Regular paper

Journal of the KIECS. pp. 303-308, vol. 14, no. 2, Apr. 30. 2019, t. 94, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2019.14.2.303

스프링 서버 원격코드 실행 취약점(CVE-2018-1270)을 이용한 응용 공격 시나리오의 대응 방안

정병문* · 장재열** · 최철재***

Countermeasure of an Application Attack Scenario Using Spring Server Remote Code Execution Vulnerability (CVE-2018-1270)

Byeong-Mun Jung* · Jae-Youl Jang** · Chul-Jae Choi**

요 약

스프링 프레임워크는 우리나라 공공기관의 웹서비스 개발도구의 표준이라 할 만큼 전자정부 프레임워크의 기반 기술로 많이 사용되고 있다. 그러나 최근 스프링 프레임워크를 이용한 애플리케이션에서 원격코드 실행 취약점(CVE-2018-1270)이 발견되었다. 본 논문은 스프링 프레임워크를 서버를 대상으로 발생한 취약점의 위험성을 해킹 시나리오 POC(Proof Of Concept)를 이용한 취약점 실험 분석 방법을 제안한다. 종국적 대응방안으로 버전 4.3.16와 버전 5.0.5 이상으로 패치를 제안한다. 아울러 제안한 해킹시나리오 취약점 실험분석이 보안 프로그램의 성능향상 및 새로운 인증체계의 구축을 위한 자료로 활용될 것으로 기대한다.

ABSTRACT

Spring framework is widely used as a base technology for e-government frameworks and to the extent it is a standard for web service development tools of Korean public institutions. However, recently, a remote code execution vulnerability(CVE-2018-1270) was found in an application using a spring framework. This paper proposes a method of analyzing the vulnerability experiment using a hacking scenario, Proof Of Concept(POC), in which the spring framework is a hazard to the server. We propose the patch to version 4.3.16 and version 5.0.5 or later as an ultimate response. It is also expected that the proposed experiment analysis on vulnerability of hacking scenario will be used as a data for improving performance of security programs and establishing a new authentication system.

키워드

CVE-2018-1270, E-Government, Remote Code Execution Vulnerability, Spring Framework, Stomp Protocol
CVE-2018-1270, 전자 정부, 원격 코드 실행 취약점, 스프링 프레임워크, 스톰프 프로토콜

Ⅰ. 서 론

스프링 프레임워크(spring framework)는 매우 편리한 서버 설계 구조이다. 2003년 6월 최초로 아파치 2.0 라이선스로 공개된 자바(java), 동적 웹 사이트 개발을 위해 만들어진 프레임워크이며 현재 대한민국

* 경동대학교 정보보안학과(jbm2112@naver.com)
** 경동대학교 정보보안학과(ccy@kduniv.ac.kr)
*** 교신저자 : 경동대학교 정보보안학과
· 접 수 일 : 2018. 11. 28
· 수정완료일 : 2019. 02. 05
· 게재확정일 : 2019. 04. 15
· Received : Nov. 28, 2018, Revised : Feb. 05, 2019, Accepted : Apr. 15, 2019
· Corresponding Author : Chul-Jae Choi
Dept. of Cyber Security for information, Kyungdong University.
Email : cj-choi@kduniv.ac.kr

전자정부 표준 프레임워크의 기반 기술로써 국내 기업들의 70% 이상이 스프링 프레임워크 구조를 사용하고 있다. Model, View Controller 3가지 개념으로 나누어 서버와 클라이언트 간의 서비스를 효율적으로 제공하며 소스코드의 유지보수와 재사용에 편리하다는 장점이 있다.

그러나 스프링 프레임워크는 다른 프레임워크와 마찬가지로 취약점이 존재한다. CVE-2018-1270은 스프링 프레임워크 4.3~4.3.15, 5.0~5.0.4 버전에서 웹 소켓 연결을 통한 스톰프 프로토콜(stomp protocol) 통신을 할 때 발생하는 취약점이다[1].

클라이언트 사이드에 공격자가 원하는 실행 코드를 삽입하여 웹 소켓 연결을 구성하면 서버에게 메시지(from client)를 전송할 때마다 서버에서 실행코드를 처리하는 과정에서 서버의 기능이 실행코드에 따라 공격자가 의도한 행위를 수행하게 된다.

스프링 프레임워크는 국내 기업의 70% 이상이 사용하고 있는 프레임워크인 만큼 서버가 공격자의 의도대로 동작한다는 것은 매우 치명적인 취약점이다.

따라서 본 논문에서는 우리나라 전자정부 프레임워크의 기반 기술인 스프링 프레임워크에서 발생하는 원격코드 취약점 CVE-2018-1270의 개념과 공격 시나리오 기반으로 발생 원인에 대하여 알아보고, 해결방안으로 스프링 프레임워크 버전 업그레이드를 통해 취약점이 어떻게 보안 조치가 되었는지 실험 분석하였다. 해결책으로 안전한 스프링서버의 운영을 위한 최신 스프링 프레임워크의 버전업 권면을 제안하고, 취약점 분석실험 방안을 제시한다.

본 논문의 구성은 1장 서론에 이어 2장에서는 CVE-2018-1270의 기본개념과 공격시나리오 및 시연을 통한 실험방법을 제시하며, 3장에서는 대응방안, 4장 결론으로 구성되어 있다.

II. CVE-2018-1270

2.1 용어 및 개념

CVE(:Common Vulnerabilities and Exposures)는 '정보보안 취약점 표준 코드'의 약자이다. 1999년, 미국 비영리 연구 개발 기관인 MITRE가 취약점들을 파악하고 보안 강화에 사용하는 무료 코드(Dictionary) 모듈로, 취약점 식별 방식의 표준화가 목적이다[1].

스프링 메시징 모듈은 스프링 프레임워크에서 지원하는 대화형 웹 서비스 모듈이며 메시징 아키텍처와 프로토콜을 지원한다. 그리고 스프링 웹 소켓(web socket) 모듈이란 스프링 프레임워크에서 지원하는 웹 소켓 모듈로 스톰프 프로토콜을 지원함으로써 보다 효율적인 메시지 전송을 지원하는 모듈이다.

2.2 선행 관련연구

지금까지의 연구는 웹 개발에 확장성이 높은 스프링 프레임워크 기반을 활용하는 연구[2-4], 웹 공격에 스프링 프레임워크를 활용하는 방안[5], 취약점 공격의 대응[6-8]연구, 네트워크상의 취약점[9], 취약점 사례연구[10]가 있으나 공공기관 웹 개발에 사용되는 스프링 원격코드 실행 취약점 CVE-2018-1270관련 연구논문은 전무하다.

2.3 정상 시나리오

취약점을 이용한 공격 가능성을 보여주는 시현코드를 POC(Proof of Concept)라고 한다. 그림 1은 테스트 페이지에 대한 POC의 정상적인 시나리오 단계적 절차를 보인다.

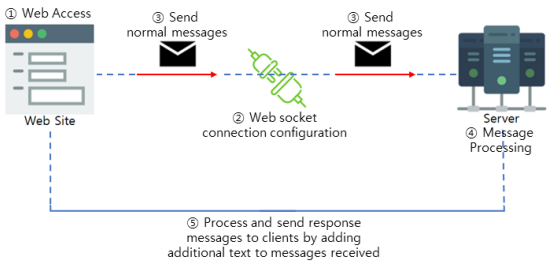

그림 1. 테스트 페이지 정상 프로세스
Fig. 1 Test page normal process

① 웹 사이트에 접속한다.
② 웹에서 서버로 웹소켓 연결을 요청하여 C/S 간의 웹소켓 연결이 구성된다.

1) http://www.itworld.co.kr/howto/108107

③ 서버로 메시지(from Client)를 전송한다.
④ 서버에서 메시지(from Client)를 받아 처리한다.
⑤ 메시지(from Client)를 처리하는 과정에서 메시지에 추가적인 문자열을 붙여 클라이언트에게 응답 메시지를 전송한다.
⑥ 클라이언트는 메시지(from Server)를 받아 출력한다.

2.4 공격 시나리오

어플리케이션이 간단한 메모리 안의 STOMP 브로커가 있는 웹 소켓 엔드 포인트를 통해 STOMP를 노출할 수 있다. 악의적인 공격자는 메시지를 브로커에게 전달하여 원격 코드 실행 공격을 유발한다[2].

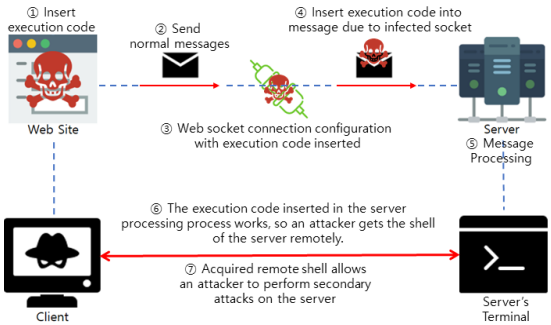

그림 2. 테스트 페이지 공격 시나리오
Fig. 2 Test page attack scenarios

① 공격자는 웹 사이트의 Web Socket 연결 소스에 서버와 공격자간의 원격 쉘 연결을 하는 악의적인 실행코드를 삽입하여 웹 사이트를 변조한다.
② 변조된 웹 사이트로 서버에 요청을 하면 서버와 클라이언트 간에 웹 소켓에 실행코드가 삽입된 채로 연결이 구성된다.
③ 클라이언트에서 전송한 메시지가 웹 소켓에 저장되어 있던 실행코드가 메시지에 삽입되어 서버로 전송된다.
④ 서버에서 메시지(from client)를 받아 처리한다.
⑤ 서버 내 메시지 처리과정에서 메시지 내부의 실행코드가 추출되어 실행된다.
⑥ 공격자가 삽입한 실행코드로 인해 공격자는 서버로부터 원격 쉘을 획득할 수 있다.
⑦ 공격자는 획득한 원격 쉘을 통해 서버 시스템 정보, 디렉토리 및 주요정보 열람, 서버 마비, 멀웨이(malware)를 통한 감염 등 2차적인 공격을 가할 수 있다.

III. POC 시연과 대응 방안

3.1 POC 시연 환경

POC 시연환경은 표 1의 VM, Test OS, 스프링 프레임워크 5.0.4, 네트워크 구성은 브리지를 사용하며 STS(:Spring Tool Suite) 서버 플랫폼을 사용한다.

표 1. 시연 환경
Table 1. Demonstration environment

VM OS	Ubuntu 16.04
Test OS	Windows 10
Spring Framework Version	5.0.4
Network Configuration	Bridge
Server Platform	Spring Tool Suite

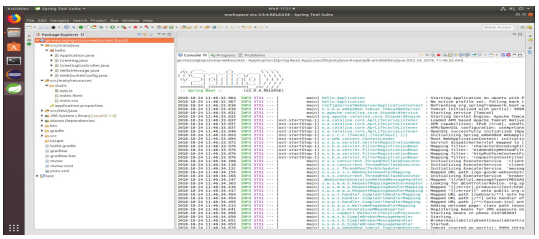

그림 3. 스프링 프레임워크 서버 구축
Fig. 3 Building a spring framework server

3.2 정상 페이지 및 기능

구축한 서버에 접속하면 다음과 같은 기능이 구현되어있는 웹 사이트에 접속이 가능하다.

그림 4. 테스트 페이지 기능
Fig. 4 Test page features

① Web Socket 연결 기능

2) https://nvd.nist.gov/vuln/detail/CVE-2018-1270

② 메시지 입력 및 전송
③ 서버로부터 처리된 응답 메시지

3.3 페이지 변조 및 공격

사이트에 접속할 때 app.js 소스가 클라이언트에게 전송된다. app.js는 웹 소켓 관련 함수가 구현되어있는 소스코드로, 크롬 개발자(chrome developer)의 console 기능을 통해 Web Socket 연결하는 함수인 connect() 함수에 실행코드를 삽입하여 현재의 웹 사이트를 변조한다(그림 5). 실행코드는 서버에서 원격 쉘을 공격자에게 제공하도록 명령하는 java 환경 내 어플리케이션 기능 실행 명령어이다.

```
var header =
{"selector":"T(java.lang.Runtime).getRuntime().exec('xter
m -display 공격자_IP:0.0')"};
```

그림 5. app.js 내 실행코드 삽입
Fig. 5 Insert execution code in app.js

변조된 웹사이트를 통해 서버에 메시지를 전송하고, 서버에서 메시지를 처리하는 과정에서 실행코드가 실행되어 원격 쉘을 공격자에게 제공하게 된다.

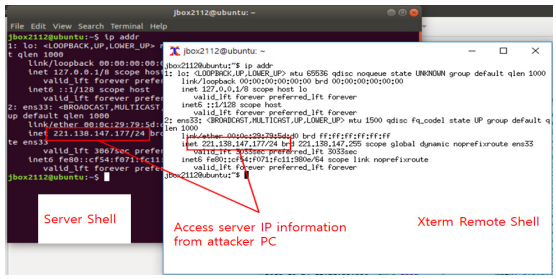

그림 6. 2차 공격 예시[IP 조회]
Fig. 6 Example of a secondary attack

3.4 공격 동작 원리

공격자가 서버의 원격 쉘을 얻는 원리를 요약설명하면 그림 7과 같은 절차로 진행된다.

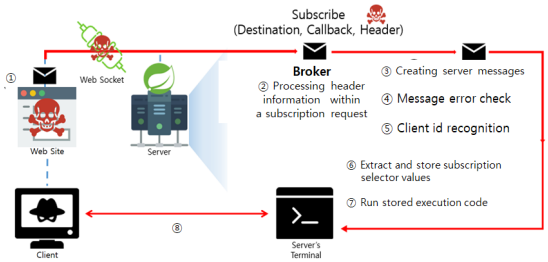

그림 7. 서버 내 실행코드 처리 프로세스
Fig. 7 In-server execution code processing process

① 변조된 클라이언트와 서버가 웹 소켓 연결을 구성하여 통신한다.
② 웹 소켓 연결 시 클라이언트와 서버 사이에 통신 중개자(Broker)가 구성되며 Web Socket 연결이 성공적으로 구성이 된다면 서버에서 클라이언트가 전송할 메시지를 처리할 함수의 경로를 설정하는 행위(subscribe)를 Broker에 저장한다. 즉, 클라이언트에 삽입한 실행코드가 Subscribe 행위를 통해 Header에 내포되어 Broker로 전달 및 저장이 된다.
③ 클라이언트에서 메시지를 서버에게 전송하면 Broker에서 평문 메시지(from client)와 Subscribe 정보(Destination, Callback, Header)를 패킹하여 응답 메시지(from Server)를 생성하는 함수로 전달한다.
④ 생성된 응답 메시지(from Server)에 대한 오류 검사를 수행한다.
⑤ Broker에서 메시지에 함께 패킹하였던 Session 정보를 통해 클라이언트 ID를 인식하여 응답 메시지(from server)를 전달할 클라이언트를 식별한다.
⑥ 식별 후 응답 메시지(from server)를 클라이언트로 전송해야하지만 Header에 삽입했던 Selector 변수가 Null이 아니라면 값을 추출하여 Expression 객체에 저장한다.
⑦ Expression 객체에 저장된 값(실행코드)을 java application에서 처리하면 java application 실행구문으로 인해 실행코드가 실행된다.
⑧ 서버의 원격 쉘을 공격자에게 제공하는 명령어 실행코드로 인해 공격자는 서버의 쉘을 원격으로 획득하여 원격 통신을 통해 2차 공격이 가능하다.

3.5 대응 방안

스프링 프레임워크 4.3~4.3.15 / 5.0~5.0.4 구간 버전에서 웹 소켓통신을 한다면 위와 같이 치명적인 공격을 받을 위험성이 존재한다. 이에 대한 대응 방안으로 네트워크의 침입여부를 탐지하는 탐지 규칙 Snort Rule을 그림 8과 같이 제시한다.

이때 각 버전에 대한 패치(patch)는 Drupal 7.x는 7.58, 8.3.x는 8.3.9, 8.4x는 8.4.6, 8.5x는 8.5.1로 업데이트를 권고한다. 이렇게 조치하면 스프링 프레임워크 보안 특화 모듈 스프링 시큐리티에서 메시지에 대한 인증 및 권한을 부여하여 미 인증 클라이언트에게 취약점이 노출되는 것을 방지할 수 있다.

```
>
> alert http any any -> $HOME_NET any (msg: "[PT OPEN] Drupalgeddon2 <8.3.9
> <8.4.6 <8.5.1 RCE through registration form (CVE-2018-7600)"; flow:
> established, to_server; content: "/user/register"; http_uri; content:
> "POST"; http_method; content: "drupal"; http_client_body; pcre:
> "/(%23|#)(access_callback|pre_render|post_render|lazy_builder)/Pi";
> reference: cve, 2018-7600; reference: url, research.checkpoint.com/
> uncovering-drupalgeddon-2; classtype: attempted-admin; sid: 10002808;
> rev: 2; )
>
```

그림 8. 대응을 위한 Snort Rule
Fig. 8 Snort Rule for response

Ⅳ. 결 론

국내 기업 70% 이상이 스프링 프레임워크 서버를 운영하며 Model, View Controller 3가지 개념으로 서버와 클라이언트 간의 효율적 서비스를 제공하여 소스코드의 유지보수와 재사용에 편리한 장점이 있다.

하지만 시큐어 코딩, 보안 장비만으로 취약점을 완벽히 차단한다는 것은 불가능하다. 취약점은 하나의 요소라도 취약하다면 공격에 노출될 수 있기 때문에 서버 설정, 보안장비 설정, 프레임워크 버전 등 다양한 요소에 대하여 주의를 기울여야 한다. CVE 취약점의 경우 크고 작은 보안 이슈로 인해 공식적으로 정리하여 발표한 취약점이며 보안조치 방안까지 안내하고 있지만 아직도 과거의 CVE 취약점에 피해를 입는 서버는 존재한다.

본 논문은 공격 시나리오를 통해 스프링 프레임워크 서버에서 발생하는 CVE-2018-1270 취약점에 대해 분석추적을 실험을 통해 살펴보았다. 아울러 이에 대한 대응방안으로 Snort Rule을 제시하였으며 실행에서는 스프링 프레임워크 버전을 4.3.16 / 5.0.5 이상으로 업그레이드하기를 권고하였다.

앞으로 전자정부 기본 프레임워크가 갖는 중대성에 따라 이에 대한 지속적인 연구가 필요하며, 본 제안이 향후 보안 프로그램의 성능 향상 및 새로운 취약점 보안 구축에 활용되기를 기대한다.

Reference

[1] B. Jung and C. Choi, "Spring Server Remote Code Execution Vulnerability (CVE-2018-1270)," *Proc. of the 2018 Fall Conf. on The Korea Institute of Communication and Information Sciences*, Seoul, Korea, Jan. 2016, pp. 665-667.

[2] S. Lee, T. Ahn, and J. Jung, "A Study of Broadleaf Open Source Commerce Swift based on Spring Framework," *Proc. of the 2014 Summer Conf. on The Korea Society of Computer and Information*, Jeju, Korea, July 2014, pp. 213 - 216.

[3] H. Choi and D. Son, "Spring Framework Responsive Web AWS," *Proc. of the 2017 Summer Conf. on The Korea Society of Computer and Information*, Seoul, Korea, July 2017, pp. 149-150.

[4] J. Ye, J. Kim, and M. Chung, "Spring Framework based Integrated Support System for Assets and Insolvency Prediction in the Cloud Computing," *Proc. of the 2014 Fall Conf. on Korea Information Processing Society*, Seoul, Korea, Jan. 2014, pp. 699 - 702.

[5] K. Kwon and J. Byun, "Implementation and evaluation of enabling skill for POJO programing Based on Spring Framework," *Proc. of the 2013 Fall Conf. on Korea Information Processing Society*, Jeju, Korea, Jan. 2013, pp. 1027-1029.

[6] W. Choi, "Countermeasures research on HTTP trapping attack in the spring framework based web environment," Master's Thesis, *Sungkyunkwan University*, 2014.

[7] W. Seo and M. Jeon, "A Study on Security Hole Attack According to the Establishment of Policies to Limit Particular IP Area," *J. of the Korea Institute of Electronic Communication*

Sciences, vol. 5 no. 6, 2010, pp. 625-630.
[8] S. Kim, S. Lim, and D. Kim, "Regulatory Requirements Analysis for Development of Nuclear Power Plants Cyber Security Vulnerability Inspection Tool," *J. of the Korea Institute of Electronic Communication Sciences,* vol. 12 no. 5, 2017, pp. 725-270.
[9] H. Cho, J. Kim, and B. Noh, "Analysis for Security Vulnerabilities on DSTM Tunneling," *J. of the Korea Institute of Electronic Communication Sciences,* vol. 2 no. 4, 2007, pp. 215-221.
[10] J. Jeon, "A Study on The Vulnerabilities and Problems of Security Program," *J. of Information and Security, Korea Information Assurance Society*, vol. 12 no. 6. 2012, pp. 77-84.

저자 소개

정병문(Byung-Moon Jeong)

2014년 경동대학교 정보보안학과 입학
2018년 경동대학교 정보보안학과 4학년 휴학 중(육군 정보보호병 입대 예정)
2018 한국인터넷진흥원주관, S/W개발보안 경진대회, 한국정보보안학회장상 수상
※ 관심분야 : 모바일 보안, 웹서버보안

장재열(Jae-Yeol Jang)

1984년 동국대학교 전자계산학과 졸업(공학사)
1995년 경희대학교 교육대학원 전자계산교육전공 졸업(석사)
2001년 관동대학교 대학원 전자계산공학과 졸업 (공학박사)
1995년~현재 경동대학교 정보보안학과 교수
※ 관심분야: 웹서버보안, 컴퓨터교육

최철재(Chul-Jae Choi)

1983년 광운대학교 전자계산학과 졸업(이학사)
1987년 한양대학교 산업대학원 전자계산학전공 졸업(공학석사)
2000년 강원대학교 컴퓨터과학과 졸업(이학박사)
1988년~현재 경동대학교 정보보안학과 교수
2015년~2016년 경동대학교 평생교육원장
※ 관심분야 : 데이터처리, 영상처리, 웹보안