تکلیف دوم امنیت شبکه

نگار سخایی

۹۵۲۸۰۰۳

مکانیزم احراز اصالت به کمک یک فایل (USERDATA.txt) که در آن نام کاربری و hash رمزعبور همه کاربران ذخیره شده است کار می‌کند. وقتی کاربری به سرور درخواست می‌دهد، برخلاف حالت ساده قبل، هم نام کاربری و هم رمز عبور خود را برای سرور ارسال می‌کند. از آنجایی که حمله‌کننده نمی‌تواند ارتباط بین کاربران و سرور را شنود کند، نمی‌تواند این رمز عبور را بدست بیاورد.

سرور پس از دریافت این اطلاعات، سطر مربوط به این کاربر را در فایل USERDATA.txt پیداکرده و مقدار hash پسورد دریافتی را با مقداری که در فایل وجود دارد مقایسه می‌کند. این کار باعث می‌شود که نه تنها درستی رمز عبور چک شود، بلکه حمله کننده نتواند حتی با وجود دسترسی به فایل کاربران رمز عبور درست را پیدا کند. (البته با فرض امنیت تابع hash استفاده شده.)

البته مشخصا، فرض شده است که در سیستم، نام و رمز عبور تمام کاربران و تعداد آن‌ها از قبل تعیین و فیکس شده است و کاربر جدیدی وارد سیستم نخواهد شد.

```
try {
    Scanner in = new Scanner(new File( pathname: "USERDATA.txt"));
    while (in.hasNextLine())
    {
        String s = in.nextLine();
        String[] sArray = s.split( regex: ",");
        if (this.username.equals(sArray[0]))
            return computeHash(password).equals(sArray[1]);
    }

    in.close();

} catch (FileNotFoundException e) {
      e.printStackTrace();
}
```

```java
private String computeHash(String pass) {
    HashFunction hf = new HashFunction();
    hf.reset();
    hf.update(pass.getBytes());
    byte[] passByte = hf.digest();
    BigInteger bi = new BigInteger( signum: 1, passByte);
    return String.format("%0" + (passByte.length << 1) + "X", bi);
}
```

طبیعتا با این ملاحظات، دانستن تعداد و نام کاربران، کد منبع و الگوریتم هیچ اطلاعات اضافه‌ای به کاربر مخرب اضافه نخواهد کرد. چون قابلیت شنود وجود ندارد، مشکل replay attack و دیگر مشکلات مشابه نیز پیش نخواهد آمد. به علاوه چون فرض شده به فایل‌های سیستم دسترسی ندارد، به هیچ عنوان نمی‌تواند رمز عبورها را ببیند و hash کردن آن‌ها تاثیر خاصی نخواهد گذاشت.