사용자가 변경하는 일회용 패스워드 알고리즘에 관한 연구

김영수*, 박연식**, 임재홍*

A Study on One Time Password Algorithm to change by end-user

Yeong-su Kim, Yeoun-sik Park, Jae-hong Yim

요 약

패스워드는 컴퓨터보안의 첫 단계이다. 패스워드 보안이 허술해지면 방화벽은 사실상 무용지물이나 다름없다. 그러나 일반 사용자들은 고난이도의 패스워드 관리가 불가능하므로 이에 대한 보완이 필요하다. 본 논문에서는 클라이언트 측에서 난수를 이용하여 일회용 패스워드를 생성하는 알고리즘을 사용하였다. 이것은 사용자들이 패스워드 관리를 용이하게 할 수 있으며, 서버해킹에 대한 불안을 해소시킬 수 있다.

Abstract

The password is the first step for computer security. If security of password is unimportant even constructing of fire-wall, it is useless. But end-user is not able to manage a high-difficulty password. So complement for password management is needed. In this paper, algorithm which produces one time password by using random number in client is used. Not only this is easy for end-user to manage password, but also this can eliminate insecurity for server hacking.

* 한국해양대학교

** 경상대학교 정보통신공학과, 해양산업연구소

접수일자 : 1999년 12월 31일

Ⅰ. 서 론

원격제어(Remote Control)의 대중화는 네티즌들에게 편리성을 제공하는 반면에 악의가 있는 해커들의 공격통로가 되고있다. 이로 인하여 네티즌들과 네트워크 관리자들은 항상 해킹의 위험 속에서 살고 있다.[5]

더구나 Unix 소스의 공개로 인하여 해킹은 더욱 쉬워졌다.[6] 해커들은 호스트에 접속 후 OS의 버그를 이용하므로 초기단계에서의 해킹방지는 매우 중요하다. 그런데 일반 사용자들은 고난이도의 패스워드 사용을 회피하므로 패스워드 관리에 대한 실효성을 거두지 못하고 있다. 또한 네트워크 상에서 사용자 ID와 패스워드를 훔치기 위해 스니퍼(sniffer)를 사용하므로 일반 사용자들에게는 대책이 전혀 없다.[7]

본 논문은 클라이언트에서 난수를 발생시켜 일회용 패스워드를 생성하여 서버에 접속하고 접속 후에는 서버에 등록된 패스워드를 변경시켜주는 알고리즘에 대해서 제시하고자 한다. 이 때 클라이언트에서 서버로 전송하는 패스워드는 비밀키의 사용과 XOR를 이용하여 암호화하였다.

이 방법은 사용자가 패스워드를 관리하게 함으로써 서버의 부담을 줄여주는 효과가 있으며, 패스워드를 사용자 자신이 만들어 내는 것이 아니고 프로그램에 의해서 난수발생에 의해서 일회용으로 생성되므로, 양질의 패스워드를 만들어 낼뿐만 아니라 사용자가 간편하게 사용할 수 있다.

본 논문의 구성은 Ⅱ장에서 기존의 일회용 패스워드에 대한 기법을 알아보았고, Ⅲ장에서 제안하는 패스워드 관리의 특징과 구현된 알고리즘을 제시하였으며, Ⅳ장에서 결론을 맺었다.

Ⅱ. One Time Password System

일반적으로 호스트에 접속할 때마다 동일한 패스워드를 매번 사용하므로 네트웍의 도청에 대한 위험을 안고 있으나, 원 타임 패스워드는 호스트에 로그인 할 때마다 항상 다른 패스워드를 사용함으로써 도청의 위험을 해결하고 있다. 일단 한 번 사용된 패스워드는 재사용이 불가능하므로 침입자가 네트워크 도청을 통해서 패스워드를 알아내어도 더 이상 이용할 수 없게 된다[1].

원 타임 패스워드 시스템을 구현하는 방법에는 다음과 같은 것이 있다.

① 동기화 된 시간을 유지하여 Time-Stamp를 사용
② 양쪽에 있는 임의의 패스워드 리스트 내의 위치를 동기화 하여 패스워드를 사용
③ Sequence generator의 상태를 동기화 하여 임시적인 sequence number 사용
④ Challenge-Response Schemes 이용

원 타임 패스워드 기법을 이용해 구현한 제품에는 여러 가지가 있겠으나, Challenge-Response Schemes를 이용한 방법을 요약하면 다음과 같다.

① 사용자가 로그인을 시도하면 서버 쪽에서 임의의 Challenge 메시지를 생성해서 사용자에게 전송한다.
② 사용자는 PIN(Personal Identification Number)과 Challenge를 이용하여 서버에 전송할 원타임 패스워드를 생성하고, 서버에게 Response 메시지를 전송한다.
③ 서버는 동일한 Challenge와 등록된 사용자의 정보를 이용해 원타임 패스워드를 생성한 후 사용자가 전송한 Response와 비교하여 로그인을 허락한다[3][4].

Ⅲ. 제안한 알고리즘

1. 알고리즘의 특징

본 알고리즘의 특징은 다음과 같다.

① 클라이언트가 정해놓은 패스워드를 서버에서 등록해 놓으면 이 후부터의 패스워드 관리는 클라이언트에서 전담한다
② 클라이언트에서 필요로 하는 일회용 패스워드는 난수발생에 의존하므로 사용자가 다양한 종류의 패스워드를 만들기 위해 노력할 필요가 없다.
③ 생성된 패스워드는 비밀키와 XOR에 의해 암호화되어 전송되므로 안전성을 높여준다.
④ 또한 비밀키는 서버관리자와 협의하여 주기적으로 변경하면 안전성을 더욱 높일 수 있다. 물론 이 때의 비밀키는 난수발생에 의하

여 생성시켜야 좋다.

⑤ 기존의 일회용 패스워드는 클라이언트의 로그인을 허락하기 위해서 서버-클라이언트간에 challenge 메시지를 주고받는 과정이 필요했으나, 본 알고리즘에서는 그럴 필요가 없다.

⑥ 서버가 해킹을 당하더라도 패스워드 관리는 클라이언트에서 주관하므로 피해를 최소화할 수 있다.

⑦ 서버가 해킹을 당하여 패스워드가 변경되었다면 클라이언트가 접속시 단번에 알아낼 수가 있다.

⑧ 해커가 서버를 해킹을 해도 동일계정을 지속적으로 사용할 수는 없게된다.

서버로부터 계정과 패스워드를 받은 후 자신의 PC에 클라이언트를 설치한 후에 이루어지는 본 알고리즘의 구현 체계는 다음과 같다.

① 서버에 접속할 때마다 클라이언트에서는 난수 발생에 의하여 일회용 패스워드를 생성한다.

② 생성된 패스워드는 비밀키와 XOR 함수에 의해 암호화하여 서버로 전송한다.

③ 서버에서는 클라이언트에서 전송된 패스워드와 자신이 보관하고 있는 패스워드가 일치하면 로그인을 허락한다.

④ 서버와 클라이언트에서는 새로이 생성된 패스워드를 저장한다.

2. 알고리즘의 구현

```
#include <stdio.h>
#include <stdlib.h>
main()
{
/* 변수 선언 */
  FILE *fopen(), *fpc, *fps, *fpc1, *fps1;
  char login();
  char c_buff[16], s_buff[16];
  char sec_key[8], t_passwd[16], yn;
  int i, ii, j;

  i  = 8;
  ii = 16;

/* 암호화를 위한 비밀키의 setting */
  sec_key[0] = '$';
  sec_key[1] = 'k';
  sec_key[2] = 'x';
  sec_key[3] = '=';
  sec_key[4] = 'k';
  sec_key[5] = '5';
  sec_key[6] = 's';
  sec_key[7] = 'g';

/* 클라이언트측의 file open 및 reading */
  fpc  = fopen("c_pass.pwd", "r");
  fpc1 = fopen("c_pass1.pwd", "w");
  if (fpc1 == NULL)
    {
      printf("ERROR!!! Can't open File
            (c_pass.pwd)");
    }
  fscanf(fpc, "%16c", c_buff);

/* 클라이언트에서의 난수 생성 */
  randomize();  /* initializing */
  for (j=0; j<i; j++)
      {
       t_passwd[j]   = c_buff[j];
       t_passwd[7+j] = (rand() % 129)
                        ^ sec_key[j];
      }

/* 서버측의 file open 및 reading */
  fps  = fopen("s_pass.pwd", "r");
  fps1 = fopen("s_pass1.pwd", "w");
  if (fps == NULL)
    {
     printf("ERROR!!! Can't open File
           (s_pass.pwd)");
    }
  fscanf(fps, "%16c", s_buff);

/* 로그인 허락여부를 위한 함수 호출 */
  yn = login(c_buff, s_buff, ii);
```

```
/* 클라이언트에게 허락여부를 통보 */
  if (yn == 'y')
    {
     printf("Login successful...");
/* 사용자의 다음 로그인을 위한 자료 갱신 */
     for (j=0; j<16; j++)
         {
          fprintf(fpc1, "%c", t_passwd[j]);
          fprintf(fps1, "%c", t_passwd[j]);
         }
     }
     else printf("Login failed...");
/* 모든 file을 close */
  fclose(fpc);
  fclose(fps);
  fclose(fpc1);
  fclose(fps1);
}

/*-------------------------*/
char login(c_buff, s_buff, i)
  char c_buff[], s_buff[];
  int i;
{
  char yn;
  int j;
  yn = 'y';

  for (j=0; j<i; j++)
      {
       if (c_buff[j] != s_buff[j])
         {
          yn= 'n';
          break;
         }
       }
  return(yn);
}
```

본 논문의 알고리즘을 구현함에 있어서 한 개의 알고리즘으로 통합하여 구현하였다. 그것은 구현의 편리성과 전체 알고리즘의 흐름을 쉽게 제시하기 위함이다. 클라이언트 쪽에서 일어나는 동작과 서버 쪽에서 일어나는 동작을 한 눈에 볼 수 있다. 실제로 웹 상에서는 서버/클라이언트라는 두 개의 독립된 알고리즘으로 존재하면서 정보를 공유하게 된다.

본 절의 알고리즘의 핵심은 클라이언트에서의 난수발생과 로그인 확인, 그리고 정당한 사용자의 다음 로그인을 위한 데이터베이스의 갱신에 있다. 난수발생 루틴에서는 난수발생에 의해서 로그인 할 때마다 새로운 패스워드를 생성되며, 비밀키에 의해서 암호화되므로 보안의 비도(秘度)를 높여 준다.

login 함수의 호출루틴에서는 클라이언트에서 보낸 암호화된 패스워드와 서버에서 보관중인 암호화된 패스워드를 비교하여 정당한 사용자인지를 확인한다. 그리고 확인결과에 따라 허락여부의 코드를 반환시켜 준다.

데이터베이스 갱신 루틴에서는 다음 로그인을 위해서 데이터베이스를 갱신하도록 한다. 이 과정으로 인하여 일회용 패스워드의 기능을 제대로 수행할 수가 있게된다. 정당한 사용자는 자신의 다음 로그인을 위해서 새로운 패스워드를 서버로 보내고 자신도 동일한 패스워드를 보관한다. 이렇게 하여 클라이언트와 서버는 항상 새로운 패스워드를 가질 수 있게된다.

3. 성능 분석

기존의 일회용 패스워드 알고리즘에서는 <그림 3.1>과 같이 4단계를 거친 후에 클라이언트의 접속여부를 결정한다. 그 과정 속에는 서버와 클라이언트 사이에서 Challenge 메시지를 주고받는 절차가 있다. 이것은 그만큼 네트워크의 사용이 증가되며 네트워크의 트래픽 상태에 따라 클라이언트의

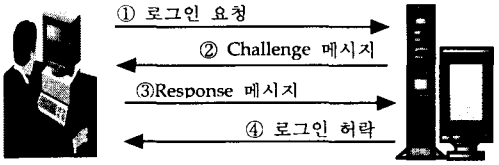


그림 3.1 일회용 패스워드 시스템의 로그인
Fig 3.1 Login of One Time Password System

대기시간이 길어질 수 있으며, 또한 서버에서 Challenge 생성과 암호화의 수행은 서버의 부하를 증가시키는 직접적인 요인이 된다.

그러나, 개선된 알고리즘에서는 <그림 3.2>와 같이 2단계를 거친 후에 클라이언트의 접속여부를 결정한다. 클라이언트는 로그인 요청과 동시에 자신이 생성한 암호화된 패스워드를 서버로 보낸다. 그러면 서버에서는 확인 절차를 거쳐 정당한 사용자로 확인되면 기존의 패스워드를 새로이 보내온 패스워드로 갱신하고, 로그인을 허락한다.

본 알고리즘은 서버의 부하를 현저히 줄일 수 있고, 네트워크의 사용횟수도 절반정도로 줄일 수 있다. 그러므로 결과적으로 클라이언트의 대기시간을 상당히 단축시킬 수가 있게 된다.

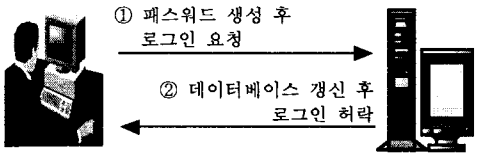


그림 3.2 개선된 알고리즘의 로그인
Fig 3.2 Login of Improvement Algorithm

Ⅳ. 결 론

본 논문에서는 기존의 일회용 패스워드 시스템의 Challenge 메시지에 해당되는 부분을 클라이언트에서 처리하도록 하였다. 사용자들은 호스트에 접속할 때마다 변하는 패스워드를 자신의 하드디스크나 디스켓에 저장한다. 만약 호스트가 해킹을 당해 패스워드가 변경되었다면 사용자들은 접속이 불가능하게 되므로 즉각적으로 이를 판단할 수 있게된다.

본 논문에서는 기존의 일회용 패스워드 알고리즘보다 간단한 확인 절차를 거치므로 네트워크의 사용횟수와 서버의 부하를 줄임으로 말미암아 클라이언트의 대기시간을 상당히 단축시키도록 하였고, 패스워드를 사용자가 관리하는 방안을 제시하였다.

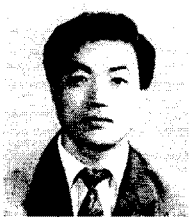
또한 해킹의 문제에 있어서는, 해커가 개개인의 PC를 해킹하기란 여간 피곤하지 않을 것이다. 더구나 사용자들이 디스켓에 패스워드를 저장한다면 해킹은 거의 불가능할 것이다.

그런데 여기에도 문제유발의 가능성은 생각해 볼 수 있다. 개인 PC의 해킹은 피곤한 일이겠지만, 개인 PC는 외부침투에 대해서 거의 무방비하며, 자신의 디스켓이 손상을 입었을 경우에는 번거로운 절차를 거쳐야 한다.

그러나 이러한 문제는 향후 새로운 기록매체의 등장과 함께 해소될 수 있을 것이다.

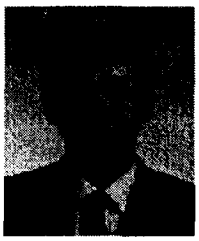
참고문헌

1. "A One-Time Password System RFC", ftp://ds.internic.net/rfc/rfc1938.txt
2. http://www.ietf.cnri.reston.va.us/html.charters/otp-charter.html
3. Brent Chapman and Elizabath D. Zwicjy, "Building Internet Firewalls", pp. 359-365
4. 한국정보보호센터, http://www.kisa.or.kr/technology/sub4/password.htm
5. 한국정보보호센터 기술본부 기술대응팀, 웹서버 안전 운영대책, 1998
6. http://home.taegu.net/~zeus01/hack9.htm
7. 월간 인터네트, 해킹의 최신 형태와 방지 테크닉, http://wwwcs.dongguk.ac.kr/~dh999/Security/Doc/Text/hack_sec.html



김 영 수(Yeong-Su Kim)
한국해양대학교 전자통신공학과
(Korea Maritime University Dept of Eletronics and Communication Engineering)
관심분야 : 지역정보망, 그룹웨어, 데이터베이스
연구세부분야 : 컴퓨터 네트워크
650-160 경남 통영시 인평동 445
경상대학교 해양과학대학 LAN관리실
0557-640-3016 011-573-3071

박 연 식(Yeoun-Sik Park)
경상대학교 정보통신공학과, 해양산업연구소
(Gyeong-Sang National University Dept of Information and Communication Engineering Major, The Institute of Marine Industry)
관심분야 : 수중화상통신
연구세부분야 : 컴퓨터 네트워크
650-160 경남 통영시 인평동 445
경상대학교 해양과학대학 정보통신공학과
(0557-640-3137 011-886-3137)



임 재 홍(Jae-Hong Yim)
1986년 2월: 서강대학교 전자공학과 졸업(공학사)
1988년 8월: 한양대학교 대학원 전자공학과 졸업(공학석사)
1995년 2월: 한양대학교 대학원 전자공학과 졸업(공학박사)
1995년 3월 ~ 1997년 2월: 한국해양대학교 전자통신공학과 전임강사
1997년 3월 ~ 현재: 한국해양대학교 전자통신공학과 조교수
* 관심분야: 컴퓨터네트워크, 분산 컴퓨팅, 그룹웨어