취약한 HTTPS 인증 시스템에 대한 SSL 프록시 중간자 공격

지우중, 이경문, 이병천

중부대학교 정보보호학과

SSL Proxy MITM Attacks Against Insecure HTTPS Authentication System

Woojoong Ji, Kyungmoon Lee, Byoungcheon Lee

Department of Information Security, Joongbu University.

요 약

모바일 기술이 발전하고 스마트폰이 확산되면서 기존의 다양한 인터넷 서비스들이 스마트폰 어플리케이션 형식으로 제공되고 있으며 스마트폰의 장점을 이용한 새로운 서비스들도 등장하고 있다. 하지만 인터넷 서비스 개발자 및 스마트폰 어플리케이션 개발자들의 보안개발에 대한 인식은 아직 취약한 부분이 많이 있는 것으로 보인다. 로그인 등과 같이 중요한 통신내용의 보안을 제공하기 위해서는 SSL/TLS 프로토콜을 이용한 https 통신을 적용하는 것이 일반적인데 클라이언트가 서버의 SSL 인증서를 검증하는 과정을 생략하는 등 불완전하게 구현하는 경우 공격자가 SSL 프록시를 이용하는 중간자 공격을 통해 정보를 탈취하거나 바꿔치기 하는 등의 공격이 가능하다.

본 논문에서는 SSL 인증서에 대한 검증을 생략한 스마트폰 어플리케이션들의 취약점을 조사하였다. SSL 프록시를 이용한 중간자 공격으로 통신을 도청하여 사용자의 ID/Password를 알아내는 것이 가능하고, 토큰을 이용하는 인증시스템에서는 토큰을 도청하고 타인의 것으로 바꿔치기 함으로써 타인의 이름으로 로그인할 수 있게 된다. 대표적으로 요즘 널리 확산되고 있는 스마트폰을 이용한 출석관리 어플리케이션을 분석하여 이런 취약성을 가지고 있다는 것을 보인다.

I. 서론

최근 모바일 기술이 발전하고 스마트폰이 확산되면서 여러 가지 업무를 인터넷뿐만 아니라 스마트폰으로도 처리할 수 있도록 발전하고 있다. 또한 성능이 높은 모바일 컴퓨터로서의 스마트폰의 장점을 이용한 새로운 서비스들도 등장하고 있다.

예를 들면 우리나라 대학가에서는 스마트폰을 이용한 출석관리 어플리케이션의 도입이 확산되고 있다. 중부대학교에서는 X사에서 개발한 스마트체크[1]라는 출석관리 어플리케이션을 도입하였는데 이것은 학생의 스마트폰과 교수의 스마트폰이 블루투스 통신을 통하여 동일한 공간에 있다는 것을 체크함으로써 출석을 관리하는데 특별한 하드웨어 장치의 도입이 필요없이 스마트폰만으로 출석관리가 가능하다는 장점이 있어서 국내 수십개의 대학에 도입되어 사용되고 있다.

http 방식의 인터넷 통신은 쉽게 도청될 수 있으므로 로그인 등의 중요 정보가 전달되는 채널에서는 SSL/TLS[2] 프로토콜을 이용한 https 보안통신을 적용하는 것이 일반적이다. https 프로토콜을 적용하면 클라이언트와 서버가 인증서를 검증하여 상호 신분을 확인할 수 있고 세션키를 유도하여 암호화 통신을 함으로써 통신정보의 노출을 막을 수 있게 된다. 그런데 시스템 구현시 이것이 불완전하게 적용되는 경우 취약성을 가질 수 있다.

X사의 스마트체크 어플리케이션에서도 통신보안을 위해 https 통신을 이용하지만 스마트폰 앱이 서버의 인증서를 검증하지 않는 등 취약하게 구현되어 있어서 SSL 프록시 서버를 이용한 중간자 공격으로 쉽게 공격할 수 있음을 발견하였다.

본 논문에서는 SSL 프록시를 이용한 중간자 공격 기법을 설명하고 X사의 스마트체크 어플리케이션의

취약성을 분석한다. 아울러 SSL 통신을 사용하지만 SSL 프록시를 이용하여 ID/Password가 쉽게 도청되는 사례들을 제시한다.

II. SSL 프록시를 이용한 중간자 공격

SSL/TLS 프로토콜을 이용한 https 통신을 적용하면 기본적으로 서버와 클라이언트 사이에 비밀 세션키가 생성되고 이들 사이에 전송되는 데이터가 세션키로 암호화되어 전송되기 때문에 공격자가 중간에 패킷을 가로챈다고 해도 원하는 정보를 얻을 수 없다. 그러므로 서비스의 운영자와 사용자 모두 통신정보의 보안에 대해 안심할 수 있게 된다.

2.1 SSL 중간자 공격

그런데 SSL을 우회할 수 있는 중간자 공격[3]에 대해 여러 가지 시도가 있어왔다. sslstrip[4]은 공격자가 서버와 클라이언트 사이의 중간에 프록시 서버를 운영하면서 서버와 https 통신을 맺고 클라이언트와는 http 통신을 맺어 통신을 중개하면서 사용자의 로그인 정보를 서버에 그대로 전달하도록 하는 공격기법으로 사용자가 안심하고 입력하는 로그인 정보를 탈취할 수 있는 방법이다. 그런데 이 툴은 클라이언트가 SSL 통신을 하도록 되어있는 경우에는 중간자 공격을 위해 적용하기 어렵다.

SnoopSpy[5]는 sslstrip이 아닌 ssl-sniff 기능과 함께 프록시 서버가 자신도 가짜 인증서를 만들어 클라이언트와 SSL 통신을 할 수 있도록 하고 있어서 클라이언트와 서버간의 SSL 통신 전체를 중개할 수 있다. 또한 통신정보 바꿔치기 등 많은 기능을 내장하고 있다. 여기에서는 SnoopSpy 툴을 이용하여 SSL 프록시 공격을 수행하는 사례를 보인다.

2.2 공격 기법

본 논문에서 사용되는 공격기법과 용어들에 대해 짧게나마 설명하고자 한다.

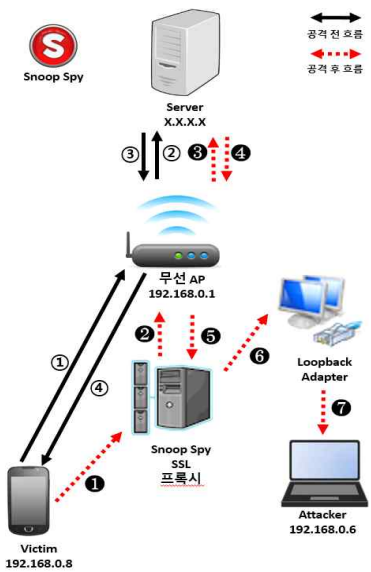
가장 먼저 이루어져야 할 공격은 통신 내용을 도청할 수 있는 스니핑인데 이것은 네트워크를 통해 오고가는 패킷들을 제 3자가 동의 없이 패킷을 가로채서 보는 것을 말한다. 스니핑을 위해서는 통신 패킷이 공격자를 거쳐가도록 해야 하는데 본 논문에서는 ARP 스푸핑을 이용한다.

스푸핑이란, 악의적인 공격자가 자신의 신분을 속이는 것을 말한다. 스푸핑의 기법에는 ARP 스푸핑, IP 스푸핑, 쿠키 스푸핑, DNS 스푸핑 등의 기법들이 존재한다. 공격자는 ARP 스푸핑을 통해 공격 대상의 ARP 테이블을 변조시켜 그 컴퓨터의 모든 네트워크 패킷은 공격자의 컴퓨터를 거쳐가게 되고 공격자는 해당 패킷에 대해서 정상적으로 처리해 줌으로써 공격 대상 사용자는 아무런 의심없이 네트워크 통신을 한다.

2.3 공격 시나리오

공격자가 SnoopSpy를 이용하여 운영하는 SSL 프록시는 서버와 클라이언트 사이의 SSL 통신을 중개하는 역할을 한다. [그림 1]에 전체적인 공격 시나리오를 도시하였다.

SSL 프록시가 없는 경우 사용자의 스마트폰은 무선AP를 통해 서버와 SSL 통신을 이용하여 서비스를 이용하게 된다. 공격자가 SSL 프록시를 동작시키고 사용자의 스마트폰 IP주소와 무선AP의 게이트웨이 주소를 입력하여 ARP스푸핑을 실행하면 사용자 스마트폰의 모든 통신패킷이 SSL 프록시 서버를 경유하게 되고 공격자는 루프백 어댑터를 이용하여 사용자의 통신패킷을 볼 수 있게 된다. SSL 프록시는 SSL 통신을 중개하기 위하여 가짜 인증서를 만들어 SSL 패킷을 사용자 스마트폰에 보내고 서버에서는 클라이언트를 가장하여 SSL 통신을 한다. 만일 사용자 스마트폰이 서버의 SSL 인증서의 유효성을 검증하고 신분을 확인한다면 이런 공격을 찾아내어 에러메시지를 내고 중단시킬 수 있겠지만 사용자 스마트폰이 SSL 프로토콜을 사용하기만 하고 서버의 인증서를 검증하지 않는다면 이런 중간자 공격은 훌륭하게 동작하게 된다.



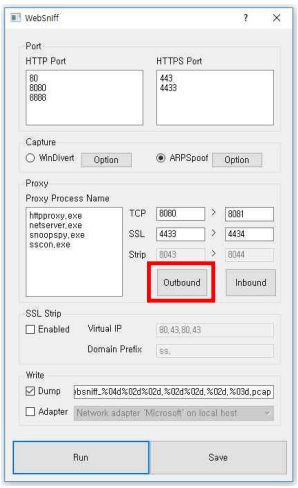
[그림 1] SnoopSpy SSL 프록시를 이용한 공격

III. HTTPS 인증시스템의 공격 사례

3.1 ID/Password의 노출 사례

일반적으로 ID/Password의 로그인 정보를 https 프로토콜로 보내면 안전하다고 생각하는데 만일 클라이언트에서 서버의 SSL 인증서를 검증하지 않는다면 위와 같은 SSL 프록시를 이용한 중간자 공격으로 통신 내용을 도청할 수 있다. 다음의 [그림 2]는 SnoopSpy의 WebSniff 기능을 이용하여 Outbound 통신 데이터를 스니핑 하는 사례를 보인다.

업무용 메신저 어플리케이션으로 많은 사람들이 이용하고 있는 J[6]라는 어플리케이션에서 이러한 취약점이 발견되었다. 이 어플리케이션의 모바일 앱을 이용하는 경우 SSL 프록시를 이용하여 도청하면 [그림 3]에 보인 바와 같이 ID/Password가 평문으로 보이는 것을 알 수 있다. 이 정보를 이용하면 J 서비스에 타인의 신분으로 로그인할 수 있게 된다.



[그림 2] SnoopSpy의 WebSniff

```
POST /inner-api/token HTTP/1.1
Content-Type : application/json
Accept: application/vnd.tosslab.jandi-v3+json
Authorization:
User-Agent: JandiApp(android; 19; LG-F240S; 2.3.1.6;)
Content-Type: application/json; charset=UTF-8
Content-Length: 232
Host: i2.jandi.com
Connection: Keep-Alive
Accept-Encoding-SS: gzip

{"uuid":"df0b907128afd322","app_version":"2.3.1.6","model":"LG-
F240S","name":"LGFeeefhd_skt_kr","grant_type":"password","platform":"android","platform_version":"19"
,"tokens":[],"username":"wldnwnd98@naver.com","password":          }HTTP/1.1 200 OK
```

[그림 3] SSL 프록시로 본 J 서비스 로그인 정보

3.2 인증토큰의 노출 및 변조 사례

모바일 어플리케이션의 경우 ID/Password의 입력이 매우 불편하므로 인증토큰[7]을 만들어 자동 로그인 기능을 사용하는 경우가 많다. 인증토큰을 이용하는 서비스에서는 처음 로그인 할 때는 ID/Password를 요구하지만 한번 로그인을 하면 서버에서 인증토큰을 만들어 사용자에게 제공하고 이것을 사용자 컴퓨터에 저장하게 된다. 사용자가 다음 서비스 접속시에는 인증토큰을 첨부하게 되며 서버는 인증토큰을 검증하여 로그인을 허용하게 된다. 인증토큰에는 해당 사용자에 대한 정보가 내장되어 있어서 인증토큰만 있으면 사용자 신분을 확인할 수 있게 된다.

인증토큰은 이렇게 로그인에 사용되므로 매우 중요한 정보라고 볼 수 있으며 안전하게 전송되어야 한다. 이를 위해 https 통신을 이용하게 되는데 여기에서도 클라이언트가 서버의 인증서를 검증하지 않으면 SSL 프록시를 이용한 중간자 공격을 이용하여 인증토큰을 도청해내고 변조하여 사용할 수 있다. 다음의 [그림 4]는 중부대학교 출석관리 시스템에서 도청된 key라는 이름의 인증토큰 정보이다. 더구나 이 key 정보는 일정기간 변경되지 않고 고정되어 사용되는 정보이다.

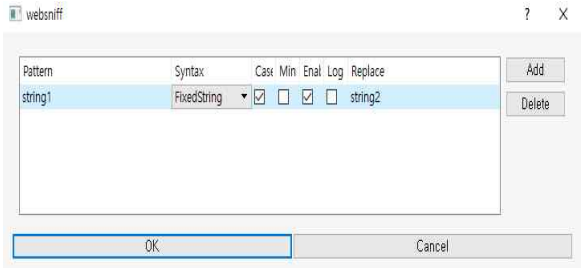
```
POST /attend/rb_login.php HTTP/1.1
Content-Length: 708
Content-Type: application/x-www-form-urlencoded
Host: attend.joongbu.ac.kr
Connection: Keep-Alive
User-Agent: android-async-http/1.4.1 (http://loopj.com/android-async-http)
Cookie: JSESSIONID=DF601E174F9E2CF3B18374A10A8267DA
Cookie2: $Version=1
Accept-Encoding-SS: gzip

key=ED06609279F2C20B5399C396B0246F477A7484027D4F21E20B4787FD4F6B12BD8BA78E6A1E60A0E378634236F7F39F928
7ABC36BB3DCFBB46D2A08AEE2F06E8295AF95E367869C01BFCFB6B95FD9D994E7EEBA6D34DBBDBCFA94CAF5DBBF577460E053
934E34BCD9B9E81E8C505D4CB9061300F14893CB4885D2A288BE76A79BDB3D041A6F101946F2A3EB40A8D98FBDB1455F7C25D4
```

[그림 4] 중부대학교 출석관리 시스템 로그인

이러한 key 정보를 도청하고 변조하여 사용하면 다른 사람의 아이디로 쉽게 로그인할 수 있게 된다. 예를 들어 학생이 교수님의 인증토큰을 도청하고 변조하여 사용함으로써 교수님의 신분으로 로그인할 수 있다. 이를 테스트하기 위해 SnoopSpy의 Data-Change 기능을 이용하였다.

[그림 5]는 SnoopSpy SSL 프록시에서 학생의 key 값을 미리 획득한 교수님의 key 값으로 변조하여 요청하는 것을 보여준다.



[그림 5] SnoopSpy의 Data-Change 하는 기능

이렇게 인증토큰으로 사용되는 key 값을 변조하여 요청하면 스마트폰은 교수님의 이름으로 로그인되는 것을 확인할 수 있었다.

3.3 취약점 개선 방안

여기에서 확인된 취약점은 사용자 클라이언트가 서버의 SSL 인증서를 검증하고 신분을 확인하지 않아서 발생한 문제이므로 서버의 SSL인증서를 확인하도록 개선하면 해결할 수 있는 문제이다.

하지만 ID/Password를 이용한 로그인, 토큰을 이용한 로그인은 기본적으로 고정된 정보를 이용하고 있어서 재전송 공격에 취약하다. 전자서명을 이용한 인증을 적용하면 매 로그인시마다 달라지는 챌린지 메시지에 사용자의 개인키를 이용한 전자서명을 요구하게 되므로 더 안전한 인증시스템을 운영할 수 있다.

IV. 결론

SSL 보안통신을 이용하더라도 사용자 클라이언트가 서버의 인증서를 검증하지 않으면 SSL 프록시를 이용한 중간자 공격에 취약하다는 것을 보였으며 우리가 일상적으로 사용하는 많은 서비스, 어플리케이션들이 이런 취약한 방법으로 운영되고 있다는 것을 확인하였다. 또 중부대학교에서 사용되고 있는 스마트 출석관리 시스템뿐만 아니라 같은 회사의 제품을 도입하여 사용하는 다른 대학교의 스마트 출석관리 시스템에서도 동일한 취약점이 발견되었다. 이 회사뿐만 아니라 다른 회사의 출석관리 시스템에서도 동일한 취약점이 발견된 바가 있고, 아예 SSL 통신을 하지 않는, 즉, ID/Password가 http 통신으로 평문으로 전송되는 경우도 있었다. 서비스 개발자, 스마트폰 어플리케이션 개발자들은 이러한 취약성들에 대해 인식하고 보안 프로토콜의 원리에 대한 이해를 바탕으로 좀 더 신뢰성 있게 구현하려는 노력이 필요하다고 생각된다.

[참고문헌]

[1] 중부대학교 스마트 출석 체크 시스템, https://play.google.com/store/apps/details?id=com.jbu.jbu_abookn

[2] The Transport Layer Security (TLS) Protocol Version 1.2, https://tools.ietf.org/html/rfc5246

[3] 임차성, 이우기, 조태창, "SSL MITM 프록시 공격에 대한 효과적 방어방법", 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 16(6), 693-697. 2010.6.[3]

[4] sslstrip, http://sectools.org/tool/sslstrip/

[5] SnoopSpy, http://www.snoopspy.com/

[6] Jandi, https://www.jandi.com/

[7] JWT, https://jwt.io/