웹 어플리케이션 보안을 위한 프로파일 기반 탐지시스템 설계

박채금*, 노봉남
전남대학교 정보보호학과
e-mail: seagull*@lsrc.jnu.ac.kr

Design of Profile-based Intrusion Detection System For Web Application Security

Chae-Keum Park, Bong-Nam Noh
Dept of Linux Security, Chonnam University

요 약

최근 웹(World Wide WEb)은 전자상거래, e-commerce의 눈부신 성장과 더불어 그 이용률이 급격히 증가하였고, 이와 더불어 웹 취약점을 이용한 해킹사례도 증가하고 있다. 이 해킹 사례의 대부분은 웹 어플리케이션의 취약점을 이용한 것이다. 기존의 네트워크 침입탐지 시스템에서는 침입을 탐지하기 위해 시그너쳐 방법이 주로 사용되었다. 시그너쳐 방식은 시그너쳐를 기반으로 우수한 탐지율을 보인다. 그러나 웹 어플리케이션 공격은 다양한 원인과, 변형된 특성들을 가지고 있기 때문에 기존의 시그너쳐 기반의 방법으로는 특정한 패턴을 찾아내기가 곤란하다. 본 논문에서는 이를 보완할 수 있는 방법으로 프로파일 기반의 탐지방법을 제시한다.

1. 서론

최근 웹 서비스의 증가와 더불어 웹 취약성을 이용한 공격이 증가하고 있다. 2004년 말부터 급증하고 있는 웹 페이지 변조 공격과 같이 웹 어플리케이션의 취약점을 노린 해킹 사건들도 더불어 급증하고 있다. 웹 어플리케이션 및 웹 서비스의 보안을 이행하고 향상시키는 것을 돕기 위한 단체인 Open Web Application Security Project(OWASP)에서는 가장 심각한 10가지 웹 어플리케이션 보안 취약점을 발표하였다. 웹 해킹의 위험성과 피해정도가 심해지고 있는 만큼 그에 대한 대응이 절실해진 것이다.

침입탐지기술은 악의적인 사용자에 대한 방어 수단으로 보안에 있어 매우 중요한 요소이다. 네트웍 보안기술(NIDS)은 네트웍상의 트래픽을 모니터링하며, 시스템에 침입하려는 시도나, DDos공격과 같은 시도를 발견해낸다. 일반적으로 네트웍 보안기술은 네트웍 트래픽을 모니터링하고 분석하며, 침입패턴을 인식하여 보안경고를 발생한다.

이 과정에서 침입패턴을 인식하는 방법은 시그너쳐 방식과, 프로파일을 기반으로 하는 방식이 있다. 기존의 네트웍 보안기술에서는 시그너쳐 방식을 다수 채택하였다. 시그너쳐 방식에서는 정확한 탐지와 실시간 분석이 가능하기 때문이다. 그러나 네트웍 보안기술과는 다르게 웹 어플리케이션 보안에는 이 방식을 적용하기가 용이하지 않다. 왜냐하면, 웹 어플리케이션 공격은 다양한 원인과 변형된 특성들을 가지고 있기 때문이다. 그래서 기존의 시그너쳐 기반에서와 같이 특정한 패턴을 찾아내기가 곤란하다. 더군다나 웹 서비스와 관련된 취약점은 매일같이 발견되고 있어서, 시그너쳐 기반의 탐지방법에서는 관리자가 때마다 서버의 취약점을 패치해야하는 어려움이 있다. 그러므로 웹 보완을 위해서는 기존의 네트웍 침입탐지시스템의 시그너쳐기반을 보완할 수 있는 방식이 요구된다. 이 논문에서 제안하는 침입탐지 시스템은 웹 요청(Web request)의 파라미터들을 프로파일하여 웹 공격을 탐지한다.

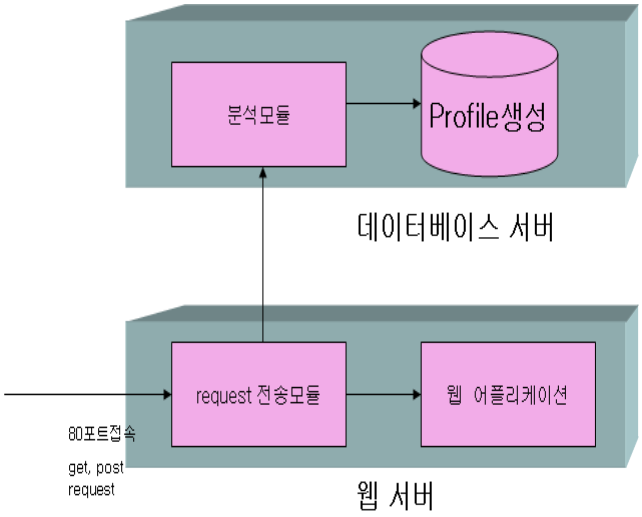
본 논문의 2장에서는 전체적인 시스템의 설계를 제시하고, 3장에서는 파라미터의 프로파일링 기준을 제시한다.

2. 프로파일기반 침입탐지 시스템 설계

2.1 프로파일기반 침입탐지 시스템

프로파일기반의 침입탐지시스템은 정상행위를 기준으로, 이와 상반되는 행위를 침입으로 간주하는 방법이다. 즉 프로파일 탐지기법은 공격행위가 정상행위와 다르다는 점을 가정한다. 이러한 가정 하에서 프로파일은 네트웍 트레픽에 대한 데이터마이닝, 감사데이터 분석을 위한 통계적 분석, 그리고 운영체제 시스템 콜을 위한 시퀀스분석으로 나누어진다.

이 논문에서는 통계적인 분석, 좀 더 자세히 설명하면 클라이언트의 GET 요청과 POST 요청의 파라미터 정보와, 그 파라미터 개개의 특징의 발생빈도를 측정함으로써 접근한다.



(그림 1) 정상행위프로파일 자동 생성 모듈

정상행위 프로파일링은 정상행위를 학습한다. 더 구체적으로 설명하자면 GET과 POST 요구패킷 안에 있는 특정 파라미터 정보들을 분석하여 그 빈도수를 누적한다.

먼저 request 전송모듈은 웹서버에서 80포트로 접속하는 데이터를 데이터베이스 서버로 전달한다. 데이터베이스 서버는 GET과 POST 정보의 파라미터들을 분석하여 정상행위 프로파일을 생성한다. 데이터베이스 서버를 따로 두는 이유는 1초에 수 십개에서, 많게는 수 천개가 발생하는 HTTP패킷을 하나하나 분석할 때에 많은 시스템 자원이 요구되기 때문이다. 그러므로 원활한 웹 서비스를 제공하기 위해서는 별개의 분석서버가 필요하다

분석모듈은 GET과 POST 요구패킷의 파라미터를 추출하고 파라미터의 길이, 문자의 발생빈도, 토큰정보 등의 특징을 기준으로 프로파일을 생성한다. GET과 POST의 차이는 클라이언트가 HTTP 요청을 할 때에 파라미터들이 포함되어 가느냐, 아니면 HTTP BODY 부분에 파라미터가 포함되어 요청이 되느냐에 따라 차이가 난다. 일반적으로 폼(form)을 이용하여 인증을 처리하는 경우 GET을 사용하게 되면 웹 서버 로그에 사용자의 계정과 비밀번호가 그대로 남게 되므로 대부분은 POST로 전송을 한다.

이 POST 요구정보는 URL 인코딩이 되므로 분석모듈에서는 URL디코딩을 통해 POST 요청를 분석하는 과정이 필요하다. (그림 2)는 디코딩전의 POST 정보이고, 그 아래는 URL 디코딩된 내용을 보여준다.

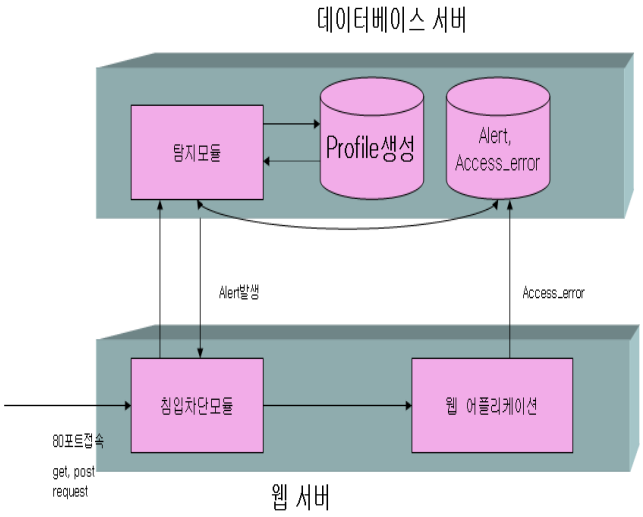
```
g%5Fgame%5Fid=jj0906&g%5Fgame%5Fscor
=24450&g%5Fgame%5Ftemp=0&use%5Fgam
e%5Fid=jj0906&use%5Fgame%5Fscor=24450
&use%5Fgame%5Ftemp=0

g_game_id=jj0906&g_game_scor=24450&g_ga
me_temp=0&use_game_id=jj0906&use_game_
scor=24450&use_game_temp=0
```

(그림 2) POST 요청의 디코딩 전과 디코딩후의 Body

프로파일은 각 파라미터들의 발생빈도를 누적한다. 즉 자동으로 학습하며, 프로파일을 생성한다. 결과적으로 보면 어떠한 파라미터 특징에 대한 발생빈도가 높으면, 정상행위로 간주하고, 발생빈도가 작으면 비정상행위로 간주하게 된다.

2.2 침입탐지모듈

일정한 학습기간이 지났을 때, 즉 충분한 프로파일이 생성되었을 때, 관리자는 탐지모듈을 작동시킬 수 있다. 탐지모듈은 (그림 1)의 분석모듈과 같이 GET과 POST 요청에 포함되어 있는 파라미터들을 분석하며, 그 특징들을 데이터베스에 저장된 프로파일과 비교한다. 만약 프로파일의 특정항목이 탐지모듈의 결과와 비교하여 상당한 차이가 발생할 경우 이를 침입으로 간주하게 된다. 이 때 침입으로 간주되는 기준은 보안 수준에 따른 관리자의 선택으로 이루어진다. 만약 기준 빈도수를 낮게 잡으면 발생 빈도가 적은 즉, 침입으로 더 확실시되는 웹 요청을 탐지하게 된다.



(그림 3) 침입탐지모듈

이와 더불어 데이터베이스에는 웹 어플리케이션에서 발생한 access_error로그가 저장된다. 이 로그는 웹 페이지가 잘못 오동작하여 발생하는 경우도 있지만, XSS(Cross Site Scripting)이나 Unbound File Call 혹은 Buffer overflow공격 등으로 발생할 확률도 높다. 왜냐하면 이러한 공격들은 한 번에 성공시키기가 매우 어렵기 때문에 반복되어 시도되고, 그 결과 access_error가 발생한다. 이 Access_error.log정보는 탐지모듈에서 발생한 Alert의 Source IP와 비교하여 탐지모듈에서 발생한 Alert에 대해 더욱더 확신을 가지게 된다. 또한 DB에 저장된 Access_error로그 자체만으로도 같은 Source IP에서 발생한 error의 시간 간격을 통해 공격 시도를 모니터링 할 수 있다.

결국 탐지모듈에서 발생한 경보로 인해 침입차단모듈에서는 침입으로 간주되는 연결을 끊거나, 그 IP에 대해 서비스를 거부하게 된다.

3. 프로파일을 위한 파라미터의 프로파일링 기준들

프로파일은 기본적으로 웹서버에 대한 GET과 POST 요청의 파라미터를 이용한다.

(그림 4)에서 먼저 GET 요청패킷의 구조를 보면, 경로정보와 파라미터는 "?"로 구분되어있고, 각 파라미터들은 "&"로 구분되어있어서 비교적 쉽게 파라미터정보들을 추출할 수 있다. GET 요청에서 해킹에 주로 이용되는 또 다른 정보는 Cookie값 변조이다. 쿠키값은 각 파라미터가 ";"로 구분되어 있음을 볼 수 있다.

```
218.151.111.32 [28/Aug/2005:12:59:42 +0900] "GET
/~hcchoi/board_zb/zboard.php?id=gallery_01_04&cred=
admin HTTP/1.1" 404 307
Cookie: CH_popup=done;
G%5FU%5FLTYPE=YES;USER%5FSEX=W;USER%
5FNAME=%C8%B2%C1%F8%C8%F1;USER%5FEM
AIL=jj0906%40orgio%2Enet;USER%5FOLD=20;USER
%5FID=jj0906;
```

(그림 4) 변조에 이용되는 Get 요구정보와 쿠키 값

POST방식에서는 주로 POST정보와 Body부분의 정보를 변조한다. Body 부분은 인코딩되어 있으므로, URL디코딩이 필요하다. (그림 5)는 POST정보와 디코딩된 BODY정보를 보여준다.

Body의 파라미터 역시 '&'로 구분이 된다.

```
POST /event/naruto/g_save_data.asp HTTP/1.1

g_game_id=jj0906&g_game_scor=24450&g_game_tem
p=0&use_game_id=jj0906&use_game_scor=24450&use
_game_temp=0
```

(그림 5)변조에 이용되는 POST 요구정보와 POST 요구정보의 Body부분

[3]프로파일링의 기준은 다음과 같다.

설계의 주요 아이디어는 [3]Krugel의 연구에서 얻었다. Krugel은 웹서버 log중 get정보만을 이용했으나, 이 논문에서는 get과 post요구정보 및 프로파일 결과를 이용한다.

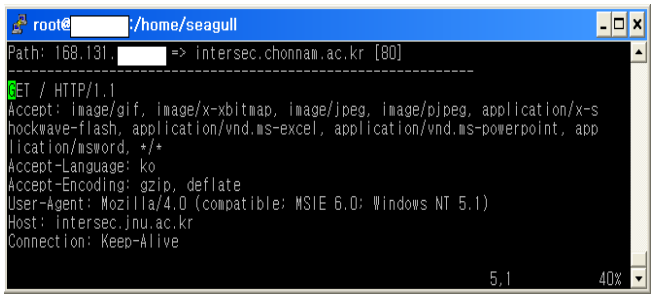
첫째, 길이정보를 통해 기준을 삼는다. 일반적으로 GET 요청의 파라미터는 고정된 길이가 대부분이다. 그러나 사용자 입력 부분에 긴 문자열을 삽입하는 Application 버퍼 오버플러우, SQL Injection, Directory/file/system Information Indexing등은 일정한 파라미터의 길이를 초과 할 수밖에 없다. 즉 정상적인 길이정보를 프로파일링하여 비정상 행위를 탐지한다.

둘째, 파라미터 문자의 특징을 기준으로 삼는다. 대부분의 파라미터는 사람이 읽을 수 있는 문자이며, 문자들의 분포역시 비교적 고르게 분포되어 있다. 그러나 악의적인 쉘 코드 등이 삽입 되면 그 문자열 특정문자가 반복되거나, 사람이 읽을 수 없는 문자가 발생할 확률이 높다.

셋째, 토큰 정보이다. 웹 어플리케이션은 플레그나

인덱스 같은 토큰 정보를 요구한다. 악의적인 사용자는 이 정보를 변조하는데, 정상적인 토큰정보를 프로파일링하여 비정상행위를 탐지한다.

넷째 특정항목의 부존재나 항목의 순서를 기준으로 한다. 일반적으로 스크립트나 HTML 폼같은 클라이언트쪽 프로그램은 request하기 전에 일정한 형식에 맞추어 전처리를 한다. 해커가 작성한 임의적인 웹 요청정보는 숫자나, 이름의 정보를 누락하거나 항목의 순서를 어기는 경우가 많다. 일정형식의 파라미터 내용들을 프로파일링 하여 여기에서 벗어난 요구정보를 비정상적인 행위로 간주한다. (그림 6)은 정상적인 GET 요구패킷을, (그림 7)은 웹 스케너가 임의적으로 보내는 GET요구정보를 모습이다. 웹 스케너가 보내는 요구정보는 일반적인 GET요구정보에서 보여지는 일반적인 형식에서 벗어나 있음을 볼 수 있다.



(그림 6) 정상적인 GET 요구패킷

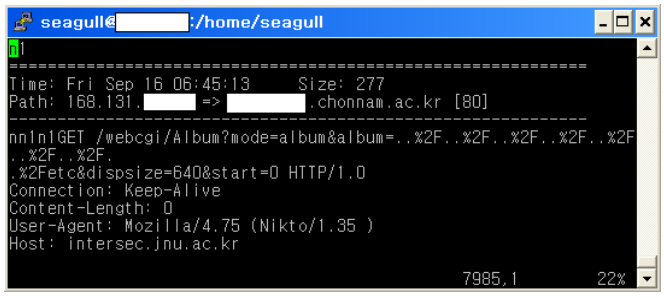


그림 7) 웹 스케너가 임의적으로 보내는 GET요구패킷

4. 결론

해커의 공격에 대한 시그너쳐 탐지방식은 시그너쳐를 기반으로 우수한 탐지율을 자랑한다. 그러나 웹 어플리케이션 공격은 다양한 원인과, 변형된 특성들을 가지고 있기 때문에 기존의 시그너쳐 기반의 방법에서와 같이 특정한 패턴을 찾아내기가 곤란하다. 본 논문에서는 이에 착안하여 웹 어플리케이션 공격에 대한 프로파일 방식의 침입탐지 시스템의 설계를 제시하였다. 물론 이 논문에서 제시한 프로파일의 몇 가지 특징들은, 정상행위와 너무도 흡사하게 변조되어 거의 정상행위나 다름없는 공격에 대해서는 탐지하지 못하는 한계점을 지니고 있다. 하지만, 이러한 한계점은 웹 어플리케이션에서 가지는 access_error.log와 같은 로그정보는 침입탐지를 위한 중요한 단서들과 결합하여, 프로파일방식의 한계점을 보완 할 수 있을 것이다. 앞으로는 탐지율을 높이기 위해 추가적인 프로파일링 기준을 연구해 나갈 것이다.

참고문헌

[1] Anitha Nalluri and Dulal C.Kar. "A Web-Based System For Intrusion Detection". CCSC. pp274-280. 2005

[2] The Open Web Application Security Project(OWASP) Top Ten Most Critical Web Application Security Vulnerabilities. http://www.owasp.org.2004

[3] Christopher Krugel. "Anomaly Detection of Web-based Attacks". In Proceeding of CCS'.PP251-261, 2003

[4] 이영석. " 어플리케이션 보안기술동향". 정보보호학회지. 제15권 제1호, pp83-89, 2005.2

[5] 황순일 저, "웹 해킹 패턴과 대응" 2005.4