감염된 TTF(True Type Font)파일에 대한 연구 및 대응방법

박연진, 오주혜, 이근호
백석대학교 정보통신학부
e-mail: h_tea@naver.com, oho166@naver.com, root1004@bu.ac.kr

A Study and Countermeasure of the Infected TTF(True type Font) Files

Yeon-Jin Park, Ju-Hye Oh, Keun-Ho Lee
Division of Information and Communication, Baek-Seok University

요 약

최근 정보통신 기술의 발전과 함께 M2M(Machine-to-Machine) 산업분야의 시스템이 다기능 고성능화 되고 있으며 IoT(Internet of Things), IoE(Internet of Everything)기술 등과 함께 많은 발전해가고 있다[1]. 가장 오래 된 정보통신 기술의 근간인 웹 애플리케이션은 점점 발전하고 고도화 되어가고 있으며 이러한 웹 애플리케이션을 해킹하는 기술도 다양한 관점에서 발전하고 있다. 웹 애플리케이션을 구성하는데 필수적인 파일인 TTF(True Type Font)파일에 대한 보안적 관심이 필요하다. TTF파일을 외부에서 받아옴으로서 웹 애플리케이션에 적용시키는 방식을 사용할 때, 다른 서버에서 URL을 통해 받아오는 TTF파일에 대해 보안적 검사가 제대로 실행되지 않는다. 본 논문에서는 TTF파일의 감염과 그 파일에 대한 대응 방법을 제안하고자 한다.

1. 서론

웹 애플리케이션이 널리 보급됨에 따라 웹 애플리케이션을 해킹하는 기술 또한 점점 발전되고 있다. 이러한 상황에서 웹 애플리케이션을 구성하는 구성요소들은 해킹의 위험이 있다고 볼 수 있다. TTF파일은 웹 애플리케이션을 구축하는 가장 처음에 명시되는 폰트파일이다. 최초로 소스코드 몇 줄로 폰트가 인식이 되며 그 이후에는 소스코드 내에서도 잘 언급이 되지 않아, 사용자에게 인지되는 일이 적기 때문에 주목 하기가 쉽지 않다. 웹 애플리케이션에 글꼴을 적용하기 위해서는 두 가지 방법 중의 하나를 선택할 수 있다. 첫 번째 방법은 클라이언트의 컴퓨터에 있는 폰트파일(Font File)을 적용시키는 것이다. 이러한 적용은 클라이언트 본인의 컴퓨터에만 적용되며, 해당 폰트가 깔려있지 않은 다른 클라이언트들은 애플리케이션이 제대로 보이지 않을 수 있는 인바운드(Inbound)적 적용이다. 두 번째 방법은 TTF파일을 외부에서 받아옴으로서 웹 애플리케이션에 적용시키는 아웃바운드(Outbound)적 적용이다. 아웃바운드 방식을 사용할 때, 보통 다른 서버에서 받아오는 TTF파일에 대해서 보안적 검사가 제대로 실행되지 않는다. 본 논문에서는 이러한 TTF파일을 검사 없이 받아오는 것의 위험성과 이러한 파일에 대한 대응방안을 논하고자 한다.

2. 관련연구

2.1. TTF 파일 포맷

트루 타입(TrueType)은 외곽선 글꼴 표준으로, 1980년대 말에 애플 컴퓨터가 어도비의 포스트스크립트에 쓰이는 글꼴에 대항하기 위해 개발하였다. 트루 타입은 글꼴이 어떻게 표시될 것인지에 대한 높은 수준의 제어를 할 수 있다는 것을 말한다. 이를 힌팅기술 혹은 힌팅 인스트럭션이라고 하며 글꼴 표준을 사용하는 파일 확장자를 ttf라고 한다. 적당히 힌팅된 트루타입폰트는 웹 사이트에 높은 가독성을 제공한다[2].

2.2. 샌드박스(SandBox)

샌드박스(sandbox)란 외부로부터 들어온 프로그램이 보호된 영역에서 동작해 시스템이 부정하게 조작되는 것을 막는 보안 형태이다. 대표적인 일례로 자바의 JVM(Java Vitual Machine)과 아이폰의 앱 샌드박스가 있다. 자바의 JVM은 자바(Java)가 지원하는 기본 보안 소프트웨어로, 외부에서 받은 프로그램을 JVM이라는 보호된 영역 안에 가둔 뒤 작동시키는 방법으로 프로그램이 폭주하거나 악성 바이러스의 침투를 막는다.

2.3. Javascript

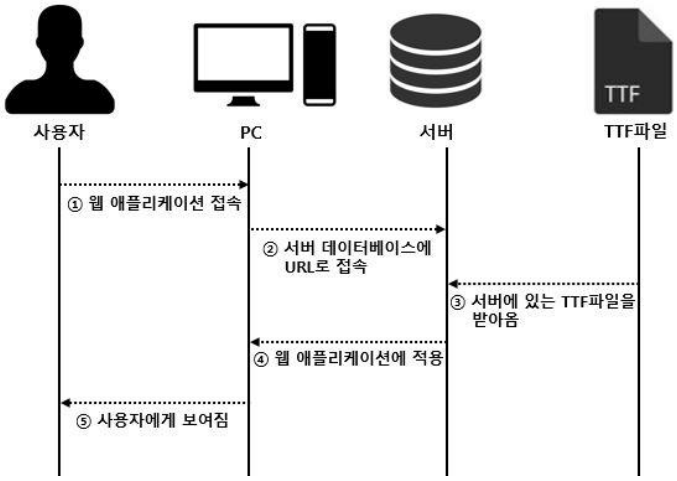
Javascript는 Visual Basic과 함께 사용되는 스크립트 언어이다. 이들 두 언어는 기본적으로 제공되는 스크립트

언어의 기능 외에 마이크로소프트에서 COM Object나 Active X Control등의 자원을 사용할 수 있도록 했다. 이를 이용해서 스크립트언어는 레지스트리 수정, Outlook등의 강력한 기능을 사용할 수 있게 되었다[3].

사용자의 허가 없이 스크립트를 실행하는 스크립트 언어의 강력성 때문에, Javascript에는 File I/O 기능이 없으며 Javascript를 웹 애플리케이션에서 구동할 때 샌드박스 처리를 함으로서 보안성을 구축하였다.

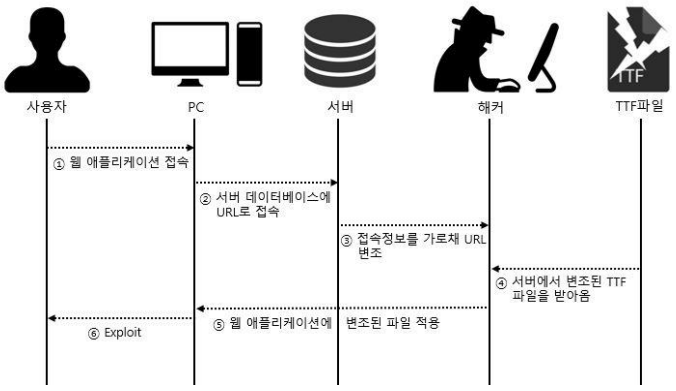
3. 감염 시나리오

일반적으로 TTF파일이 웹 애플리케이션에 적용되는 원리는 다음과 같다. 사용자가 웹 애플리케이션으로 접속을 하는 순간 웹 애플리케이션은 TTF파일을 저장한 서버의 데이터베이스에 접속한다. 이때 접속방식은 간단하게 URL을 추출해서 URL로 접속한다. 접속한 URL에서 따라온 TTF 파일을 웹 애플리케이션에 적용한다.



(그림1) 정상적인 웹 애플리케이션에서의 TTF파일동작

해커는 웹브라우저 사용 기록, 쿠키, 검색어 등을 타임라인, 필터 등으로 분석하여 사용자에 대한 정보와 패턴을 제공받는다[4]. 여기서 분석된 특정 URL로 TTF파일을 받으려 서버에 접속하는 순간 웹 프록시 프로그램을 이용하여 접속정보를 가로채 변조된 TTF파일이 저장되어있는 URL로 접속하게 한다.



(그림2) 변조된 TTF파일에 의한 Exploit

결과적으로 변조된 TTF파일이 다운로드 되어 클라이언트의 컴퓨터를 감염시킨다.

4. 활용공격

TTF파일은 3MB정도의 용량을 가지고 있지만 쉘 스크립트 등 많은 악성코드들을 삽입 할 수 있다. 대표적인 예로 백도어를 들 수 있다. 일명 좀비PC라고도 불리는 DDOS를 일으키는 백도어는 적은 용량으로도 큰 효율을 낼 수 있다. 백도어는 감염자의 컴퓨터를 지켜볼 수 있으며 감염자의 컴퓨터를 사용하여 2차 공격을 낼 수 있다는 것이 위협이라 볼 수 있다. 또, MBR을 망가트려 클라이언트의 컴퓨터 자체를 마비시키는 단순한 웜, 끊임없는 자체 증식으로 클라이언트의 컴퓨터 자원(가용 RAM, 저장 메모리 등)을 고갈시키는 공격 등을 야기 시키는 악성코드를 삽입 할 수 있다.

5. 대응방안

Javascript를 웹 애플리케이션에서 동작하면서 샌드박스 처리를 하는 것처럼, TTF파일을 샌드박스 처리를 하는것이 그 대응방안이 될 수 있다. 샌드박스 처리를 함으로서 샌드박스 내에서 TTS 파일이 어떠한 권한을 요구할 때, 웹 애플리케이션을 종료하는 알고리즘을 만들어 샌드박스를 구동하는 것이다. 이러한 알고리즘을 구상하면서 폰트파일에게 권한이 어느 정도가 필요한지 제대로 파악해서 솔루션을 구상해야만 한다.

```
1  알고리즘 : Sandbox(FontSandBox, FontFile ,List)
2  input          - FontSandBox : 현재 폰트를 검사하는 샌드박스 클래스
3                 - FontFile : 폰트파일 링크로부터 받아온 TTS File
4                 - List : TTS파일에 부여하는 권한 리스트
5
6             While ( 웹 어플리케이션 실행)
7                     if (폰트파일이 권한을 요구함)
8                         if (List내에 있는 권한임)
9                         Call WebApplicationFont
10                        elseif(List 내에 없는 권한임)
11                        Exit WebApplicationFont
12
```

(그림3) TTS파일을 검사할 수 있는 알고리즘

(그림3)은 제안된 기법의 pesudo-code로서 대략적인 구

상방안을 확인 할 수 있다. 웹 애플리케이션이 실행되어서 화면이 뜨기 전에 가장 먼저 이 알고리즘을 구동한다. TTS파일은 기본적으로 외부에 저장되어있는 서버에서 URL 형식으로 받아오기 때문에, 링크로 받아온 TTS파일과 TTS파일이 어떠한 권한을 가질 것인지에 대해서 리스트를 작성한다. (그림 3, 줄번호 3, 줄번호4) 폰트가 권한을 요구할 때, 웹 애플리케이션에서는 폰트가 요구하는 권한과 리스트와 비교한다.(그림3, 줄번호 8) 리스트에 있는 권한을 요청 했다면 웹 애플리케이션을 불러온다. (그림3 줄번호 9) 만일 리스트 내에 없는 권한이라면 확인 후 웹 애플리케이션을 종료한다. (그림3, 줄번호 11)

6. 결론

해킹에 대한 기사 및 관련 언론자료들이 서서히 노출됨에 따라 사용자들이 보안에 대해서 관심을 표하고 있다. 웹 애플리케이션을 구축할 때 가장 기본적으로 하는 몇 가지 서식중의 하나가 바로 글꼴 설정이다. 이러한 폰트에 대한 보안 이슈 및 폰트에 대한 솔루션은 웹 애플리케이션 제작자들에게도 큰 도움이 될 것이다. 특히 이런 폰트 파일의 권한에 대한 샌드박스 알고리즘의 도입은 많은 사용자에게 도움을 주며 보안에 대한 경종을 울릴 수 있을 것이다.

감사의 글

이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2013R1A1A1A05012348)

참고문헌

[1] 한군희, 배우식, “M2M 통신환경에서 안전한 P2P 보안 프로토콜 검증”, 디지털융합학회논문지, Vol. 13, No. 5, pp. 213-218, 2015.

[2] 김용호, 이윤배, “인터넷에서의 개선된 벡터라이징 기법에 관한 연구”, 한국정보통신학회논문지, Vol. 6, No. 2, pp. 271-281, 2002.

[3] 윤영태, 예홍진, 조은선, 고재영, “코드 수정을 통한 스크립트 형태의 악성 이동 코드 대응 기법”, 한국정보과학회 학술발표논문집, Vol. 28, No. 2Ⅰ, pp. 727-729, 2001.

[4] 이준연, “디지털 포렌식을 위한 활성데이터 기반 증거분석도구 개발”, 디지털융복합연구, Vol. 10, No. 3, pp. 99-104, 2015.