WWW 에서 메시지 축약을 사용하는 인증기법의 변형된 구현

유정각 O, 이건희, 손태식, 채송화, 김동규
아주대학교 정보통신공학과
{ kagi, icechoco, tsshon, portula, dkkim }@madang.ajou.ac.kr

Modified Implementation of Authentication using Message Digest in WWW

Jeong-Gak Lyu O, Gun-Hee Lee, Tae-Shik Shon, Song-Hwa Chae, Dong-Kyoo Kim
Dept. of Information Communication Engineering, GSIC, AJOU

요 약

WWW 의 발전과 함께 발생한 역기능으로 사용자의 계정 도용을 통해 사용자의 프라이버시를 침해하거나 도용한 계정의 권한을 남용하는 등의 문제가 발생하고 있다. HTTP 는 기초적인 사용자 인증 기법을 지원하고 있으나 취약성이 존재하고, 이를 위해 소개된 메시지 축약을 응용한 새로운 인증기법 역시 구현의 불편함이나 브라우저의 지원 부족으로 거의 사용되지 않고 있다. 이러한 인증 과정을 단순화하고 대부분 웹 브라우저가 지원하는 애플릿을 통해 구현함으로써 적은 수정으로 현재 서비스 중인 사이트에 적용할 수 있다. 이러한 안전한 인증 과정을 통해 사용자 도용 문제를 줄일 수 있다.

1. 서론

WWW(World Wide Web)의 발전은 네트워크를 통해 정보의 교환이나 검색 등의 기존의 역할을 뛰어넘어 웹을 통한 메일 송·수신, 인트라넷(Intranet) 환경에서 회사나 기관의 업무를 웹으로 옮겨오는 등 그 기능이 다양해 졌다. 이에 따라 다양해진 서비스를 보완하기 위한 인증이나 비밀 통신 등이 요구되었고 현재 여러 방법들이 적용되어 사용되고 있다.

특히 사용자 인증은 HTTP(Hyper Text Transfer Protocol) 안에 기본적인 인증 기법을 포함되어 있을 정도로 필수적인 보안 요소로 볼 수 있다.[1] 그러나 현재 대부분의 웹 사이트에서 사용되는 사용자 인증 기법은 계정과 비밀번호가 패킷(packet) 안에 일반 텍스트로 포함되어 있어 스니핑(sniffing)에 의한 취약성을 가지고 있다.

본 논문에서는 현재 사용되는 사용자 인증 기법의 취약성을 살펴보고 이를 보완할 수 있는 메시지 축약(message digest)을 사용하는 인증 기법을 간소화하여 설계 구현한 결과를 살펴보도록 하겠다.

2. 현재 사용되는 인증 기법

웹 메일 서비스나 커뮤니티 서비스를 제공하고 있는 국내의 대형 포털 사이트들에서는 한번의 사용자 인증으로 모든 서비스를 제공하기 때문에 계정의 도용에서 오는 피해는 상당히 크며 최근 그러한 사건이 많이 발생하고 있다. 그러나 웹사이트의 감사(audit) 기록 등의 부족으로 도용한 범인에 대한 추적은 용이하지 않다. 이러한 포털 사이트를 중심으로 사용되고 있는 인증 기법은 다음과 같다.

2.1 계정, 비밀번호, 쿠키를 사용하는 사용자 인증

계정과 비밀번호, 쿠키를 사용하는 사용자 인증 방식은 사용자가 해당 사이트에 회원으로 가입 시 결정되는 계정과 비밀번호를 통해 이루어진다. 전체 인증 과정은 다음과 같다.

① 회원 가입 시 사용자 계정과 비밀번호를 포함한 사용자 정보를 웹 서버에 전달한다.

② 웹 서버는 사용자 정보를 별도의 데이터베이스에 저장한다.

③ 사용자 인증이 필요한 경우 사용자는 HTML 문서의 FORM 태그를 통해 생성된 양식에 계정과 비밀번호를 입력한다.

④ 입력된 계정과 비밀번호는 HTTP 의 POST 방식(method)으로 서버에게 전달된다.

⑤ 서버에서 *CGI(Common Gateway Interface)* 프로그램에 의해 데이터베이스에서 비밀번호를 얻어와 사용자를 확인한다.

⑥ 서버는 인증 정보를 쿠키로 생성하여 사용자에게 전달한다. 쿠키에 포함되는 정보는 서버마다 다르나 보통 사용자의 계정을 비롯한 개인 정보를 가지고 있다.

⑦ 이후 인증이 필요한 경우 쿠키를 통해 확인한다.

이 인증 기법은 계정과 비밀번호가 서버에게 전달되는 과정에서 스니핑에 의해 쉽게 노출되는 취약성을 가지고

있다. 아래는 실제 인증을 요구하는 패킷의 일부이다.

표 1. 인증을 요청하는 패킷

```
POST /Mail-bin/login.cgi?dummy=997371263 HTTP/1.1.
Content-Type: application/x-www-form-urlencoded.
Host: www.???????.net.
id=kagi&pw=1234
```

패킷을 살펴보면 HTTP 의 POST 방식을 통해 login.cgi 라는 파일을 요청하고 있음을 알 수 있다. 마지막 줄에 나타난 데이터 부분에서 "id"라는 항목으로 계정이 표시되고 있고, "pw"로 비밀번호가 나타나 있음을 알 수 있다.

최근 인터넷을 통해 간단한 사용법을 가진 스니퍼(sniffer)가 확산되면서 전문적인 지식이 없는 사람도 쉽게 다른 사용자의 계정과 비밀번호를 가로채는 것이 가능해졌고, 특히 스니핑의 특성상 인터넷보다는 인트라넷에서 더욱 쉽게 행해질 수 있으므로 회사나 기관에서 더욱 주의가 요구된다.

또한 최초 인증 후에 사용되는 쿠키를 통한 인증 역시 취약점을 가지고 있다. HTTP 의 state-less 한 특성을 보완하기 위해 사용되는 쿠키 역시 패킷 안에 일반 텍스트로 표현되어 있어 위와 같은 스니핑을 통해 획득할 수 있고, 악성 코드에 의한 유출도 가능하다. 때문에 쿠키를 사용할 때는 비밀번호나 신용 카드 정보 등 기타 중요 개인 정보를 포함해서는 안되며 클라이언트의 IP 주소나 타임 스탬프를 사용하여 replay 공격에 대비하고 서버가 가진 비밀키로 암호화하여 무결성과 비밀성을 보장하는 등의 대책이 필요하다. 또 서버의 자원을 통해 클라이언트를 관리하는 세션기능을 사용할 수도 있다. [3]

2.2 HTTP 에 포함된 기본적인 인증 기법

현재 사용되는 HTTP 1.1 의 경우 그 안에 사용자 인증 기법을 지원하는 기능을 포함하고 있다.(HTTP 1.0 에도 포함되어 있음) 인증 헤더를 통해 웹 서버와 브라우저 사이에서 이루어지는 이 기법은 역시 스니핑에 의해 취약성을 가지고 있다. 헤더 안에 사용자 계정과 비밀번호가 일반 텍스트로 포함되기 때문이다.[1]

2.3 HTTP 의 메시지 축약을 사용한 인증

HTTP 에 기본적으로 포함된 인증 기법의 취약성의 보완을 위해 소개된 메시지 축약을 사용한 인증 기법은 challenge-response 방식에 기초한 기법으로 비밀번호가 네트워크를 통해 전달되는 근본적인 문제점을 해결하고 있다. 추가된 authorization header 를 통해 인증이 필요한 페이지에서 인증을 처리하게 된다. 인증 과정은 다음과 같다.

① 회원 가입 시 계정과 비밀번호를 웹 서버에 전달한다.

② 웹 서버는 계정과 비밀번호를 지정된 파일에 보관한다.

③ 인증이 필요한 경우 서버는 메시지 축약에 사용되는 nonce 를 포함한 인증 헤더를 보낸다.

④ 브라우저에서 인증 헤더를 받으면 사용자에게 계정과 비밀번호를 요구한다.

⑤ 사용자가 계정과 비밀번호를 입력하면 브라우저는 메시지 축약을 포함한 인증 요청 헤더를 생성해 서버에게 전달한다.

⑥ 서버는 지정된 파일에서 비밀번호를 얻어 메시지 축약을 검증한다.

⑦ 이후 인증은 인증 정보 헤더를 통해 이루어 진다.

메시지 축약은 기본적으로 MD5 알고리즘을 사용해 이루어지며 기존의 비밀번호에 기초한 인증 기법들과 마찬가지로 비밀번호의 최초 교환 방법에 대한 문제와 challenge-response 기법이 가진 replay 공격에 대한 취약성이나 brute-force 공격 등에 대한 취약성을 가지고 있다. 물론 앞서 살펴본 두 가지 인증 기법보다 안전하다.[2]

3. 메시지 축약을 사용하는 인증의 변형된 설계

지금까지 살펴본 인증 기법 중 가장 많이 사용되는 것은 첫번째로 살펴본 기법이다. 다른 두 기법은 브라우저와 웹 서버에서의 지원이 필요하고 관리 등이 복잡하기 때문에 또, 보안에 대한 무관심 속에서 가장 많이 사용되고 있다. 이를 위해 메시지 축약을 사용하는 인증 기법과 계정, 비밀번호, 쿠키를 사용하는 인증기법을 통합하여 구현할 수 있다. 즉 기존 브라우저와 웹 서버 사이에서 이루어지는 메시지 축약 및 검증을 애플릿과 CGI 프로그램에서 동작하도록 구현할 수 있다. 인증 기법의 구조와 전체 과정은 다음과 같다.

① 회원 가입 시 사용자 계정과 비밀번호를 포함한 사용자 정보를 웹 서버에 전달한다.

② 웹 서버는 사용자 정보를 별도의 데이터베이스에 저장한다.

③ 사용자 인증이 필요한 경우 사용자는 애플릿을 통해 생성된 양식에 계정과 비밀번호를 입력한다.

④ 애플릿은 계정과 비밀번호, 파라미터로 입력된 nonce 를 이용해 메시지 축약을 생성하여 서버에 전달한다.

⑤ 서버에서 CGI 프로그램에 의해 nonce 의 유효성을 검사하고 데이터베이스에서 비밀번호를 얻어와 메시지 축약을 검증한다.

⑤ 서버에서 인증과 관련한 쿠키를 생성 암호화하여 사용자에게 전달한다.

⑥ 이후 인증이 필요한 경우 쿠키를 통해 확인한다.

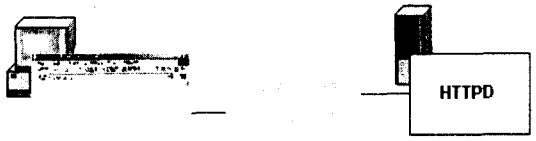

그림 1. 기존의 메시지 축약을 사용한 인증 기법의 구조

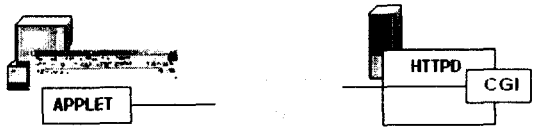

그림 2. 메시지 축약을 사용한 인증 기법의 변형된 구조

이처럼 두 기법을 통합하여 인증이 이루어 지며 부가적으로 보완을 위해 생성된 nonce 를 검증할 수 있도록 하고 다양한 축약 알고리즘이 사용 가능하도록 하며 쿠키 역시 암호화되어 보안이 이루어진다.

4. 메시지 축약을 사용하는 인증의 변형된 구현

애플릿은 JDK 1.2 를 사용해 구현하였고 서버는 Apache 와 JSP(Java Server Page)를 사용하였다. 메시지 축약을 수행하는 알고리즘은 애플릿의 파라미터로 지정되는 알고리즘을 사용하도록 하여 MD5, SHA 등을 지원하고 nonce 는 아래 코드에서처럼 시간과 서버 정보에 대한 메시지 축약을 포함하고 있어 유효성 여부의 검사가 가능하며 Base64 코드로 표현되어 전송된다.

표 2. nonce 생성 소스 코드

```
nonce=(new Date()).getTime().toString()+":" + md.digest(hostinfo);
(new Base64Decoder()).decodeBuffer(nonce.getBytes());
```

애플릿에서 계정과 비밀번호가 입력되면 입력된 값과 nonce 를 지정된 알고리즘을 통해 축약을 수행하여 전송된다. 검증이 완료된 후 발행하는 쿠키는 시간과 클라이언트의 IP 주소, nonce, 축약된 결과 등이 포함되어 서버의 비밀키로 암호화되어 전달된다.

표 3. 암호화된 쿠키에 포함된 정보

이름	설명
id	사용자의 계정
name	사용자의 이름
time	로그인 시각
ip	로그인한 클라이언트의 IP 주소
algorithm	메시지 축약을 생성한 알고리즘
response	로그인 시 생성된 메시지 축약
nonce	로그인 시 전달된 nonce

이후의 인증과정은 암호화된 쿠키를 복호화하여 유효시간과 IP 주소 확인 및 nonce 에 대한 축약을 검증해 이루어 진다.

5. 결론

본 논문에서 구현된 결과를 통해 현재 대부분의 사이트에서 사용되고 있는 인증 기법의 스니핑에 대한 취약성을 최소의 수정을 통해 방어할 수 있다. HTML 태그를 이용한 입력 양식대신 애플릿을 위한 태그로 변경하고 비밀번호를 확인하는 루틴이 축약을 검증하는 루틴으로 변경된다. 때문에 현재 서비스가 이루어지고 있는 웹 사이트에도 바로 적용이 가능하다.

향후에는 계정과 비밀번호를 사용하지 않고 X.509 포맷의 인증서를 사용하는 인증과 다양한 접근 통제 정책을 적용할 수 있는 기법에 대한 연구가 요구된다. 이를 통해 최근 보안과 프라이버시 문제에 대한 요구가 점차 증가하고 있는 웹 분야에서 안전한 서비스를 제공할 수 있다.

6. 참고문헌

[1] R.Fielding 외, Hypertext Transfer Protocol - HTTP/1.1, RFC 2616, 1999.06

[2] J.Franks 외, HTTP Authentication : Basic and Digest Access Authentication, RFC 2617, 1999.06

[3] CERTCC-KR, KA-2000-041- 악성 코드에 의한 HTTP Cookie 유출 문제점 및 대책, CERTCC-KR 권고문, 2000.11