# 로컬 프록시를 활용한 악성 스크립트 실행 방지 기술

오상환*, 윤수진**, 배한철**, 김환국***
*한국인터넷진흥원
osh1, sjyoon, hcbae, rinyfeel@kisa.or.kr

# Malicious Script Execution Prevention Technology Using A Local Proxy

Sang-Hwan Oh*, Soo-Jin Yoon**, Han-Chul Bae**, Hwan-Kuk Kim***
Korea Internet & Security Agency

요 약

기존의 HTML의 경우 미디어 재생, 3D 그래픽 처리, 웹 소켓 통신 등을 위해서는 Silverlight나 Active-X 와 같은 비 표준 플러그인을 설치해야만 구현할 수 있었다. 하지만, 차세대 웹 표준인 HTML5에서는 별도의 플러그인 설치 없이 자바스크립트의 신규 기능만으로 미디어 재생 등과 같은 동적인 기능을 구현할 수 있다. 이처럼 HTML5에서 자바스크립트 기능이 강화됨에 따라 이를 악용한 신종 사이버 공격의 위협이 증가 하고 있다. 기존의 악성 코드 공격과는 달리 사용자 단말에 어떠한 감염도 없이, 악성 스크립트를 삽입한 웹 페이지 접속만으로 악성 행위를 유발하기 때문에 기존의 보안 기술로는 탐지에 한계가 존재한다. 이에 본 논문에서는 로컬 프록시를 활용하여, 사용자 단말에서 송/수신되는 HTTP 트래픽을 수집하고, 이를 분석하여, 악성 자바스크립트를 탐지 및 차단하고, 나아가서 난독화 된 악성 자바스크립트를 탐지하는 방법을 제안하고자 한다.

## 1. 서론

2015년 10월에 W3C에서 발표한 차세대 웹 표준 HTML5는 기존의 HTML과의 호환성을 유지하면서, 비표준 플러그인들을 대체할 수 있는 다양한 기능들이 추가되었다.[1] 그래픽처리, 멀티스레드, 웹 소켓 통신 등을 신규 기능을 활용하여 구현할 수 있게 되었다. 이러한 흐름에 맞춰 세계적인 브라우저 개발사들도 앞 다투어 HTML5 지원 기능을 자사 브라우저에 도입하기 시작했다. 그 중 Google 의 경우 자사 웹 브라우저인 크롬과 파이어폭스의 동영상 재생을 위해 HTML5 플레이어를 강제로 적용한다는 방침을 세우고 있다. 이와 같이 HTML5로의 전환이 필수 불가결하게 되는 상황에서 HTML5의 보안 위협이 증가하고 있다. 신규 태그인 video, audio 등을 사용한 XSS(Cross Site Scripting)공격의 경우 패턴 매칭을 하는 XSS 공격 탐지 필터를 우회할 수 있다. 특히, 다양한 신규 기능들 중 핵심이 되는 자바스크립트에 대한 위협이 증가하고 있다. 2014년 중국 동영상 공유 사이트인 소후 티비 등에서 스크립트 기반 공격들이 발생하고 있다. 스크립트 기반의 사이버 공격의 경우 브라우저가 종료 되면, 사용자 PC에는 어떠한 흔적도 남지 않게 되어, 기존의 보안 기술로는 대응에 큰 어려움이 있다. 또한, 대다수의 자바스크립트는 난독화 되어 사용되기 때문에 기존의 시그니처 기반의 정적 탐지 기술을 우회하기가 용이하다. 이에 본 논문에서는 로컬 프록시 기술을 사용하여 악성 자바스크립트를 탐지하고, 실행을 방지하는 기술과 난독화 된 악성 자바스크립트에 대응하는 기술을 설명한다.

## 2. 본론

본 논문에서 설명할 기술은 로컬 프록시로 구현 된 분석 대상 수집/처리 모듈과 악성을 판별하는 분석 에이전트 모듈로 구성된다. 이 기술은 클라이언트의 웹 브라우저와 외부 웹 서버 사이에서 동작하며, 해당 위치에서 HTTP 패킷을 수집 및 분석을 수행하고, 악성으로 판별 된 경우 악성 요소를 삭제하거나, 안전한 페이지로 리다이렉션을 통해 스크립트 기반의 공격으로부터 클라이언트를 보호하는 기술이다. 분석 대상 수집/처리 모듈은 웹 브라우저에서 발생하는 패킷으로부터 URL 정보를 수집 하고, 웹 페이지 수신 패킷을 확인하여 스크립트를 추출하는 기능, 분석 에이전트로부터 분석 결과를 전송 받아서 악성인 경우에는 실행 방지를 위해 후 처리 기능으로 나뉜다.

분석 에이전트 모듈은 시그니처 매칭을 통한 악성 스크립트를 판별하는 악성 스크립트 처리 기능과, 악성 URL을 판별하는 악성 URL 분석 기능, 웹 브라우저 프로세스의 메모리 스캔을 통한 메모리 검사 기능, 탐지 정보를 사용자에게 알리고 탐지 설정 및 시스템 설정을 관리하는 탐지 알림 및 사용자 GUI 기능으로 구성된다.

분석 대상 수집/처리 모듈은 사용자 웹 브라우저에 외부 웹 서버로 요청하는 HTTP Request 패킷으로부터 접속 URL 정보를 추출하여, 분석 에이전트로 요청을 한다. 해당 URL이 안전으로 판단 된 경우 해당 패킷을 웹 서버로 전송한다. 이후, 웹 서버로부터 수신한 HTTP Response 패킷으로부터 HTML 문서 내에 있는 모든 스크립트 정보를 태그 단위로 추출한다. 추출한 스크립트 중 src 속성 등을 이용해 스크립트 코드 전문이 외부에 있는 외부 스크립트의 경우 해당 경로로 리소스를 요청하여, 해당 리소스를 수집한다. 수집한 스크립트를 분석 에이전트로 분석 요청하고, 분석 결과에 따라 후 처리를 진행한다. 정상인 경우 별도의 처리 없이 진행하게 되고, 악성인 경우 두 가지 후 처리를 실행한다. 첫 번째, HTML 문서 내에 악성 스크립트 코드가 있는 경우에는 해당 웹 페이지를 안전한 웹 페이지로 강제 리다이렉션을 실행한다. 두 번째, src속성 등을 이용하여 HTML 문서 외부에 악성 스크립트 코드가 있는 경우, 해당 요청 코드를 삭제하고, 그 외에 안전한 컨텐츠만을 웹 브라우저로 보냄 으로서 악성 스크립트로부터 클라이언트 웹 브라우저를 보호하게 된다.

분석 에이전트 모듈은 크게 실시간 분석과 정밀 분석 두 가지 분석을 수행한다. 먼저, 실시간 분석은 URL 검사와 스크립트 정적 분석으로 구성 되어 있다. URL 검사는 자체적으로 보유한 URL 블랙리스트와 외부에서 제공하는 URL 블랙리스트를 확보하여, HTTP Request 패킷으로부터 추출한 접속 URL과의 매칭을 통해 악성을 판별한다. 만약, 악성 URL로 판별 된 경우 그 뒤의 분석 프로세스는 실행하지 않고, 안전한 페이지로 리다이렉션을 진행 한 뒤 분석을 종료하게 된다. URL이 정상인 경우, 해당 웹 서버로 패킷을 전송하고, 웹 서버로부터 수신한 패킷에서 스크립트를 추출하여 스크립트 정적 분석을 진행한다.

정적 분석은 YARA[2] 기반의 시그니처 패턴 매칭을 통해 악성을 판별한다. YARA는 악성을 식별하고 구별하는 기술로서, 가장 큰 특징으로는 다양한 표현을 담을 수 있는 Rule Set을 생성할 수 있다. 일반적인 텍스트 문자열 뿐만 아니라 HEX 문자열 그리고 정규표현식을 생성할 수 있다. 이러한 YARA를 이용하여  이전에 발생된 악성 스크립트의 코드 패턴 정보를 추출하여 정적 분석용 시그니처로 활용한다. 이러한 정보가 대상 스크립트에 있는지 여부를 통해 악성을 판별한다. 그러나 난독화 된 악성 스크립트를 시그니처 패턴 매칭을 기반으로 하는 정적 분석에서 탐지하기에는 많은 어려움이 있다. 이러한 난독화 된 악성 스크립트 탐지를 위해 정밀 분석 기술이 도입 되었다. 정밀 분석은 크게 스크립트를 추출하는 메모리 스캔과 난독화를 해제하여 원본을 추출하는 난독화 분석으로 구성되어 있다. 1단계인 메모리 스캔은 실행 중인 웹 브라우저의 PID를 추출하고, 분석 대상 프로세스의 메모리로부터 스크립트 정보를 추출한다. 이 시점에 난독화 해제 루틴이 실행된 난독화 스크립트의 경우 원본 스크립트를 추출할 수 있다. 1단계에서 풀리지 않은 난독화를 2단계인 난독화 분석을 진행하게 된다. 난독화 분석의 핵심 요소는 모든 난독화 스크립트는 브라우저의 자바 스크립트 엔진에서 실행되는 점을 이용하였다. 크롬 브라우저에서 사용되는 V8 자바 스크립트 엔진[3]을 커스터 마이징 하여 난독화를 해제한다. 난독화 된 스크립트를 V8 엔진에 넣고 실행한 뒤, 난독화가 해제 된 시점에 BreakPoint를 설정하여 해제된 원본 스크립트를 추출하게 된다. 추출한 원본 스크립트와 정적 분석에서 사용되는 시그니처와 패턴 매칭을 통해 악성을 판별하게 된다.

## 3. 결론

W3C에서 발표한 HTML5의 등장으로 기존의 HTML에 비해 자바 스크립트의 기능이 대폭 향상 되었다. 하지만, 그와 함께 자바 스크립트를 악용한 위협도 크게 증가하고 있다. 본 논문에서는 브라우저 프록시를 통해서 사전에 탐지 및 차단하였다. 하지만, 본 논문에서 제안한 기술에도 몇 가지 한계점이 존재한다. 먼저, 정밀 분석에서 사용된 난독화 탐지 기술의 경우, Base62나 Base64 인코딩 등 잘 알려진 난독화 방법에 대해서는 높은 탐지율을 보였으나, 새로운 난독화 방법을 사용한 경우에는 탐지에 어려움이 발생하였다. 또한, 이 기술은 클라이언트 중에서 PC환경을 타겟으로 제안하였기 때문에 모바일 환경에서의 탐지는 불가능하다. 모바일 기기 이용증가로 인해 공격자들은 모바일 기기를 통해 악성 행위를 유발하는 공격으로 진화하고 있다. 이러한 한계 사항을 보완하기 위해 난독화 탐지 알고리즘에 대한 심층 연구를 수행할 예정이며, 모바일 환경에서 악성 스크립트를 탐지할 수 있는 기술 연구를 도출할 예정이다.

## ACKNOWLEDGMENT

이 논문은 2014년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0101-15-0230, 스크립트 기반 사이버 공격 사전 예방 및 대응 기술 개발)

## 참고문헌

[1] W3C. "HTML5 Standard", 2015.04.20. http://www.w3.org/standards

[2] YARA Documentation. "http://yara.readthedocs.org/en/latest/index.html"

[3] Chrome V8, "https://developers.google.com/v8"