1. 기사 내용

제목	웹개발자, '취약한 세션 관리' 포인트
기사 날짜	2006년 3월 31일
스크랩 담당자	손우규
요약	• 취약한 세션 관리 (Cookie Injection) 쿠키(Cookie)는 사용자 정보를 유지할 수 없는 HTTP(Hyper Text Transfer Protocol)의 단점을 해결할 수 있는 방법의 하나로서 각 서버는 쿠키를 사용하여 브라우저가 갖고 있는 정보를 참조할 수 있다. 이와 같이 클라이언트에서 동작되는 쿠키는 암호화 등의 문제를 비롯하여 그 구조상 클라이언트 측에서의 조작으로 인한 다양한 문제점을 가지고 있어, 많은 웹 프로그래밍 언어들에는 서버에 클라이언트의 정보를 저장하는 세션(Session)을 지원하고 있다. 적절히 보호되지 않은 쿠키를 사용하면 Cookie Injection등과 같은 쿠키 값 변조를 통하여 다른 사용자로의 위장 및 권한상승 등의 문제가 생길 수 있다. 또한 쿠키 및 세션은 Cookie Sniffing 및 이후에 이야기될 악성스크립트실행(XSS)를 통한 Cookie Hijacking 등과 같은 쿠키 값 복사를 통해 현재 활성화된 사용자의 권한복제 위험성이 존재한다.
출처	http://www.boannews.com/media/view.asp?idx=1837

2. 용어

용어	설명
세션(Session)	두개 또는 그 이상의 의사소통하는 장치들(devices) 또는 컴퓨터 그리고 유저 간에 대화, 회화 또는 회의와 같은 반영구적인 쌍방향 정보 교환
쿠키(Cookie)	하이퍼 텍스트의 기록서(HTTP)의 일종으로서 인터넷 사용자가 어떠한 웹사이트를 방문할 경우 그 사이트가 사용하고 있는 서버를 통해 인터넷 사용자의 컴퓨터에 설치되는 작은 기록 정보 파일

3. 관련 기술(세션 하이재킹)

3.1 개요

세션 하이재킹이란 "세션 가로채기"를 의미하는 것으로 세션이란 사용자와 컴퓨터, 또는 두 대의 컴퓨터 간 활성화된 상태를 의미한다.

그 중 TCP 세션 하이재킹은 서버와 클라이언트가 통신을 수행할 때 TCP 프로토콜의 시퀀스 넘버를 제어하는데 문제점이 있다는 것을 알고 이를 통해 공격을 수행하는 방법이다.

원격 세션 하이재킹 공격 역시 기본적인 알고리즘은 로컬 세션 하이재킹 공격과 비슷한, 한가지 차이점이 있다면 로컬 세션 하이재킹 공격은 공격 대상을 탐지할 수 있고 서버와 클라이언트가 통신을 수행할 시 시퀀스 넘버를 알아낼 수 있는데 반해, 원격 세션 하이재킹 공격은 그것이 불가능하다는데 있다.

따라서 원격 세션 하이재킹 공격은 서버와 클라이언트간의 시퀀스 넘버를 무작위로 대입해 보아야 하며 성공 확률이 매우 낮다. 이는 시퀀스 넘버는 32비트의 숫자로 이루어져 있기 때문이다.

3.2 공격 형태

3.2.1 Session ID 공격

Session ID에 대한 공격은 크게 2 단계로 이루어진다. 1단계는 Session ID 취득 단계이고 2단계는 취득한 Session ID 공략 단계이다.

공격자는 Session ID를 취득하기 위하여 웹 서버와 웹 클라이언트의 트래픽을 직접적으로 스니핑하거나, 웹 서버 상에 공격 코드를 삽입하여 두고 사용자가 클릭할 때 null, Session ID 값을 전송받을 수 있도록 한다. 웹 서버와 웹 클라이언트 사이의 트래픽을 직접적으로 스니핑하는 방법은 일반적인 네트워크 트래픽 스니핑 기법을 통해 가능하다.

3.2.2 네트워크 트래픽 스니핑을 통한 HTTP 요청 헤더 안의 쿠키값

..comAccept-Language: koContent-Type: application/x-www-form-urlencodedAccept-Encoding: gzip, deflateUser-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)Host: ibn.kbstar.comContent-Length: 297Connection: Keep-AliveCache-Control: no-cachenull: Session=QJ48621878865

웹 서버 상의 HTML 코드 삽입이 가능한 페이지는 주로 사용자가 글을 게시할 수 있는 게시판이나 자료실 등에 존재한다. 정상적인 글을 게재하는 대신 공격자는 HTML 코드 및 스크립트를 심어 넣는다. 일반 사용자는 해당 게시물을 열람하게 될 때 자신도 모르는 사이 null, Session ID 정보가 제 3 의 공격자 서버나 이메일로 전송되게 된다. (이러한 공격 기법을 Cross-Site Scripting이라고 부른다.)

3.2.2 쿠키값 탈취를 위한 HTML 삽입

<form name=f method=POST action="http//host/hello.php"><input type=hidden name="name"value="<script>alert(doc-ument.cook-ie)</script>"></form><script>f.submit()</script>

결과적으로 공격자는 취득한 타인의 Session ID 값을 웹 서버에 요청함으로써 HTTP Session Hijacking 공격을 시도할 수 있다. 물론 이 공격이 성공하려면 Session ID 값이 유효해야 하므로, 사용자가 로그온한 상태에서 공격이 이루어져야 하거나 Session ID 값의 유지 시간이 긴 경우라는 제한 사항이 필요하다.

그러나, 기본적으로 잘못 설계된 세션 관리 기법을 사용하고 있는 웹 서버는 이러한 Hijacking 공격에 취약할 수 밖에 없다. 굳이 타인의 Session ID 값을 직, 간접적으로 취득하지 않고도 무작위 추측 대입(Brute-force Guessing)함으로써 공격이 가능하다. 공격자는 정상적인 로그인 과정시 분석한 자신의 null 값, 웹 브라우저의 주소창의 URL, 웹 페이지 폼 소스 hidden field 내에 노출된 Session ID 값 자체를 분석한다.

Session ID 생성 방식의 취약점을 파악한 후, 공격자의 컴퓨터에서 로컬 프록시 툴이나 웹 브라우저 창 URL 주소창에서 직접 Sessioion ID 값 단일 대입을 시도한다. 더 나아가 자동적인 Session ID 대입 스크립트를 작성하여 공격할 수도 있다.

3.3 공격 원인

TCP 세션 하이재킹은 서버와 클라이언트가 통신할 때 TCP 시퀀스 넘버를 제어하는 데 문제점이 있음을 알고 공격하는 것이다. TCP 세션 하이재킹은 Non-Blind Attack과 Blind Attack이 있다. Non-Blind Attack과 Blind Attack의 차이점은 Non-Blind Attack은 공격 대상을 탐지할 수 있으며 서버와 클라이언트가 통신할 때 시퀀스 넘버를 알아낼 수 있다. Blind Attack은 서버와 클라이언트 간의 시퀀스 넘버를 생성할 경우 성공 확률은 0%에 가깝다

TCP 세션 하이재킹에서는 사람이 자리에서 일어나서 다른 곳에 가기를 기다릴 필요가 없다. 공격 대상이 공격자가 원하는 접속만 생성하면 공격자는 네트워크 공격을 통해 세션을 빼앗을 수 있다. TCP 세션 하이재킹은 TCP가 가지는 고유한 취약점을 이용해 정상적인 접속을 빼앗는 방법이다. 시퀀스 넘버가 잘못되면 이를 바로 잡기 위한 작업을 하는데, TCP 세션 하이재킹은 각각 서버와 클라이언트에 잘못된 시퀀스 넘버를 위조해서 연결된 세션에 잠시 혼란을 준 뒤 자신이 끼어 들어가는 방식을 사용한다.

3.4 보완 방법(해결방안)

3.4.1 생성범위 지정

Session ID 생성 범위값을 사용자 수에 대비하여 충분히 큰 값으로 설정한다.

3.4.2 랜덤 생성

Session ID는 가능한 한 추측 불가능(random)하게 생성한다. 무작위 추측 대입 공격을 할 때 공격자는 그만큼 더 많은 시간을 투입하여야 하며 현재 연결되어 활성화 되어 있는 유효한 Session ID 값을 찾는 확률은 낮아진다.

3.4.3 주기적 생성

Session Timeout 기능과 Session ID 재생성 기능을 사용한다. 일정 시간 동안 활동이 없는 사용자는 새로운 Session ID로 재로그인 하도록 하고, 사용자 로그아웃 시에는 Session ID 값을 폐기한다. 장시간 접속이 필요한 어플리케이션의 경우에는 일정 주기마다 Session ID값을 자동으로 재생성하여 세션을 유지하도록 한다.

3.4.4 사용자 잠금 기능

무작위 추측 대입(Brute-force Guessing)에 대비하여 일정 회수 이상의 인증 실패 시에는 사용자 잠금 기능을 구현한다.

3.4.5 사용자 인증 제한

로그인 이후에도 중요한 서비스 이용 시에는 사용자 인증을 통하여 인가된 사용자만이 해당 서비스를 이용할 수 있도록 통제한다.

3.4.6 변수 암호화

null 내용 안의 Session ID와 기타 변수 값 자체를 암호화한다.

3.4.7 SSL 구현

웹 서비스 자체의 중요성에 따라 null이 전달되는 방식을 SSL로 구현함으로써 스니핑 공격에 대응할 수도 있다.

3.4.8 주기적 필터링

웹 서비스 상에 공격자가 HTML 공격 코드 삽입이 가능한 페이지가 있는지 점검한다. 직접적인 공격 코드 삽입을 차단할 수 있도록 특수 문자 및 스크립트 코드를 필터링하여야 한다.

4. 참고 자료

- https://ko.wikipedia.org/wiki/HTTP_%EC%BF%A0%ED%82%A4
- http://ingorae.tistory.com/362