "Securing Enterprise Web Applications at the Source: An Application Security Perspective", OWASP Tech. Report, 2004

[2] J. McHugh, "Intrusion and intrusion detection," International Journal of Information Security, vol. 1, pp. 14--35, 2001

[3] M. Stolze, R. Pawlitzek, and A. Wespi, "Visual Problem-Solving Support for New Event Triage in Centralized Network Security Monitoring:Challenges, Tools and Benefits," GI-SIDAR Conf. ITIncident Management & IT-Forensics (IMF), 2003.

[4] R. Ball, G. A. Fink, and C. North, "Home-Centric Visualization of Network Traffic for Security Administration," ACM Workshop Visualization and Data Mining for Computer Security (VizSEC/DMSEC), 2004, pp. 55-64.

[5] L. Girardin and D. Brodbeck, "A Visual Approach for Monitoring Logs," Proc. 12th Systems Admin. Conference (LISA), 1998, pp. 299-308.

[6] Anja Feldmann, "Continuous online extraction of HTTP traces from packet traces", Position paper for the W3C Web Characterization Group Workshop, November 1998

[7] John R. Goodall, "User Requirements and Design of a Visualization for Intrusion Detection Analysis", IEEE Workshop on Information Assurance and Security, 2005

[8] W. Yurcik, J. Barlow, K. Lakkaraju, and M.Haberman, "Two Visual Computer Network Security Monitoring Tools Incorporating Operator Interface Requirements," ACM CHI Workshop HCI and Security Systems HCISEC), 2003.

[9] The World Wide Web Consortium, http://www.w3.org/Protocols/rfc2616/rfc2616.html

[10] W3C Extended Log File Format, http://www.w3.org/TR/WD-logfile.html

[11] N. Friedman and Y. Singer. Efficient bayesian parameter estimation in large discrete domains, 1999

[12] B.H. Lee, S.H. Cho, S.D. Cha,"Real-Time Visualization of Web Usage Patterns and Anomalous Sessions", KIISC, Vol.14. no.4., p.97-110,2004