보안성과 유용성을 고려한 RTT기반의 패스워드 인증 방안

이희정° 이금석
동국대학교 컴퓨터공학과
{cle3°, kslee}@dongguk.edu

RTT based Password Authentication regarding Security and Usability

Heejeong Lee° Keumsuk Lee
Dept. of Computer Engineering, Dongguk University

요 약

패스워드 인증방법은 온라인 사전 공격에는 약하다는 단점이 있음에도 불구하고, 사용하기에 가장 편리하다는 장점 때문에 오늘날 가장 일반적으로 사용되는 인증방법이다. 기존의 패스워드 인증방법처럼 편리하게 이용할 수 있으면서도 보다 보안성을 높이는 방안으로 RTT를 이용한 인증 프로토콜이 제안되어 왔다. RTT를 이용한 인증 프로토콜은 사용자가 아이디와 패스워드를 입력할 뿐만 아니라, 자동 프로그램과 사람을 구별할 수 있는 질문에 응답하게 함으로써 자동 프로그램으로 공격하는 것을 막는다. 그러나 이 프로토콜에 이용되는 RTT의 여러 모델들에서 간단한 이미지는 공격 프로그램으로 공격 가능성이 있고, 복잡한 이미지는 사용자 입장에서 유용성이 취약함을 보인다. 따라서 이런 모델들의 취약성을 분석하여 공격에 대해서는 강하면서도 사용자들이 이용하기에는 편리하도록 하기위해 새로운 모델을 제안하고, 보안성과 유용성을 고려한 RTT기반의 인증방안을 제안한다.

1. 서 론

인터넷의 발전으로 인해 실생활에서 일어나는 일들이 가상공간을 통해 해결될 수 있게 되었다. 이런 변화에서 가장 중요하게 고려되어야할 것은 네트워크 이용시에 사용자가 정당한 사용자인지를 인증해주는 기술이다. 보안성을 강화하기 위해 현재 사용되는 인증 방법들을 살펴보면, 크랙킹 프로그램을 이용하여 일정한 자릿수 이상이나 추측하기 힘든 패스워드를 사용하도록 사용자들에게 강요하거나[1,2], 시스템이 보안성이 강한 패스워드를 직접 사용자에게 부여하는 방법이 있는데 이들은 사용자가 기억하기 어렵다는 단점이 있다. 스마트카드나 하드웨어 디바이스 등과 같은 하드웨어 지원 방법을 사용하거나, 인증서를 발급받고 비밀키를 이용하는 소프트웨어 지원 방법도 있으나, 잃어버렸을 때 문제가 되고 추가적인 디바이스를 필요로 하며, 다른 기계에서는 이용할 수 없고 비밀키의 안정성이 보장되어야만 한다. 홍채·지문인식이나 키보드 입력 패턴을 이용한 생체학적 인증방법은 가장 강력한 보안성을 제공하지만 신뢰성과 이식성에서 문제가 있다[3,4]. 이와 같이 다양한 인증방법들이 제공되고 있는데, 그중에서도 널리 사용되고 있는 패스워드 기반 인증방법에서 보안성을 향상시키기 위해서는 공격자가 패스워드를 전송 중에 엿듣는다거나, 크랙 프로그램을 이용한 오프라인 사전 공격을 하는 등의 공격을 막아줘야 한다.

이 논문에서는 SSL과 같은 암호화된 통신채널을 이용하여 패스워드 전송 중에 엿듣기를 방지하고, 패스워드 파일에 접근하지 못하도록 제한을 두어 오프라인 사전 공격을 막아주기 때문에, 온라인 사전 공격만을 다루는 환경으로 가정한다. 온라인 사전 공격은 네트워크에 접근만 가능하다면 공격자가 특정 아이디의 접근권한을 획득하려는 게 아니라, 어떤 아이디라도 접근권한만을 획득하면 되는 것이다. 이는 글로벌 패스워드 공격으로 이루어지므로 응답시간을 지연하거나 계정을 막는 대응책은 무의미하다. 따라서 사용자 관점에서는 기존의 패스워드 인증방법처럼 편리하게 이용할 수 있으면서도, 기존의 방법보다 보안성을 높이는 방안으로 사람과 기계를 구분할 수 있는 인증방안을 제안하였다.

2장에서는 관련연구로써 RTT를 이용한 인증방안과 RTT 모델들을 소개한다. 3장에서는 보안성과 유용성을 고려한 RTT기반의 개선된 인증방안을 제시하고 마지막으로 결론 및 향후 연구과제를 제시한다.

2. 관련연구

2.1 RTT를 이용한 인증방법

2.1.1 RTT를 이용한 기존 프로토콜

기존 프로토콜은 사용자가 아이디와 패스워드를 입력하는 인증과정을 거치기전에, Reverse Turing Tests(RTT)에 응답하도록 요청받는다[5]. 이 방법은 자동화된 프로그램으로 많은 패스워드를 시도해보는 글로벌 패스워드 공격을 막아준다. 따라서 보안성 측면에서 보면 패스워드를 추측하기 위해서는 RTT에 응답해야만 하는데, 이 RTT에 대한 응답을 추측하기가 매우 어렵다. N개의 패스워드에 대해 N번의 RTT에 응답할 수 있어야 하므로, 공격 속도가 매우 느려진다.

하지만 이 방법은 유용성 측면에서 사용자가 일반 패스워드 인증방법에 비해 매번 RTT에 응답해야 하는 번거로운 일을 해야만 하고, 확장성 측면에서 매번 로그인 시도마다 서버는 많은 RTT를 생성해야만 하므로 취약하다.

2.1.2 RTT를 이용한 향상된 프로토콜

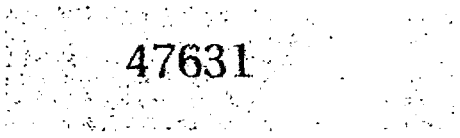
대부분의 사용자는 제한된 컴퓨터를 이용하고 온라인 사전 공격은 다른 컴퓨터로부터 행해진다는 점에 착안하여, 서버가 쿠키에 저장되어 있는 정보를 이용하여 기계를 구분해낼 수가 있다. 쿠키에 저장되는 정보는 사용자 아이디, 패스워드, IP 주소, 쿠키의 유효기간이 되는데, 처음 초기화는 사용자가 올바른 아이디와 패스워드를 입력하고 RTT에도 응답할 수 있었을 때 서버에 쿠키정보가 저장된다[6].

인증은 사용자가 아이디와 패스워드를 올바르게 입력했을 때, 쿠키정보가 맞고 유효기간도 지나지 않았다면 적법한 사용자로 인증되어 서버에 대한 접근권한이 부여된다. 하지만 쿠키가 없거나 유효기간이 지난 쿠키거나 사용자 아이디에 대한 쿠키정보가 다를 때에는, 패스워드를 올바르게 입력하였다 하더라도 서버가 RTT를 생성하여 사용자에게 응답하도록 한다. 이때, 사용자가 RTT에 대해서도 올바르게 대답해야만 서버에 접근권한을 부여받을 수 있다. 만약 아이디와 패스워드가 맞지 않는다면 사용자는 확률 p로써 RTT에 응답하도록 요청되고, RTT에 대한 응답이 올바른지에 상관없이 서버에 대한 접근권한은 거절된다. 1-p의 확률로는 사용자는 서버에 대한 접근권한이 즉시 거절된다.

이와 같이 향상된 프로토콜은 유용성 측면에서 보면 사용자가 새로운 컴퓨터에서 처음 로그인을 시도할 때와, 입력한 패스워드가 올바르지 않을 때에 p의 확률로 RTT에 응답하면 되므로 사용자는 RTT를 사용하지 않는 일반적인 로그인 프로토콜과 거의 비슷하게 사용할 수 있어 유용성의 저하가 없다. 확장성 측면에서 보면 서버는 올바르지 못한 로그인 시도중에서 확률 p만큼만 RTT를 생성하면 되므로, 매번 로그인 시도마다 RTT를 생성하는 프로토콜에 비해 확장성이 향상되었다. 서버에서 RTT를 캐싱을 이용하면 사용자가 올바른 아이디와 패스워드를 입력했을 때, 서버에서는 RTT를 검색하여 같은 RTT를 요청하므로 새 컴퓨터에서 로그인을 시도하더라도 매번 RTT를 생성할 필요가 없으므로 확장성이 향상되고, 사용자는 같은 RTT에 응답하도록 요청받으므로 유용성도 향상되는 두 가지 측면에서 이득을 얻는다[7].

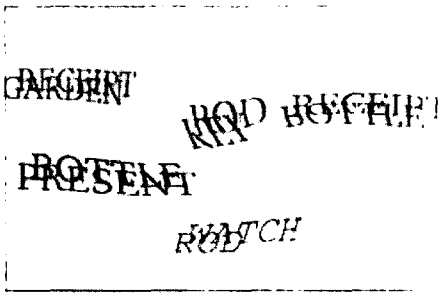
2.2 RTT 모델 관련 연구

RTT로써 사용할 수 있는 여러 모델들이 CAPTCHA 프로젝트에 소개되어있다[8]. Yahoo!나 Paypal과 같은 사이트들에서는 현재 자동으로 아이디를 생성하는 것을 막기위해 [그림 1]과 같은 EZ-Gimpy 모델을 사용하는데, 이는 사용하기에는 편리하나 간단한 문자열로만 이루어진 이미지는 자동화된 공격 프로그램으로 이미지를 스캔하여 문자열을 추측할 수 있다는 가능성이 있다. 따라서 간단한 문자열이 아닌 다른 대안방안이 필요하다.



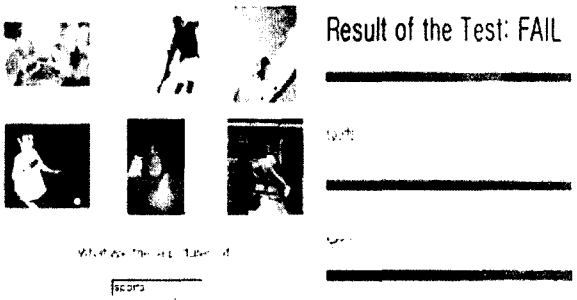
[그림 1] 문자열 이미지인 EZ-Gimpy의 예제

복잡한 문자열로 이루어진 [그림 2]와 같은 Gimpy 모델은 간단한 문자열 이미지에서 취약했던 보안성을 높여주는 모델이라 할 수 있는데, 이 모델은 세 단어를 모두 맞춰야만 하고, 가독성에 있어서 어렵다는 단점이 있다.



[그림 2] 복잡한 문자열 이미지인 Gimpy의 예제

문자열 이미지가 보안상 취약하기 때문에 또 다른 대안방법으로 그림 이미지를 보여주는 [그림 3]과 같은 Pix 모델이 있다. 이 모델은 5개에서 6개 정도의 그림 이미지를 보여주고 그림이 나타내는 적절한 단어를 입력하도록 하는 RTT이다. 이 모델은 보안성 면에서는 강화되었지만, 유용성면에서 답을 유추해내는데 어려움이 따르고 영어단어를 모르면 이용이 불가능하다는 취약성을 보인다.



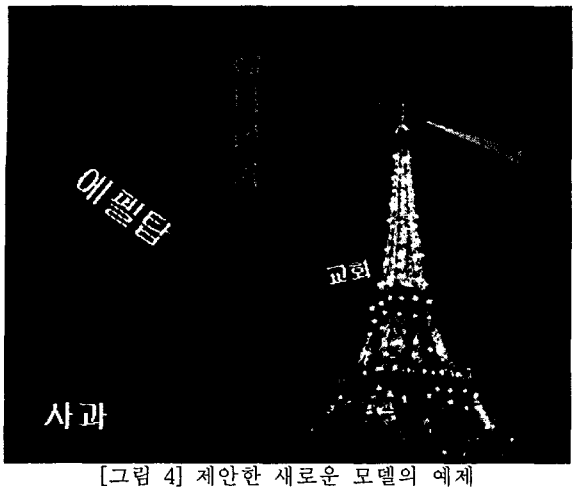
[그림 3] 그림 이미지인 Pix의 예제

3. 제안 모델

RTT 모델 관련 연구에서 살펴보았듯이, 단순한 문자열로만 이루어진 EZ-Gimpy 모델은 유용성은 높지만, 보안측면에서 취약하고, 보안성을 향상시킨 Gimpy나 Pix모델들은 유용성이 취약했다.

따라서 본 논문에서는 보안성은 더욱 강화되어 공격에 대해서는 안전하면서도 사용자가 이용하는 데에는 불편함이 없이 쉽게 인증 받을 수 있도록 하는 RTT모델을 제안하여, 쿠키정보를 이용하는 RTT기반의 향상된 프로토콜 인증방법에서 이 모델을 사용한다.

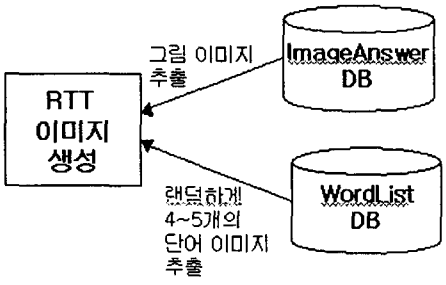
본 논문에서 제안하는 모델은 문자열로만 이루어진 이미지가 아닌 그림을 보여주는데, 유용성을 고려하여 그림과 함께 그림에 해당하는 단어와, 그림과 전혀 관련없는 단어들을 이미지내에 함께 삽입하여 그림에 대한 단어를 정확하게 입력할 수 있도록 힌트를 제공해준다. 또한 우리나라 환경에 적응될 수 있도록 문자열은 한글로 생성한다. 그림과 그에 해당하는 단어와 관련없는 단어들의 위치를 바꾸어가며 이미지를 만들기 때문에 자동화된 프로그램으로는 어떤 부분이 그림에 해당하는 위치인지 알아내기가 힘들고, 그림을 스캔한다 하더라도 내용을 알아맞추기란 힘들기 때문에 단어와 연결시키는 것은 불가능하다. 이는 보안성은 강화하면서 그림과 그에 해당하는 단어까지 함께 보여주므로 사용자 입장에서 유용성도 증가한다.



[그림 4] 제안한 새로운 모델의 예제

패스워드 인증방법은 쿠키정보를 이용하는 향상된 프로토콜 인증방법을 이용하고, 이 프로토콜에서 사용하는 RTT로 제안한 모델을 적용한다. 이 프로토콜은 기존의 인증 프로토콜에서 사용자에게 RTT를 요청할 것인지를 결정하는 부분과, RTT를 요청하여 처리하는 부분만을 추가적인 코드를 작성하고 서브루틴으로 삽입하여 구현하면 된다. 따라서 이식성이 높고 기존의 웹 브라우저에서 제공되므로 가용성 또한 좋다. 제안한 RTT 모델을 생성하는 절차는 [그림 5]에 나타나있다.

1. 그림과 해당하는 답을 이미지로 만들어 DB생성 (ImageAnswer)
2. 단어 이미지 리스트 DB생성(WordList)
3. RTT 요청이 들어오면, 쿠키정보를 이용하여
   (1) 캐쉬된 RTT가 있다면, 그 RTT를 보내준다.
   (2) 캐쉬된 RTT가 없다면, 새로운 이미지를 생성하여 RTT로 보내준다.
      a) ImageAnswer에서 그림 이미지를 추출
      b) WordList에서 랜덤하게 4~5개 단어 이미지를 추출
      c) 이미지들을 조합하여 새로운 RTT 이미지 생성



[그림 5] 제안한 RTT 모델 생성 절차

4. 결론 및 향후 연구과제

본 논문에서는 사용자가 쉽게 사용할 수 있는 기본적인 패스워드 인증방법에서 자동화된 프로그램으로 공격하는 글로벌 패스워드 공격을 막기 위한 RTT를 이용하는 향상된 프로토콜 인증방법을 사용한다. 이때 이용하는 RTT를 보안성과 유용성을 고려하여 새로운 모델을 제안하였다. 이는 자동화된 프로그램에게는 보안성을 더욱 강화하고 사용자에게는 유용성이 증가하는 모델이 된다.

사용자의 개인 정보를 보호하기 위하여 웹 기반 서비스들에서 사용자 계정은 인증을 통해서만 접근할 수 있도록 하고 있다. 무료 이메일 서비스를 제공하는 일반적인 사이트들에서는 자동으로 아이디를 생성하는 것을 막기 위하여 RTT에 응답하도록 하고 있는데, 개인 신상 정보뿐만 아니라 개인의 금융정보들이 이용되는 쇼핑몰과 같은 사이트들에서는 사용자 계정을 획득하는 것이 특히 중요하다. 따라서 사용자 계정을 획득하기 위한 글로벌 패스워드 공격이 불가능하도록 본 논문에서 제안한 RTT 모델 기반의 개선된 인증방법을 사용하여 타당성 검증을 위해 테스트용 웹사이트에서 실제 적용해본다.

참 고 문 헌

[1] Anne Adams and Martina Angela Sasse, "Users are not the enemy", Communications of the ACM, December 1999.

[2] Walter Belgers, "UNIX password security", CiteSeer, December 1993.

[3] A. Brümme, M. Kronberg, O.Ellenbeck and O.Kasch, "A conceptual framework for testing biometric algorithms within operating systems' authentication", Proceedings of the 2002 ACM symposium on Applied computing, March 2002.

[4] F. Monrose, M. Reiter and S. Wetzel, "Password hardening based on keystroke dynamics", Springer-Verlag, 2001.

[5] Moni Naor, "Verification of a human in the loop, or Identification via the Turing test", http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human_abs.html , 1996.

[6] K. Fu, E. Sit, K. Smith and N. Feamster, "Dos and Don'ts of Client Authentication on the Web", 10th USENIX Security Symposium, August 2001.

[7] Benny Pinkas and Tomas Sander, "Securing passwords against dictionary attacks", Proceedings of the 9th ACM conference on Computer and communications security, November 2002.

[8] The CAPTCHA Project. http://www.captcha.net/