한국항행학회 논문지 제14권 제5호 2010년 10월

웹 공격 분석 및 공격 데이터베이스 생성을 위한 효과적인 표현 방법에 관한 연구

An Attack Behavior Expressions for Web Attack Analysis and Composing Attack Database

이창훈*

Chang-hoon Lee*

요 약

최근에는 웹을 통한 서비스 증가와 더불어 이와 관련된 공격이 증가하고 있다. 또한, 웹 공격 형태는 공격을 성공시키기 위하여 여러 가지 공격을 사용하는 방법을 시도하고 있다. 이와 같이 웹 공격 방법이 다양화 되고 있는 추세이지만, 웹 공격을 방어하기 위한 방법에 관한 연구는 미비하다. 따라서 웹 어플리케이션을 보호하기 위해 웹 공격을 분류하고 이를 통하여 웹 공격의 특성을 파악할 필요가 있다.

본 논문에서는, 현재 웹 어플리케이션에서 수행되는 웹 공격의 특성을 파악하고, 이를 효과적으로 표현하는 방법을 제안한다. 공격이 가능한 웹 공격 시나리오를 다양하게 생성하여, 제안하는 표현 방법을 검증한다.

Abstract

Nowadays, followed the internet service contents increasing makes also increase attack case on the web system. Usually web attack use mixed many kinds of attack mechanism for successfully attack to the server system. These increasing of the kinds attack mechanism, however web attack defence mechanism is not follow the spread of the attack. Therefore, for the defends web application, web attack should be categorizing and analyzing for the effective defense.

In this paper, we analyze web attack specification evidence and behavior style that use for effective expressions what we proposed. Also, we generate web attack scenario, it is for using verification of our proposed expressions.

Key words : Web attack expression, Attack behavior analysis, Abnormal detection, Attack database

I. 서 론

웹을 통한 서비스 제공은 기업의 정보전달이나 홍보의 목적 외에도, 전자상거래나 기업의 마케팅, 개인을 위한 정보 교류를 위해 이용된다. 최근에는 인터넷 쇼핑몰과 인터넷 뱅킹 같은 비즈니스 용도로 많이 이용되고 있다. 이처럼 웹은 기존의 간단한 정보 제공의 역할에서 인트라넷이 포함된 주요 정보 매체로 발달하고 있다. 그러나 웹을 통한 서비스 증가와 더불어 이와 관련된 보안사고 및 웹 취약성을 이용한

* 한신대학교 컴퓨터공학부 (School of Computer Engineering, Hanshin University)
· 제1저자 (First Author) : 이창훈
· 투고일자 : 2010년 8월 26일
· 심사(수정)일자 : 2010년 8월 27일 (수정일자 : 2010년 10월 20일)
· 게재일자 : 2010년 10월 30일

공격 또한 증가하고 있다. CERT의 보고서에 따르면, 전체 해킹의 약 70%가 웹 해킹으로 분석되고 있다고 발표해 웹을 통한 해킹이 심각한 수준임을 알 수 있다[1].

일반적인 공격을 방어하기 위한 침임 탐지 시스템과 같은 보안 시스템은 일반적으로 시스템 레벨에서 탐지 및 방어를 수행한다. 하지만, 웹 공격은 어플리케이션 레벨에서 이루어지는 공격이므로, 일반적인 보안시스템을 통한 웹 어플리케이션 공격에 대한 대응은 기대하기 어렵다[2]. 즉, 기업에서 주로 이용하는 보안 시스템이 작동 되더라도 웹 어플리케이션 취약점을 이용한 공격을 사전에 예방할 수 없다. 또한, 기존에는 서버를 대상으로 한 공격이 대부분이었지만, 최근에 많은 웹 사이트가 사용자에게 유연한 서비스를 제공하기 위해 클라이언트 중심의 스크립트를 주로 이용함에 따라 이를 사용하는 크로스 사이트 스크립팅과 같은 클라이언트 대상 공격이 증가하고 있다[3].

최근 증가하고 있는 웹 어플리케이션 공격에 대응하기 위해, 웹 공격에 관한 분석, 다양한 웹 공격 표현 방법, 새로운 공격 시나리오 생성 방법 등에 관한 연구가 필요하다. 그러나 이와 관련된 연구는 미비하다. 따라서 본 논문에서는 웹 어플리케이션 공격에 관한 시나리오를 분석하고 다양한 공격 시나리오를 효과적으로 표현 할 수 있는 방법을 제안한다. 또한 본 논문에서 제안하는 방법을 검증하기 위해, 웹 어플리케이션 공격을 일정한 기준에 의해 분류하고, 이를 이용하여 실제 공격자들이 행할 수 있는 공격 시나리오를 작성하여 발생 가능한 시나리오를 생성하여 본다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 통하여 웹 공격들이 어떻게 분류되었고, 어떻게 표현되었는지에 대해 설명한다. 3장에서는 실제 웹 공격에 대해 분류하고, 이를 바탕으로 공격 시나리오를 표현한다. 4장에서는 제안하는 방법을 이용하여 여러 가지 공격 사나리오를 생성해 본다. 5장에서는 결론과 향후 연구에 대해서 토의한다.

Ⅱ. 웹 공격의 분류 및 분석

2-1 웹 공격 분류

웹 공격 기법들이 다양해지고 복잡해짐에 따라 웹 공격에 관한 용어와 웹 공격 기법에 대한 이해가 일관적으로 사용되지 못하여 혼란을 가중 시키게 되었다. 따라서 이에 대한 연구가 웹 보안 기관들을 중심으로 진행되게 되었다. 그중 하나가 Open Web Application Security Project(OWASP)이며, OWASP는 10개의 가장 심각한 웹 어플리케이션 보안 취약점을 보고서로 발표하였다[4].

OWASP TOP 10은 MITRE에서 제공한 취약점 유형 데이터[5]를 기반으로 웹 어플리케이션 보안 취약점에 대해 발생빈도와 중요성을 분석하여 표 1과 같이 10개의 공격기법을 선정하였으며, 이에 관한 취약점 및 대응 방법에 대해 설명하였다. 실제 발생한 사건을 기반으로 작성된 이 보고서는 최근에 가장 문제가 되고 있는 웹 보안 위협이 무엇인지 보여준다. 또한, 이 보고서를 통해 악의를 가진 공격자가 사용하려는 취약점이 무엇인지 파악하여 관리자들이 대응할 수 있도록 한다. 그러나 이 보고서는 이미 알려진 웹 공격 분류만을 포함하고 있어, 급격히 변화하는 웹 공격에 대한 신속한 대응이 어렵다. 또한 본 논문에서 제안하는 방법과 같이 여러가지 공격 방법을 사용한 공격 시나리오를 예측하기 어렵다.

J.S Seo[6]는 웹 서비스의 특성을 반영한 침입탐지시스템 개발에 유용하게 이용되기 위해 공격 발생 원인과 공격의 원인이 존재하는 위치에 따라 웹 공격 분류를 제안하였다. 공격의 발생 원인에 따라 구현 오류, HTTP Specification 오류, 잘못된 설정, 클라이언트 코드 변조, 비정상적인 입력, 과도한 처리량으로 공격을 분류하였다. 각 공격들의 발생 원인에 따라 공격을 탐지할 수 있는 위치가 다를 수 있기 때문에 웹 클라이언트, 웹 인터페이스, 웹 서버로 탐지 위치에 따라 공격을 분류하였다. 이 논문은 공격을 탐지하기 위한 특성이 공격의 원인과 밀접한 관계가 있고 웹 서비스의 공격들은 공격 발생 원인과 공격 발생 원인의 위치에 따라 그 특성이 잘 나타나기 때문에 웹 공격들을 이해하는데 매우 용이하게 사용될 수 있다. 그러나 이 연구는 인터넷 서비스에 특화된 침입 탐지 시스템을 설계하기 위한 목적으로 웹 공격들을 분류하였으므로 다른 웹 공격에 대해서도 분석할

필요가 있다.

표 1. OWASP Top 10 웹 어플리케이션 취약점
Table 1. Web Application Vulnerability List of the OWASP Top 10.

A1-크로스 사이트 스크립팅(XSS)
A2-인젝션 취약점
A3-악성 파일 실행
A4-불안전한 직접 객체
A5-크로스 사이트 요청 변조(CSRF)
A6-정보 유출 및 부적절한 오류 처리
A7-취약한 인증 및 세션 관리
A8-불안전한 암호화 저장
A9-불안전한 통신
A10-URL 접속 제한 실패

2-2 공격 표현 방법

보안 사고에 관한 취약성을 분석하기 위한 많은 연구들이 이루어지고 있지만, 이를 정형화하여 표현하는 방법에 관한 연구는 미비하다. 따라서 취약성에 대해 다양한 각도로 분석하여 표현하고자 하는 연구가 필요하다. Xinming Ou[7]는 논리적으로 접근하는 방법을 이용하여 공격 그래프를 표현하고 생성하는 방법에 대해 제안하였다. 공격 그래프에서 노드는 논리적인 상태를 표현하지만, 네트워크의 전체적인 상태를 나타내지 않는다. 공격 그래프의 끝은 네트워크 구성과 공격자의 잠재 권한 사이의 인과관계를 열거한다. 이러한 공격 그래프는 공격의 원인 또는 공격이 발생하는 이유에 관해 표현한다. 이 그래프는 시스템 구성 정보와 공격자의 잠재 권한 사이에서 분명한 인과 관계를 나타내며, 모든 가능한 공격 시나리오에 대해 표현 가능하다. 이 연구에서 제안한 알고리즘을 통해 공격 그래프를 자동적으로 생성하고, 네트워크 공격을 효과적으로 표현 할 수 있다.

STATL[8]는 침입 탐지 시스템을 위한 공격 기법 표현 언어로서 공격 시나리오를 효과적으로 표현하기 위한 방법 또한 제안하였다. 이 연구는 공격을 상태와 전송 방법에 따라 분류하여 단계적으로 모델링하여 표현하였다. STATL의 제안한 방법 중 시나리오 표현 방법은, 상태는 공격 시그니쳐를 정의하여 표현하고 전송 방법은 새로운 상태로 이동하기 위한 이벤트들을 정의하였다. 이 연구는 상태와 전송 방법을 이용하여 상태 다이어그램으로 표현 가능하다. 본 논문에서는 이러한 시나리오 공격 표현 방법을 참고하여 새로운 공격 표현 및 공격 생성 방법을 제안한다.

2-3 웹 공격

웹 공격은 웹 서비스와 관련된 웹 서버, 데이터베이스 시스템 등을 공격하여 정상적인 웹 서비스를 방해하거나 권한 없는 정보를 취득하는 것을 말한다. 기존에는 웹 서버를 대상으로 한 공격들이 대부분이었지만, 최근에는 클라이언트에서 이루어지는 공격이 급격히 증가하고 있다.

클라이언트를 대상으로 한 공격 방법 중 사용자 입력 데이터 공격은 사용자가 입력한 데이터가 웹 서버의 웹 애플리케이션으로 전송될 때 데이터를 조작하여 공격하는 방법이다. 웹 어플리케이션 프로그램은 흔히 사용자 입력 과정에서 많은 오류가 발생할 수 있으며, 입력된 데이터에 의도적으로 악의적인 코드나 스크립트, 운영체제 명령어 등을 포함시키는 경우에는 컴퓨터 시스템에 심각한 피해를 입힐 수도 있다. 경우에 따라서, 입력 데이터에 포함된 코드나 명령이 웹 서버에서 실행되어 해당 웹 사이트나 서버 자체가 심각한 보안 위험에 노출되기도 한다[9]. 다음에 설명된 공격들은, 대표적인 공격들이다.

•크로스 사이트 스크립팅

크로스 사이트 스크립팅은 사용자의 입력을 별도의 검증하지 않는 경우, 공격자가 JavaScript, Active X 등으로 작성한 악의적인 스크립트를 다른 사용자에게 전송하여 실행하도록 하는 것이다[10][11]. 크로스 사이트 스크립팅의 대표적인 두 가지 공격 방법은 Stored Attack과 Reflected Attack이다. Stored Attack은 악의적인 스크립트를 게시판 등을 통하여 웹 서버에 저장하는 공격을 의미하며, Reflected Attack은 사용자가 웹 사이트 방문에 대한 응답으로 스크립트가 실행되도록 하는 공격을 의미한다[12][13].

•SQL Injection

SQL Injection은 웹 어플리케이션 환경이 SQL 요

청과 같은 외부 명령어 실행을 허용한다는 취약점을 이용하여, 동적으로 생성되는 일부 SQL 질의 등에 SQL 명령을 삽입하는 공격이다[14][15]. 공격자는 SQL Injection을 이용하여 데이터베이스 정보에 대한 권한을 획득하여 데이터베이스에 접근 할 수 있다[16][17].

•디렉토리 노출 (Directory traversal)

디렉토리 노출은 악의적인 사용자가 접근이 웹 서버 내의 제한된 파일들의 위치를 파악하여 그 파일의 내용을 보거나 파일을 실행 시키는 것이다[9][18]. 제한된 파일의 내용을 보는 것은 비밀번호가 저장된 파일 내용을 보는 것과 같은 프라이버시 문제를 일으킬 수 있으며, 제한된 파일을 공격자의 의도대로 수정되고 통제 되어 실행시킬 수 있다[19][20].

Ⅲ. 웹 공격의 효과적인 표현 방법

3-1 웹 공격 분류

알려진 웹 공격에 대한 방어가 이루어진 웹 어플리케이션에 대해 공격자는 공격을 쉽게 성공할 수 없다. 따라서 공격자들은 이를 우회하여, 공격을 성공시키기 위해 여러 가지 웹 공격 기법들을 연속적으로 사용하여 새롭게 공격방법을 구성하여 공격을 시도한다. 이러한 복합적인 공격 기술은 알려지지 않은 공격이기 때문에, 이를 분석하여 공격자들의 어떠한 다양한 방법과 단계로 공격을 시도하는지 파악해 볼 필요가 있다. 따라서 본 논문에서는 웹 공격을 공격 수행 단계에 따라 구분하여, 관계성을 파악하고 공격자가 새로운 공격을 구성하는 방법을 유추하여 공격 생성 방법을 표현하였다.

예를 들어, 크로스 사이트 스크립팅의 대표적인 공격 방법으로 웹 사이트 게시판에 악의적인 스크립트를 삽입하여 공격하는 시나리오를 공격 수행 단계에 따라 분류하여 표 2와 같이 표현한다.

표 2. 크로스사이트 스크립팅의 공격 수행 분류
Table 2. Attack Behavior Analysis of a Cross Site Scripting

수행단계	공격 수행 단계 시나리오
공격 목표 설정	취약점 스캔을 통한 웹 어플리케이션 파악
로그인	정상적인 로그인
입력 대상 선정	게시판 사용
입력 방법	Java Script를 이용한 입력
실행	페이지 오픈으로 자동 실행
공격 성공	피해자의 위조된 사이트 방문

같은 공격 수행 단계내에서도 여러 가지 공격 방법이 존재한다. 표 3은 공격 수행 단계에서의 여러 가지 공격 시나리오를 보여준다. 웹 어플리케이션에 로그인을 하기 위한 방법으로는 아이디와 비밀번호를 모두 알고 있어 정상적으로 접속하는 방법, 무차별 공격을 통해 로그인을 하는 방법 등 여러 가지가 있다. 이처럼 같은 공격 수행 단계라 하더라도 공격자는 다양한 방법을 통해 공격을 시도 할 수 있다. 따라서 웹 공격에 대해 본 논문과 같이 분석하고 분류하여 효과적으로 표현할 수 있다면 어떤 공격 시나리오가 생성 될 수 있는지 예상 할 수 있다.

표 3. 일반화된 공격 수행 분류
Table 3. Generalized Attack Behavior Steps

수행 단계	공격 수행 단계 시나리오
로그인	정상적인 로그인
로그인	무차별 공격을 통한 로그인
로그인	세션 가로채기를 이용한 로그인

3-2 웹 공격 표현

웹 공격은 시나리오가 존재한다. 이를 보다 쉽고

자세하게 표현하기 위해 본 논문에서는 STATL[8] 기법을 확장하여 공격 시나리오를 효과적으로 표현한다.

공격 수행 단계에 따라 각 공격 시나리오를 WxY와 같이 표현한다. 이때, W는 웹 공격의 종류, x는 공격 수행 단계, Y는 같은 공격 수행 단계에서 다른 공격 방법을 나타낸다. 예를 들면, 크로스 사이트 스크립트의 공격 시나리오를 분석하여 분류한 표 4에서 공격 수행 단계에 따라 각 공격 시나리오를 AxY와 같이 표현한다. “A”는 웹 공격의 한 종류인 크로스 사이트 스크립팅을 말한다. “x”는 표 2과 같이 공격이 수행되는 단계에 따른 분류를 1, 2, 3.... 으로 표현하며, “Y”는 표 3와 같이 같은 공격 수행 단계

에서의 공격 방법을 1, 2, 3.... 으로 표현한다. 즉, 표 4에서 열로 표현한 A11에서 A61은 크로스 사이트 스크립팅의 공격 수행 단계에 따라 표현한 것이고, 행으로 표현한 A11에서 A12는 같은 공격 수행 단계에서 다른 시나리오를 표현한다. 총 p개의 단계가 있고, i번째 단계에서 발생하는 경우의 수는 a_i이다 . 그러므로 전체 단계에서 발생하는 경우의 수의 합은 $\sum_{i=1}^{p} a_i$이다. 이때 p, i, a는 정수이다. 즉, 표 4에서는 12가지의 경우가 존재한다.

공격 시나리오를 분석 및 분류한 표 4와 공격 수행 단계 진행시 필요한 조건 표 5를 이용하여 다음과 같이 공격을 수행한다.

1. 시작 : W1P 시행
2. 진행 :

a. W1P 단계가 JW1PW2P 조건에 맞아 공격이 성공하면 지정된 다음단계인 W2P로 진행

if W1P 단계가 JW1PW2P 조건에 맞지 않아 공격이 실패하면

than 지정된 다음 단계 W1q로 진행

b. 단계 Wxp가 JWXPW(X+1)P 조건에 맞아 공격이 성공하면, 지정된 다음 단계 W(X+1)P로 진행

if Wxp가 JWXPW(X+1)P 조건에 맞지 않아 실패면

than 지정된 다음 단계(WKq)로 진행 (단, k < x)

c. 공격이 최종적으로 성공하여, 끝나면 Exit

3. 끝 : 공격 성공

표 4. 공격 행위의 분석 및 분류
Table 4. Attack Behavior Analysis and Categorization

수행 단계	A_{XY}	공격 수행 단계 시나리오
공격 목표 설정	A_{11}	취약점 스캔을 통한 웹 어플리케이션 파악
	A_{12}	인터넷 검색을 통한 취약 페이지 검색
로그인	A_{21}	정상적인 로그인
	A_{22}	무차별 공격을 통한 로그인
입력 대상 선정	A_{31}	게시판 사용
	A_{32}	URL의 CGI 인자에 삽입
입력 방법	A_{41}	악성코드 삽입
	A_{42}	Java Script를 이용한 입력
실행	A_{51}	사용자 클릭
	A_{52}	페이지 오픈으로 자동 실행
공격 성공	A_{61}	쿠키 획득
	A_{62}	피해자의 위조된 사이트 방문

본 논문에서는 웹 공격 단계를 정형화하기 위해 다음 공격 수행 단계로 진행하기 위한 조건을 JWXYW(X+1)P로 표현한다. 이는 WXY의 공격 수행 단계에서 W(X+1)P로의 공격 수행 단계로 진행하기 위한 조건을 표현한다. 즉, A11에서 A21로 공격이 수행된다고 할 때, 필요한 조건을 본 논문에서는 JA11A21로 표현한다. 표 5에서는 다음 공격 수행 단계로 진행하기 위한 조건을 표현한다.

다음 공격 수행 단계로 진행하기 위한 조건은 수식으로 다음과 같이 표현한다.

표 5. 공격 수행에서의 필요 조건
Table 5. Necessary Cnditions for an Attack Execution

	A_{XY} -> $A_{(X+1)P}$을 위한 조건	조건이 맞지 않는 경우 수행 할 수 있는 공격 단계
$JA_{12}A_{21}$	알려진 취약점 리스트 필요	
$JA_{21}A_{31}$	로그인이 가능한 계정 정보를 가지고 있음	A_{11}, A_{22}
$JA_{22}A_{31}$	로그인 실패 횟수에 제한이 없음	A_{11}, A_{12}, A_{21}
$JA_{31}A_{41}$	입력 폼 필드가 Tag로 존재	A_{11}, A_{12}, A_{21}, A_{22}, A_{32}
$JA_{22}A_{32}$	CGI 인자가 Get 방식으로 전달	A_{11}, A_{12}, A_{21}
$JA_{32}A_{42}$	Java Script Code 허용	A_{11}, A_{12}, A_{21}, A_{22}, A_{31}
$JA_{42}A_{52}$	사용자의 방문	A_{11}, A_{12}, A_{21}, A_{22}, A_{31}, A_{32}, A_{41}
$JA_{52}A_{62}$	사용자가 위조된 사이트 클릭	A_{11}, A_{12}, A_{21}, A_{22}, A_{31}, A_{32}, A_{41}, A_{42}, A_{51}

(1) $JW_{XY}V_{XY}$=조건, 조건이 맞을 경우 취할 수 있는 다음 공격 수행 단계

(2) $JW_{XY}V_{XY}\neq$조건, 조건이 맞지 않는 경우 취할 수 있는 최선의 공격 수행 단계

이 수식은 'WXY' 웹 공격 수행 단계에서 'VXY' 웹 공격 수행 단계로 진행하기 위한 조건에 충족되어 수행 할 수 있는 다음 공격 수행 단계를 '=' 을 이용하여 표현하며, 조건이 충족되지 않은 경우에 다시 시도해 볼 수 있는 공격 수행 단계를 '≠'을 이용하여 표현한다. 즉, A21에서 A31로 공격이 수행될 때 필요한 조건이 충족되어 다음 공격 수행 단계로 진행 할 수 있는 경우는 다음과 같이 표현한다.

$JA_{21}A_{31}$ =로그인이 가능한 계정정보, A31

만약, 조건이 충족되지 않는 경우 공격자는 다른 공격 방식을 시도 할 수 있으며, 이는 다음과 같이 표현한다.

$JA_{21}A_{31}$ ≠ 로그인이 가능한 계정정보, A22

이때, 우선적으로 같은 공격 수행 단계에서 시도할 수 있는 공격 방식을 선택하여 공격을 좀 더 빠르게 성공 시킬 수 있도록 한다. 즉, A21에서 A31로 공격이 수행 될 때 필요한 조건 JA21A31이 충족하지 않는 경우, 같은 공격 단계인 A22에서 공격을 다시 시도하는 것이 A11에서 시도하는 것보다 빠르게 성공 할 수 있다. 만약, 같은 공격 레벨에서 다른 공격 수행 단계가 없다면 바로 이전 단계의 공격 수행 단계에서 공격을 시도한다. 표 5에서 조건이 맞지 않는 경우 수행 할 수 있는 공격 단계는 오른쪽부터 수행할 때, 공격을 빠르게 성공 시킬 수 있다.

공격 수행 단계와 공격이 수행되기 위한 조건에 대한 연관성은 '⊗'을 이용하여 표현한다. 즉, 이전 공격 단계인 WXY에서 조건 JWXYW(X+1)P가 충족되어 이후 공격 단계인 W(X+1)P로 공격이 이루어 질 수 있는 경우에 수식으로 다음과 같이 표현한다.

$$W_{XY}\otimes JW_{XY}W_{(X+1)P}\otimes W_{(X+1)P}$$

즉, A21의 공격 단계에서 A31로의 공격 수행 단계로 진행하기 위한 'JA21A31 = 로그인이 가능한 계정정보' 조건이 충족되어 공격이 성공한 경우, 현재의 공격 수행 단계는 A31임을 다음과 같이 표현한다.

본 논문에서는 그림 1과 공격을 제안한 표현 방법을 이용하여 효과적으로 표현하고자 한다.

그림 1은 시나리오에 따라 크로스 사이트 공격 수행 단계를 표현하고 있으며, 점선을 통한 설명으로 공격 수행 조건을 나타낸다. 그림 1에 대한 시나리오는 공격자가 크로스 사이트 스크립팅 공격을 수행하던 중 게시판에 파일을 첨부하는 방식으로 공격을 시도하려고 하였으나, 파일 첨부가 되지 않자 공격자는 게시판 대신 URL의 CGI 인자에 Java Script를 삽입하여 피해자에게 위조된 사이트를 방문하도록 하는 공격 시나리오이다.

공격 행위 분석 및 분류에 대해서는 표 4와 표 5를 이용하여 공격자가 인터넷을 통해 취약 페이지를 검색한 후(A12), 무차별 공격을 통한 로그인 시도 할 수 있음(A22)을 $A_{12}\otimes JA_{12}A_{22}\otimes A_{22}$ 와 같이 표현한다.

공격자가 무차별 공격을 통한 로그인 시도(A22)를 성공하여 게시판에 파일 첨부를 시도(A31)하려고

하였지만, 공격이 실패하여 URL의 CGI 인자에 Java Script를 삽입(A32)하는 공격을 시도하는 경우의 표현은 다음과 같다. 다음 수식에서 괄호 부분은 먼저 공격이 이루어짐을 나타낸다.

즉, 수식에서 A31의 공격 수행 단계를 성공하기 위해서는 $A_{12} \otimes JA_{12}A_{22} \otimes A_{22}$ 부분이 먼저 성공해야 함을 표현한다.

$$(A_{12} \otimes JA_{12}A_{22} \otimes A_{22}) \otimes JA_{22}A_{31} \otimes A_{31} \neq A_{32}$$

수식에서 가장 오른쪽에 위치한 공격 수행 단계가 현재 공격 위치를 말하며, 그 공격이 실패할 경우에 '≠'을 이용하여 다른 공격 수행 단계를 지정해 준다. 또한 '≠'이 없는 경우에는 마지막 단계로 공격이 가능하다는 것을 나타낸다. 아래의 마지막 수식에서 A62는 공격이 성공하였음을 나타내므로, 최종적으로 공격이 성공하였음을 표현한다.

그림 1에서 표현한 크로스 사이트 스크립팅의 공격이 최종 공격자의 공격 목표를 수행하는 경우의 표현은 다음과 같다.

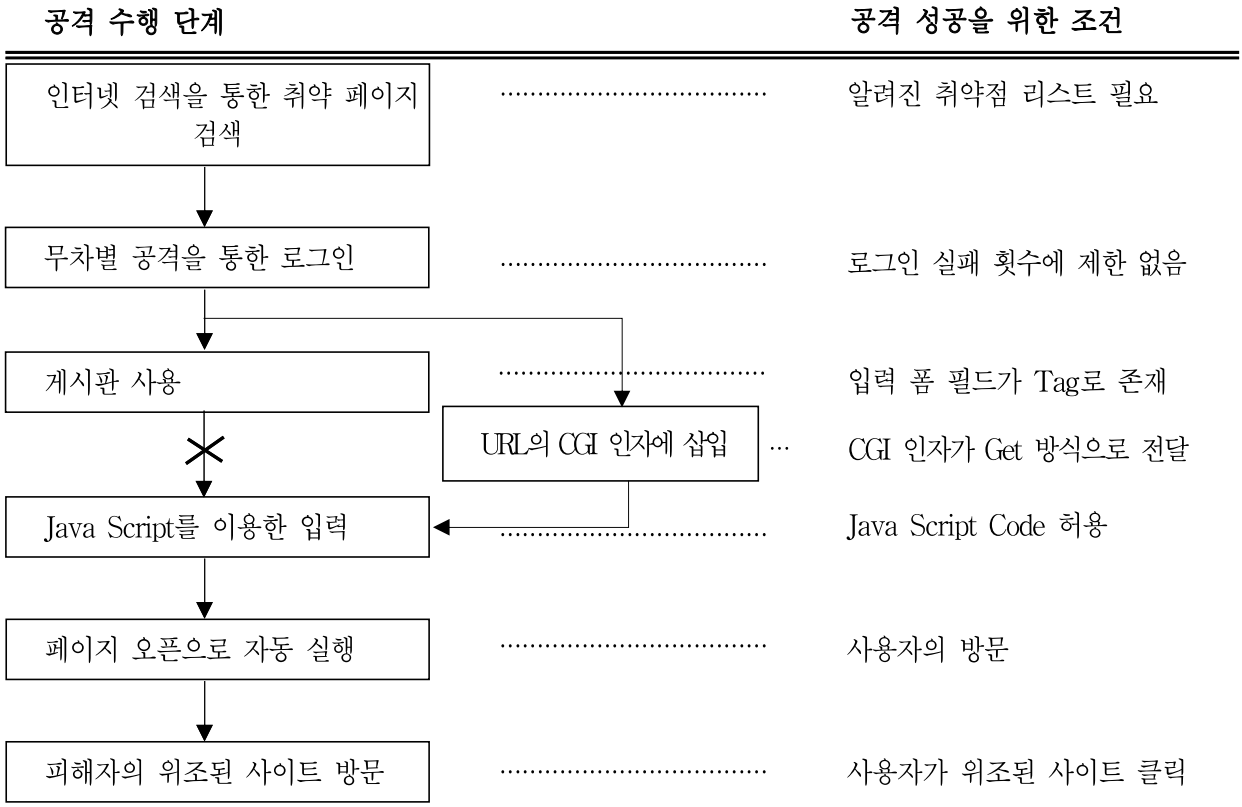

그림 1. 크로스사이트 스크립팅 공격 시나이로
Fig. 1. Attack Scenario of a Cross Site Script

다른 공격 방법으로 URL의 CGI 인자에 Java Script를 삽입(A32)하는 공격이 성공하여, 피해자가 방문하였을 경우 자동 실행될 수 있도록 하는 공격(A42)을 시도 할 수 있음을 다음과 같이 표현한다.

$$(((A_{12} \otimes JA_{12}A_{22} \otimes A_{22}) \otimes JA_{22}A_{32} \otimes A_{32}) \otimes JA_{32}A_{42} \otimes A_{42}$$

$$((((A_{12} \otimes JA_{12}A_{22} \otimes A_{22}) \otimes JA_{22}A_{32} \otimes A_{32}) \otimes JA_{32}A_{42} \otimes A_{42}) \otimes JA_{42}A_{52} \otimes A_{52}) \otimes JA_{52}A_{62} \otimes A_{62}$$

이는 다양한 공격 방법에 대한 공격 시나리오를 보다 효과적으로 표현 할 수 있다.

Ⅳ. 여러 가지 공격의 표현 및 생성

4-1 공격 표현 및 생성

최근의 웹 공격은 비슷한 행위의 여러 가지 공격 방법을 사용하는 공격을 시도하고 있다. 따라서 공격자들이 여러 가지 공격 방법들을 어떻게 사용하여 공격을 생성하는지 알아 볼 필요가 있다. 이번 장에서는 이러한 점을 파악하기 위해 2장에서 설명한 웹 공격을 일정한 기준에 의해서 분류하고, 발생 가능한 시나리오를 생성하여 효과적으로 표현하고자 한다.

표 6. 공격 행위 분석 및 분류
Table 6. Attack behavior analysis and categorization

	W_{XY} -> $V_{(X+1)P}$ 을 위한 조건	조건이 맞지 않는 경우
$JA_{11}A_{21}$	알려진 취약점 리스트 필요	B_{11}
$JA_{21}A_{31}$	로그인 실패 횟수에 제한이 없음	A_{11}, B_{11}, B_{21}
$JA_{21}B_{21}$	사용자 로그인과 관련된 폼 존재	A_{11}, B_{11}
$JB_{21}A_{31}$	입력 폼 필드가 Tag로 존재	A_{11}, B_{11}, B_{31}, B_{32}
$JA_{31}A_{41}$	Java Script Code 허용	A_{11}, B_{11}, B_{31}, B_{32}
$JA_{41}A_{51}$	사용자의 방문	A_{11}, B_{11}, B_{31}, B_{32}, B_{41}
$JA_{51}A_{61}$	사용자가 위조된 사이트 클릭	A_{11}, B_{11}, B_{31}, B_{32}, B_{41}, B_{51}

표 7에서 'A'는 크로스 사이트 스크립팅을 'B'는 SQL 삽입 공격을 표현한다. 공격 시나리오에 따라 공격자는 무차별 공격을 통해 로그인(A21)을 시도하려고 하였으나, 로그인 실패 횟수에 제한이 있어 실패하였다. 그때 SQL 삽입 공격을 통해 로그인(B21)을 시도할 수 있음을 다음과 같이 표현한다.

표 7. 두가지 공격의 수행 단계의 필요 조건
Table 7. Necessary conditions for two different attacks

	W_{XY} -> $V_{(X+1)P}$ 을 위한 조건	조건이 맞지 않는 경우
$JA_{11}A_{21}$	알려진 취약점 리스트 필요	B_{11}
$JA_{21}A_{31}$	로그인 실패 횟수에 제한이 없음	A_{11}, B_{11}, B_{21}
$JA_{21}B_{21}$	사용자 로그인과 관련된 폼 존재	A_{11}, B_{11}
$JB_{21}A_{31}$	입력 폼 필드가 Tag로 존재	A_{11}, B_{11}, B_{31}, B_{32}
$JA_{31}A_{41}$	Java Script Code 허용	A_{11}, B_{11}, B_{31}, B_{32}
$JA_{41}A_{51}$	사용자의 방문	A_{11}, B_{11}, B_{31}, B_{32}, B_{41}
$JA_{51}A_{61}$	사용자가 위조된 사이트 클릭	A_{11}, B_{11}, B_{31}, B_{32}, B_{41}, B_{51}

본 장에서는 크로스 사이트 스크립팅 공격을 수행하기 위해 무차별 공격을 통한 로그인을 시도하려고 하였으나, 실패하여 SQL 삽입 공격을 통해 로그인을 성공하는 시나리오에 대해 표현한다. 로그인이 성공된 후 원래 목적인 크로스 사이트 스트립팅을 통해 사용자에게 위조된 사이트를 방문하도록 유도하는 것이다. 이와 같은 공격 시나리오를 표현하기 위해 표 6에서 공격 시나리오를 분석 및 분류하였고, 표 7에서는 공격 수행 단계에 필요한 조건을 표현한다.

$$A_{11} \otimes JA_{11}A_{21} \otimes A_{21} \neq B_{21}$$

SQL 삽입 공격을 통해 로그인이 성공되어 다음 공격 수행 단계로 진행 될 수 있음을 다음과 같이 표현한다.

$$(A_{11} \otimes JA_{11}B_{21} \otimes B_{21}) \otimes JB_{21}A_{31} \otimes A_{31}$$

이 점은 두 가지 공격 방법이 하나의 공격에 이용될 수 있다는 점을 보여주고 있다. 여러 공격 방법들이 하나의 공격에 이용 될 수 있음을 파악하고 대응한다면, 효과적으로 공격에 대응 할 수 있다.

공격 시나리오에서 공격자가 최종적으로 공격이 성공하였음을 다음과 같이 표현한다.

$$((((A_{11} \otimes JA_{11}B_{21} \otimes B_{21}) \otimes JB_{21}A_{31} \otimes A_{31}) \otimes JA_{31}A_{41} \otimes A_{41}) \otimes JA_{41}A_{51} \otimes A_{51}) \otimes JA_{51}A_{61} \otimes A_{61}$$

이번에는 크로스 사이트 스크립팅, SQL 삽입, 디렉토리 노출 공격 기법을 복합적으로 사용하여 공격을 하는 공격 시나리오에 대해 표현하고자 한다.

표 8 : 공격 행위 분석 및 분류
Table 8 : Attack behavior analysis and categorization

A_{XY}	크로스 사이트 스크립팅
A_{11}	인터넷 검색으로 취약 페이지 검색
A_{21}	무차별 공격을 통한 로그인
A_{31}	입력 폼 필드
A_{41}	악성코드 삽입
A_{51}	자동 실행
B_{XY}	SQL 삽입
B_{11}	취약점 스캔 이용
B_{21}	로그인 폼 이용
B_{31}	Query 변경
B_{41}	Table drop
B_{51}	DBMS의 오동작 유발
C_{XY}	디렉토리 노출
C_{11}	페이지 파라미터에 의한 입력
C_{21}	../ 을 이용한 탐색
C_{22}	유니코드 변환 이용
C_{31}	중요 파일 획득
C_{32}	디렉토리 구조 파악

위의 공격은 공격자가 인터넷 검색을 통해 취약점이 있는 웹 어플리케이션을 발견하여 SQL 삽입 방법을 이용하여 로그인을 시도한다. 그 웹 어플리케이션은 디렉토리 노출 공격에도 취약하여 디렉토리 구조를 쉽게 파악할 수 있다. 공격자는 이를 이용하여 서버에 영향을 줄 수 있는 디렉토리를 찾아 악성코드를 삽입하여 자동실행 되도록 하였다.

세 가지 공격 방법을 이용하는 공격자의 최종 공격 목표를 수행하는 경우 다음과 같이 표현한다.

$$((((A_{11} \otimes JA_{11}B_{21} \otimes B_{21}) \otimes JB_{21}C_{21} \otimes C_{21}) \otimes JC_{21}C_{32} \otimes C_{32}) \otimes JC_{32}A_{41} \otimes A_{41}) \otimes JA_{41}A_{51} \otimes A_{51}$$

이처럼 세 가지 공격 방법을 사용하여 공격을 시도 할 수 있다. 또한 표 9와 같이 디렉토리 노출 공격을 이용하여 클라이언트를 대상으로 하는 공격 방법이 서버를 대상으로 하는 공격으로 변할 수 있음을 나타내었다. 이는 여러 가지 공격 방법을사용하여 생성된 공격은 해당 공격에 대한 방어조치를 취하더라도, 다른 공격 방법을 이용하여 공격이 성공할 수 있음을 보여준다. 이해하기 쉽게 알려진 공격을 대상으로 하였으나, 보다 다양하고 복잡한 공격 방법들이 사용되어 공격이 시도 될 수 있음을 이해하고, 이와 같은 방법으로 복잡하고 다양한 공격 방법에 대한 공격 시나리오를 보다 효과적으로 표현할 필요가 있다.

4-2 STATL의 표현과 비교

STATL는 침입 탐지에 관한 공격 시그니처를 효과적으로 표현하기 위해, 상태와 전송 방법에 따라 시그니처를 분류하여 단계적으로 표현하였다.

STATL은

Scenario ::= { use LibraryID{',' LibraryID} ','} scenario ScenarioID ScenarioParameters] '{' [FrontMatter] {State | Transition | NamedAction} '}' {FunctionDefinition}

와 같이 공격 시나리오를 표현한다.

여기에서 라이브러리는 어플리케이션의 타입, 이벤트, 함수로 이루어진다. 공격 시그니쳐가 어떤 것

과 어울리고 어떤 것을 할 수 있는지를 상태와 전송을 통해 표현한다. 시나리오는 최소한 처음과 끝을 나타내는 상태와 전송을 가진다. 즉, 공격 시나리오를 표현하기 위해 공격 시그니쳐를 정의하여 상태를 표현하였으며, 공격 행동과 연관시켜 새로운 상태로 이동하기 위한 이벤트들을 정의하여 전송 방법을 표현하였다. 예를 들어 상태를 각 호스트로, 전송을 TCP로 본다면, 이 시나리오를 통해 호스트 간에 어떤 이벤트들이 발생하고 어떻게 전송되는지를 알 수 있다.

STATL은 위의 시나리오 표현 방법을 이용하여 침입 탐지에 대한 공격의 상태와 전송 방법에 대해 상태 다이어그램으로 표현이 가능하였지만, 이 공격 표현 방법으로는 전체적인 공격 시나리오에 대해 표현이 불가능하다. 따라서 본 논문에서는 이를 보완하여 공격 시나리오에 대해 각 단계별로 분류하여 공격 방법들을 표현하고, J를 이용하여 각 공격 단계마다 필요한 공격 조건들을 표현한다.

표 9. 세가지 복합 공격의 수행에서의 필요 조건
Table 9. Necessary Cnditions for three different mixture attack execution

	W_{XY} -> $V_{(X+1)(Y+1)}$ 을 위한 조건	조건이 맞지 않는 경우
$JA_{11}B_{21}$	알려진 취약점 리스트 필요	B_{11}, C_{11}
$JB_{21}C_{21}$	사용자 로그인과 관련된 폼 존재	A_{11}, A_{21}, B_{11}, C_{11}, C_{21}
$JC_{21}C_{32}$	../에 대한 필터링 없음	A_{11}, B_{11}, C_{11}, C_{22}
$JC_{32}A_{41}$	접근 권한이 있음	A_{11}, A_{21}, B_{11}, B_{21}, C_{11}
$JA_{41}A_{51}$	악성 코드 삽입 허용	A_{11}, B_{11}, C_{11}

본 논문에서 제안하는 표현 방법은 STATL에서 제안하는 것과 같은 공격 시그니쳐에 대한 표현 방법이 아니라, 웹 공격에 대한 전체적인 흐름을 볼 수 있도록 하였다. 또한, 추후에 제안하는 논문을 시스템에 적용시킬 수 있도록 하였다. 예를 들면, 표를 통해 분류한 공격 수행 단계와 조건을 데이터베이스화 시킬 수 있도록 표현하였다. 또한, 데이터베이스화 된 공격 시나리오를 미리 예측하여 알려지지 않은 공격에 대해 대응 할 수 있다. 또한, 이를 이용하여 여러 가지 공격을 혼합한 새로운 공격에 대해 효과적으로 표현한다. 전체적인 흐름과, 여러 가지 공격 방법을 이용하여 공격이 생성 될 수 있음을 보였다. STATL에서 시나리오를 표현하는 방법을 기초로 하였지만, 구별되는 표현 방법이다.

Ⅴ. 결 론

본 논문에서는 웹 공격의 특성을 파악하기 위해 웹 공격들을 일정한 기준에 의해서 분석하고 분류하여 발생 가능한 시나리오를 효과적으로 표현하는 방법을 제안하였다. 제안하는 방법은 기존에 발생된 공격을 세부적으로 표현하고, 여러가지 알려진 공격을 사용하여 새로운 공격을 생성하여 표현하였다.

본 논문에서는 공격 수행 단계에 따라 각 공격 시나리오를 WxY와 같이 표현하였다. W는 웹 공격의 한 종류를 말하며, x는 공격이 수행되는 단계, Y는 같은 공격 수행 단계에서의 공격 방법을 표현하였다. 즉, WX1에서 W(X+1)1은 공격 수행 단계에 따른 시나리오를 단계별로 분류하여 표현한 것이며, W1Y에서 W1(Y+1)은 같은 수행 단계에서 나타날 수 있는 여러 가지 공격 방법을 보여준다. 또한 다음 공격 수행 단계로 진행하기 위한 조건을 JWXYW(X+1)P로 표현하였다.

예제를 통해 본 논문에서는 여러 가지 공격 방법을 사용하여 생성된 공격은 해당 공격에 대한 방어조치를 취하더라도 우회하여 공격이 성공할 수 있음을 보여주었다. 다양한 공격이 혼합되어 시도 될 수 있음을 이해하고 이와 같은 방법으로 복잡하고 다양한 공격 방법에 대한 공격 시나리오를 보다 효과적으로 표현하였다.

과거 웹 어플리케이션에 대한 공격 시도를 살펴보더라도 다른 시스템 공격이나 네트워크 공격 등과

다르게 기존의 공격 방법에 기반 하는 공격이 대부분이다. 즉 기존 공격을 이용하여 차단되거나 공격자의 성향에 따라 새로운 공격을 여러 가지로 조합하는 것이다. 이러한 점을 이용하여 추후에 웹 공격 기법에 적용하는 것 이외에도 다른 공격 기법에도 적용하여 알려지지 않은 공격에 대한 효과적인 보안 대책을 세울 수 있을 것이다.

감사의 글

본 논문은 한신대학교 학술연구비 지원에 의하여 연구되었음.

참 고 문 헌

[1] 심기명, “최신 웹 해킹 대응 및 개인정보보호 보안 기술”, *정보통신연구진흥원,* 2007

[2] Steve Pettit, Sanctum Inc, “Anatomy of a web application: Security considerations”, *Sanctum*, 2001

[3] Philipp Vogt, Florian Nentwish, Nenad Jovanovic, Engin Kirda, Christopher Kruegel, and Giovanni Vigna, "Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis.", *In Proceedings of the 14th annual Network and Distributed System Security Conference*, 2007

[4] Open Web Application Security Project(OWASP), “OWASO Top 10 2007”, http://www.owasp.org, 2007

[5] MITRE, “Vulnerability Type Distributions in CVE”, http://cwe.mitre.org/ documents/vuln-trends/ index.html 2007

[6] Jeongseok Seo, Han-Sung Kim, Sanghyun Cho and Sungdeok Cha, "Web Server Attack Categorization Based on Root Causes and Their Locations", *The International Conference on Information Technology: Coding and Computing,* 2004

[7] Xinming Ou, Wayne F. Boyer, Miles A. McQueen, "A Scalable Approach to Attack Graph Generation.", *Conference on Computer and Communications Security*, 2006

[8] Steven T. Eckmann, Giovanni Vigna and Richard A. Kemmerer, "STATL: An attack language for state-based intrusion detection", *Journal of Computer Security,* 2002

[9] Mike Andrews, James A. Whittaker, “How to Break Web Software: Functional and Security Testing of Web Applications and Web Services”, *Addison-Wesley Professional, February* 2006.

[10] Web Application Security Consortium(WASC), “Web Application Security Consortium : Threat Classification”, www.webappsec.org, 2004

[11] G.A. Di Lucca, A. R. Fasolino, M. Mastroianni, P. Tramontana "Identifying Cross Site Scripting Vulnerabilities in Web Applications", *Sixth IEEE International Workshop on Web Site Evolution*, 2004

[12] Engin Kirda, Christopher Kruegel, Giovanni Vigna and Nenad Jovanovic, “Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks”, *Proceedings of the 2006 ACM symposium on Applied computing,* 2006

[13] Omar ISMAIL, Masashi ETOH, Youki KADOBAYASHI, Suguru YAMAGUCHI "A Proposal and Implementation of Automatic Detection/Collection System for Cross-Site Scripting Vulnerability", *The 18th International Conference on Advanced Information Networking and Application IEEE,* 2004

[14] William G. J. Halfond and Alessandro Orso, “AMNESIA: Analysis and Monitoring for NEutralizing SQL Injection Attacks”, the 20th *IEEE/ACM international Conference on Automated software engineering,* 2005

[15] Gregory T. Buehrer, Bruce W. Weide, and Paolo A. G. Sivilotti, "Using parse Tree Validation to Prevent SQL Injection Attacks", *The 5th international workshop on Software engineering and middleware,* 2005

[16] Gregory T. Buehrer, Bruce W. Weide, and Paolo A. G. Sivilotti, "Using parse Tree Validation to Prevent SQL Injection Attacks", *The 5th international workshop on Software engineering and middleware,* 2005

[17] Chris Anley, "Advanced SQL Injection In SQL Server Applications", *Next Generation Security Software Ltd,* 2002

[18] Imperva, "Directory Traversal", http://www.imperva.com/resources/glossary/directory_traversal.html. 2007

[19] Guofei Jiang, "Microsoft IIS 4.0/5.0 Extended Unicode Directory Traversal Vulnerability", *Institute for Security Technology Studies, Dartmouth College,* 2000

[20] Open Web Application Security Project(OWASP), "Testing for Directory Traversal", *Open Web Application Security Project,* 2007

이 창 훈 (李昌勳)

2003년 2월 : 고려대학교 정보보호대학원 (공학석사)

2008년 2월 : 고려대학교 정보경영공학전문대학원 정보보호전공 (공학박사)

2008년 4월~2008년 12월 : 고려대학교 정보보호연구원 연구교수

2009년 3월~현재 : 한신대학교 컴퓨터공학부 전임강사

관심분야 : 정보보호, 암호학, 디지털 포렌식