스크립트 파일 기반의 효율적인 웹 공격 탐지 프로파일링

임종혁° 박재철 김동국 노봉남
전남대학교 정보보호협동과정
{azazel02°, chori}@lsrc.jnu.ac.kr {dkkim, bongnam}@chonnam.ac.kr

Efficient Script-File based Profiling for Web Attack Detection

JongHyuk Im°, JaeChul Park, DongKook. Kim, BongNam Noh
Interdisciplinary Program of Information Security, Chonnam National University
Dept. of Electronics Computer & Information Engineering, Chonnam National University

요 약

비정상행위 탐지를 위한 프로파일 기술은 침입탐지시스템의 성능 향상을 위한 핵심기술로서, 높은 공격 탐지율과 침입탐지시스템의 수행 시간 단축을 위해 반드시 요구되는 기술이다. 최근 인터넷의 보급과 활성화로 웹 어플리케이션 보안을 위한 연구가 활발히 진행되고 있으나, 웹 애플리케이션의 개발 언어와 공격 특성을 반영하지 못해 그 효율성이 저하되고 있다. 본 논문에서는 웹 공격 탐지를 위해 연구 되었던 서열정렬 알고리즘을 이용한 웹 공격 탐지의 성능 개선을 위하여 웹 애플리케이션 개발에 주로 사용되는 스크립트파일을 기반으로 한 프로파일 방법을 제안하고 실험 결과를 기술하였다.

1. 서 론

인터넷의 발전으로 많은 컴퓨터 사용자들이 전자메일, 온라인 쇼핑, 멀티미디어 정보 등과 같은 서비스를 제공하는 인터넷 사용을 생활화하고 있다. HTML 기술을 이용한 웹 기술의 발전은 현대인들의 정보를 빠르고 쉽게 공유할 수 있도록 해주고 있어 그 이용이 기하급수적으로 증가하고 있는 추세이다. 그러나 그에 따른 부작용으로 중요 시스템에 대한 정보 유출, 전산망 침해 등과 같은 침입 행위 또한 빠른 속도로 증가하고 있다. 최근, 웹 애플리케이션의 취약점을 이용한 해킹 사고가 급증하고 있으며 이에 대한 대처 방안으로 웹 애플리케이션에 특화된 침입탐지시스템에 대한 연구가 부각되고 있다.

웹 어플리케이션 취약점은 대부분 입력 값을 검증하지 않기 때문에 발생하는 문제로, 악의적인 사용자가 웹 애플리케이션 매개변수에 비정상적인 값을 입력함으로써 공격이 이루어진다. 매개변수 변조를 이용한 웹 공격들에는, 대표적으로 SQL Injection, Command Injection, Include Vulnerability, Directory Traversal, XSS(Cross Site Scripting)과 같은 공격기법들이 있다[5].

최근 연구에서 웹 애플리케이션의 매개변수 데이터에 있는 키워드를 아미노산 코드로 치환한 후 서열 정렬 기법을 적용하여 공격을 탐지하는 방법[1]이 제안되었다. 매개변수 데이터를 프로파일하고 유사도를 측정하기 위하여 서열의 전체길이를 이용하고 있지만 웹 애플리케이션의 주요 공격 대상이 되는 스크립트 파일과 이에 수반되는 매개변수를 고려하지 않았기 때문에 비교할 서열의 선택이 모호하고 정확한 유사도 측정이 어렵다는 단점을 가지고 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 기존 방법보다 정확한 탐지를 위해 스크립트 파일 기반의 프로파일 기법을 적용하여 기존의 실험 결과와 비교분석하고, 그 효율성을 측정하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 애플리케이션의 입력 값 전달 방식에 대하여 분석하고 3장에서는 웹 공격 탐지를 위한 효율적인 프로파일 방법에 대해 소개하고 4장에서 성능 분석을 위한 프로파일 방법과 실험 결과를 고찰한다. 5장에서는 결론과 함께 향후 연구 과제를 제시한다.

2. 동적 웹 페이지와 입력 값 전달 방식

웹 애플리케이션의 구동 방식은 사용자의 요청에 따른 입력 값을 쿼리스트링을 통하여 서버에 전송하게 되고, 결과를 다시 클라이언트에게 보내주는 방식이다. 매개변수 값을 서버에 전송할 때 URL을 통해 값을 직접 전송하는 GET 방식과 입력된 데이터를 패킷 헤더 안에 포함시켜 전송하는 POST 방식이 있다. 두 방법 모두 정도의 차이는 있지만 서버에 보내져 처리 된다는 점에서는 모두 같다. 즉, 웹에서 일어나는 모든 일은 GET과 POST

* 본 연구는 정보통신부 대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었습니다.

매개변수를 처리하는 것으로부터 시작되고 웹 애플리케이션을 이용한 대부분의 공격 또한 매개변수 값을 변경하는 것으로부터 시작한다는 것을 뜻한다.

2.1 GET 메서드

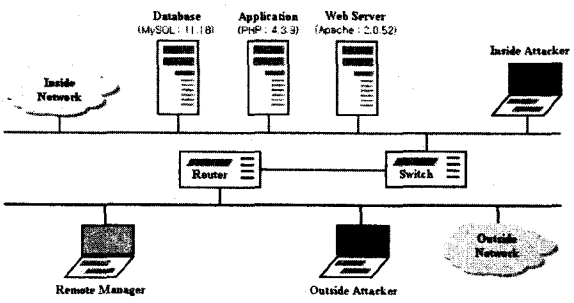
폼(Form) 태그에서 METHOD의 값을 GET으로 하거나 생략하면 사용자의 입력 값들이 환경변수(Environment Variable)에 저장되어 넘겨진다. GET 메서드는 폼 태그를 사용하지 않고 바로 URL에 인수를 첨가하여 사용할 수도 있는데 예를 들어,
http://www.abc.com/cgi-bin/abc.
cgi?First+Name=foo&Last+Name=bar와 같은 형식으로 사용될 수 있다. GET 메서드를 이용하면 입력 값들은 환경변수의 하나인 쿼리스트링 형식으로 전달되고 여러 가지 방법으로 인코딩되어 서버에 전달된다.

2.2 POST 메서드

폼 태그에서 METHOD의 값을 POST로 하면, 표준입력을 통해서 전달된다. GET 메서드가 인수를 통해서 전달되므로 명령행의 길이에 제한을 받는 반면, POST 메서드는 표준입력을 이용하므로 데이터양의 제한이 없고 또한, POST 메서드에서도 환경변수들은 표준입력과 함께 전달되어 GET 메서드와 같은 방법으로 인코딩된다. 본 논문에서는 POST 메서드의 매개변수 데이터를 GET 메서드와 동일한 형식으로 저장하여 데이터를 수집하였다.

3. 스크립트 파일명을 이용한 프로파일링

클라이언트의 요청에 의해 전송되는 입력 값은 앞서 언급했던 GET과 POST 두 가지 방법으로 웹 서버에 전송되는데 이들을 쿼리스트링 형식으로 동일하게 변환하여 로그 데이터를 수집하였다. 수집된 데이터는 추출할 키워드 설정에 의해 생성된 치환 매트릭스를 참조하여 데이터 필터링 및 아미노산 코드로 변환하고 전역정렬(Global sequence alignment) 하여 그 유사도를 측정하였다[4].



(그림 1) 데이터 수집을 위한 네트워크 환경

그림 1은 웹 애플리케이션의 매개변수 데이터 수집을 위한 네트워크 환경을 보여주고 있다. 웹 애플리케이션 서버의 OS는 커널버전 2.6.9-1인 Fedora Core 3이고 PHP로 개발된 게시판을 이용하여 매개변수를 추출하였다.

3.1 데이터 축약 및 분류

수집된 데이터는 웹 공격에 주로 사용되는 데이터베이스 질의어와 덧셈(+), 뺄셈(-)과 같은 연산자, 그리고 샵(#)이나 하이픈(-)과 같은 특수문자 등으로 구성되어있는 치환 매트릭스를 참조하여 데이터 필터링 및 아미노산 코드로 변환하였다.

기존 연구에서는 가공된 데이터를 프로파일 하기 위해 모든 정상 서열들의 길이를 측정하고 그 범위값을 구하여 프로파일 하는 방법을 사용하였다. 본 논문에서는 웹 공격 탐지율을 높이고 대응되는 프로파일 서열의 빠른 검색을 위하여 가공된 매개변수 데이터를 스크립트 파일명에 따라 나누어 저장하는 방법을 사용하였다.

3.2 프로파일 및 유사도 측정

프로파일 생성과 유사도 측정을 위하여 생물정보학에서 연구되는 서열 정렬 방법이 사용된다. 서열 정렬 방법은 대표적으로 전역정렬[2]과 지역정렬[3] 두 방법이 있는데 본 논문에서는 서열의 전체적인 일치에 초점을 두는 전역정렬 방법을 사용하였다. 전역정렬 방법은 서열 원소가 일치하는 경우의 점수와 불일치하는 경우, 공백을 넣어주는 경우 각각에 해당하는 점수를 주고, 가능한 여러 가지 정렬 중 점수 합계가 높은 서열정렬 결과를 찾는다. 정상 매개변수 데이터와 공격을 시도한 매개변수 데이터를 비교해보면, 전체적인 서열의 구조는 비슷하지만 입력 값이 부분적으로 다르거나 새로운 서열이 삽입되어 길이가 늘어난 것을 볼 수 있다. 따라서 두 가지 정렬 방법 중에서 서열의 전체적인 일치에 초점을 두는 전역정렬 방법이 효과적이다[1].

$$a[i,j] = \max \begin{cases} a[i,j-1] + gap \\ a[i-1,j-1] + p(i,j) \\ a[i-1,j] + gap \end{cases} \quad \text{(식 1)}$$

식 (1)의 $p(i\ j)$는 서열1의 i번째 원소와 j번째 원소의 일치 여부에 따른 점수이고, gap은 공백 삽입에 대한 감점 점수로서 이를 이용하여 행렬 a를 채운다. 이 행렬 a의 오른쪽 하단의 마지막 값이 전체 정렬의 최댓값이 된다.

프로파일 생성과 유사도 평가를 위한 과정은 동일한 식을 이용하여 구하여지며 그 과정은 다음과 같다. 정상데이터 N, 비정상데이터 A는 키워드 치환 매트릭스를 이용하여 N *CodeAmino*와 A *CodeAmino*로 변환된다.

프로파일 생성을 할 경우, 같은 스크립트 파일명을 가진 정상 N_a *CodeAmino*와 정상 N_b *CodeAmino*를 서열 정렬하여 일치도가 100%가 아닐 경우 프로파일에 P_a *CodeAmino*, P_b *CodeAmino*로 프로파일링 한다.

공격 여부를 판단하기 위한 유사도 측정의 경우, 프로파일 서열 P_a*CodeAmino*, 공격 서열 A_a*CodeAmino* 의 전체 정렬 일치도 I 는 아래의 식 (2)과 같이 백분율로 계산된다.

$$I = (\ alignment\ match\ code\ (\ P_a CodeAmino, A_a CodeAmino\ )\ /\ A_a CodeAmino\ length + gap\ ))\ \times\ 100 \quad \text{(식 2)}$$

[표 1] 프로파일에 이용되는 정보

스크립트 파일명	매개변수 데이터	아미노산코드로 변환된 서열	길이
bbs	bbs.php&id=websec&page=1&category=&sn=off&ss=on&sc=on...	CCIDPEPPAMEQAMEQAMEQAMEQAMEQAMEQ...	50
	bbs.php&id=websec&page=1&select_arrange=headnum&desc=asc...	CCIDPEPPAMEQAMEQAMEQAMEQAMEQAMEQ...	54
....	....	....	....
admin	admin.php&exec=view_group&group_no=1	CCADPEPPAMEQVWVSAMEQYY	20
	admin.php&group_no=1&exec=modify_group	CCADPEPPAMEQAMEQVSYY	22
....	....	....	....

표 2는 정상 매개변수 데이터를 식 1로 정렬하고 식 2의 방법으로 유사도 측정하여 프로파일 하였을 때 저장된 정보들의 일부이다.

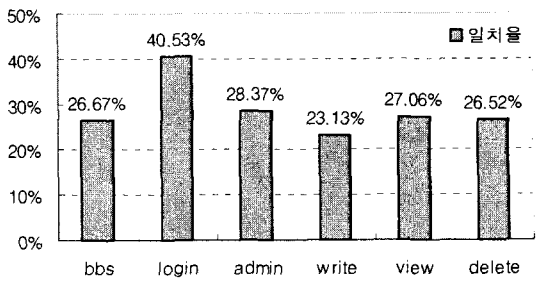
4. 실험결과

본 논문에서는 수집된 매개변수 데이터는 스트링 매치 방법을 이용하여 스크립트 파일에 따라 분류되어진다. 종전의 연구에서는 서열의 전체 길이에 따른 범위 값을 산출하여 프로파일 했기 때문에 각 서열들의 전체 길이 값의 차이가 클 경우 프로파일에 대응되는 범위 값 또한 커지게 되므로 많은 프로파일 서열과 정렬을 수행해야하고, 공격에 따른 매개변수 변화를 정확히 측정할 수 없다는 단점이 있었다. 이러한 단점을 극복하기 위해 본 논문에서 제안한 스크립트 파일에 기반의 프로파일은 대응되는 정상 서열의 개수를 줄여 빠른 연산을 가능하게 하고 웹 애플리케이션의 서비스에 따른 공격을 파악할 수 있어 보다 정확한 공격 탐지를 할 수 있었다.

[표 2] 스크립트에 따른 공격과 정상 데이터의 수집

No.	Group	File Name	Attack Name	Count
1	비정상	bbs login admin write view delete	SQL Injection	20
			Command Injection	20
			Directory Traversal	20
			Include Vulnerability	20
			XSS(Cross-Site Scripting	20
2	정상	bbs login admin write view delete	Normal Use	100

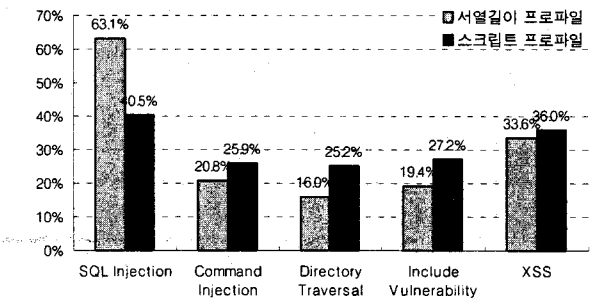
표 2의 정상 데이터는 웹 어플리케이션의 다양한 서비스를 정상적으로 이용한 100개의 매개변수 데이터이고 비정상데이터는 각 스크립트 파일별로 SQL Injection[6]과, Command Execution, Directory Traversal[8], Include Vulnerability[9], 그리고 XSS[7] 공격을 수행하여 수집하였다. 제안한 프로파일 방법의 성능 측정을 위하여 스크립트 파일에 따른 공격 탐지 결과와 공격의 종류에 따른 일치율을 비교하였다.



(그림 2) SQL Injection에 따른 스크립트 정렬

그림 2는 각 스크립트 파일에 SQL Injection 공격을 수행한 후 정상 서열과 비유사도를 측정한 평균값이다.

그래프에서 볼 수 있듯이 Login 스크립트 파일을 이용한 공격의 일치도가 40.53%로 가장 높게 나타나는데 그 이유는 사용자 인증을 우회하기 위해 SQL Injection 공격을 할 때 변조되지 않은 많은 매개변수가 함께 정렬되기 때문이다.



(그림 3) login 파일에 대한 공격 데이터 정렬 결과

그림 3은 서열 길이를 이용한 프로파일 방법과 스크립트 파일을 이용한 프로파일 방법의 공격 탐지율을 비교한 그래프이다. 사용자 인증을 할 때 사용되는 login 스크립트 파일의 경우 스크립트 파일에 따라 분류하여 프로파일 하고 비유사도 측정하였을 때 탐지율이 전체적으로 낮아진 것을 볼 수 있다. 하지만 서열의 길이에 따른 프로파일의 경우 SQL Injection과 Directory Traversal 공격 탐지율이 약 45% 이상 차이가 나기 때문에 임계값 설정에 따른 오탐 발생 가능성이 높아진다. 반면 스크립트 파일에 따른 프로파일의 경우 탐지율이 낮은 공격과 높은 공격의 평균값 차이가 약 10% 정도로 고르게 나타나 스크립트 파일을 이용한 프로파일 방법이 오탐율을 줄일 수 있다는 것을 알 수 있다.

실험 결과 서열의 길이를 이용한 프로파일의 경우 모든 스크립트 파일에서 SQL Injection 공격이 높은 일치도를 나타냈는데 그 이유는 사용자 입력 폼을 통한 로그인 인증우회 공격을 시도할 때 변조되지 않는 많은 정상 매개변수들이 함께 전달되어 정렬되기 때문이며 스크립트 파일을 이용한 프로파일과 정렬하였을 때 이러한 문제를 해결할 수 있었다.

5. 결론

본 논문에서는 서버로 전송되는 매개변수를 스크립트 파일에 따라 분류하고 프로파일 하는 방법을 제안하였다. 실험 결과 기존 프로파일 방법보다 전체적으로 웹 공격의 탐지의 정확성을 높였고 각 공격에 따른 탐지율의 차이가 크지 않아 임계값 설정에 의한 오탐율을 줄일 수 있음을 보였다. 이는 웹 애플리케이션의 스크립트 파일과 그에 따른 매개변수들이 웹 공격 탐지의 중요한 척도가 될 수 있음을 알 수 있다. 향후에는 실제 침입탐지 시스템 구축 및 성능 평가가 필요하다.

참고문헌

[1] Jae-Chul Park, and Bong-Nam Noh. SQL Injection Attack Detection: Profiling of Web Application Parameter using the Sequence Pairwise Alignment. In Proceedings of The 7th International Workshop on Information Security Applications, LNCS, 2006.

[2] Needleman, S. B., Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48:443-453, 1970

[3] Waterman, M. S., Smith, T. F., Beyer, W. A. Some biological sequence metrics. Adv, Math. 20: 367-387, 1976

[4] Jae-Chul Park, and Bong-Nam Noh. Detection of Parameter Manipulation using Global Sequence Alignment. In Proceedings of The International Conference on Next Generation Web Seervices Practices, IEEE Press, 2006.

[5] OWASP. vulnerability. http://www.owasp.org/index.php/Category:Vulnerability, 2006.

[6] William G. J. Halfond, Alessandro Orso. AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks. Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering ASE '05, November 2005.

[7] G. A. Di Lucca, A. R. Fasolino, M. Mastoianni, P. Tramontana. Identifying Cross Site Scripting Vulnerabilities in Web Applications. Proceedings of the Web Site Evolution, Sixth IEEE International Workshop on (WSE'04), September 2004

[8] Giovanni Vigna, William Robertson, Davide Balzarotti. Network intrusions: Testing network-based intrusion detection signatures using mutant exploits. Proceedings of the 11th ACM conference on Computer and communications security, October 2004

[9] Min Wu, Robert C. Miller, Greg Little. Catching phish: Web wallet: preventing phishing attacks by revealing user intentions. Proceedings of the second symposium on Usable privacy and security SOUPS '06, July 2006