Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.1, Feb. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.1.17

패스워드 매니저의 클라이언트-서버 통신 취약점 분석

홍 승 희,[1†] 소 재 우,[2‡] 정 혜 라[1]
[1,2]서강대학교 (대학원생, 교수)

Security Vulnerabilities of Client-Server Communications of Password Managers

Seunghui Hong,[1†] Jaewoo So,[2‡] Hyera Jeong[1]
[1,2]Sogang University (Graduate student, Professor)

요 약

웹사이트 로그인 정보 및 결제 정보들을 편리하게 관리하기 위해 패스워드 매니저를 이용하는 사용자가 증가하고 있다. 패스워드 매니저는 사용자의 웹사이트 로그인 정보 및 결제 정보들을 서버상에 암호화하여 저장하고, 사용자는 서버에 접속하여 패스워드 정보들을 수신하여 사용한다. 따라서, 공격자가 패스워드 매니저와 서버와의 통신 메시지를 스니핑하여 메시지 내용을 해독할 수 있거나, 또는 공격자가 사용자의 메모리 정보를 탈취하여 메시지 내용을 해독할 수 있으면, 사용자의 모든 패스워드 정보가 노출되는 심각한 문제가 발생한다. 본 논문에서는 주요 패스워드 매니저들의 클라이언트-서버 통신 및 암호 방법을 분석하고, 클라이언트-서버 통신에 심각한 취약점이 있음을 보인다.

ABSTRACT

Many users are using password managers in order to conveniently manage several usernames and passwords needed to access the web sites. The password manager encrypts and stores several passwords on the server, and the user accesses the server to receive the password information. Thus, if an attacker can sniff a message between the password manager and the server and decrypt the message content, or if an attacker can steal the computer's memory and decrypt the message content, then all the passwords will be exposed to the attacker. In this paper, we analyze the client-server communications and encryption process of password mangers and show there is a serious vulnerability in memory attack.

Keywords: Password Manager, Client-Server Communications, Password Vault, Memory Attack, Password Decryption

I. 서 론

사용자의 아이디와 패스워드를 확인하여 웹서비스를 제공하는 웹사이트들이 증가하면서 여러 개의 패스워드를 사용하는 사용자가 많아지고 있다. 이에 웹사이트 로그인 정보들을 효율적으로 관리해주는 패스워드 매니저를 이용하는 사용자들이 증가하고 있다[1, 2]. 2019년 패스워드 매니저 시장 규모는 10억 5천만 달러였으며, 2020년에서 2025년까지 연평균 19% 이상 성장하여 2025년에 298억달러에 이를 것으로 예측되고 있다[2, 3].

패스워드 매니저는 사용자의 비밀 정보(일례로 웹사이트별 아이디와 패스워드, 결제 정보 등)를 암호화하여 데이터베이스에 저장하고, 사용자는 하나의 마스터 패스워드만을 사용하여 암호화된 데이터베이스에 접근한다. 따라서 사용자는 웹사이트별 아이디와 패스워드를 기억할 필요 없이 마스터 패스워드만 기억하면 된다. 그러나 사용자의 비밀 정보가 들어있는 데이터베이스를 로컬 컴퓨터에 저장하는 경우 패스워드 저장소의 위치, 파일 포맷, 암호화 방식 등

Received(10. 23. 2019), Modified(01. 08. 2020), Accepted(01. 09. 2020)
† 주저자, shong1201@sogang.ac.kr
‡ 교신저자, jwso@sogang.ac.kr(Corresponding author)

이 알려지면서 패스워드 저장소의 보안 취약점이 심각히 드러났다[4, 5]. 따라서 많은 패스워드 매니저들은 사용자의 비밀 정보가 들어 있는 데이터베이스를 인터넷 서버에 암호화하여 저장하고 필요할 때마다 서버에 접속하여 패스워드 정보들을 수신하여 사용한다. 그러나 공격자가 패스워드 매니저와 서버 간의 통신 메시지를 스니핑하여 해독하는 경우 사용자의 비밀 정보가 공격자에게 노출되는 심각한 문제가 발생할 수 있다.

본 논문은 패스워드 매니저와 서버 간의 통신 및 암호화 과정을 분석하여 보안 취약점이 있음을 밝히고, 공격 시험을 통해 사용자의 웹사이트별 로그인 정보를 복호화할 수 있음을 보인다. 본 논문의 구성은 다음과 같다. 2장에서는 패스워드 매니저의 보안 취약점에 대한 기존 연구를 소개한다. 3장에서는 대표적인 상용 패스워드 매니저 프로그램인 LastPass의 클라이언트-서버 통신 취약점을 분석하고, 4장에서는 대표적인 오픈 소스 프로그램인 KeePass의 클라이언트-서버 통신 취약점을 분석한다. 그리고 5장에서 결론을 맺는다.

II. 관련 연구

패스워드 매니저의 보안 취약점에 대한 많은 연구가 진행되었다. 먼저, 패스워드 매니저가 암호화된 데이터베이스를 로컬 컴퓨터에 저장할 때 패스워드 저장소에 대한 보안 취약점이 보고되었다[4-7]. 논문 [4]는 패스워드 매니저 프로그램 11개를 대상으로 패스워드 저장소 위치와 파일 포맷을 분석하여 취약점을 분석하였다. 논문 [5]와 [6]은 패스워드 매니저의 저장소 취약점을 지적하고 공격 시나리오를 제시하였다. 논문 [7]는 패스워드 저장소 보안 강화를 위한 설계 지침을 제시하였다.

패스워드 매니저가 웹 브라우저의 로그인 정보를 자동으로 입력하면서 발생되는 보안 취약점이 지적되었다[8, 9], 논문 [8]은 iFrame 공격 또는 Sweep 공격을 통한 자동 로그인 취약점을 지적하였고, 논문 [9]은 북마크릿을 이용한 자동 로그인 취약점을 지적하였다. 또한 패스워드 매니저는 웹에서 동작하므로 CSRF(Cross-Site Request Forgery)와 XSS(Cross-Site Scripting) 취약점이 지적되었다[9, 10].

패스워드 매니저가 사용자의 비밀 정보가 들어 있는 데이터베이스를 인터넷 서버에 저장할 때, 패스워드 매니저는 서버에 접속하여 비밀 정보를 수신한다. 논문 [8]은 클라이언트-서버 간의 통신에 암호화되지 않은 HTTP 프로토콜을 사용할 때 스니핑 문제를 지적하였다. 또한 논문 [11]는 브라우저 기반 패스워드 매니저의 보안 취약점을 지적하고 새로운 클라우드 기반 패스워드 관리 방식을 제안하였다.

상용 패스워드 매니저들중에 하나인 LastPass 프로그램에 대한 보안 취약점 연구가 진행되었다. 논문 [4-6]는 로컬 컴퓨터에 저장된 LastPass 패스워드 저장소의 보안 취약점을 분석하였고, 논문 [8, 9]은 웹 브라우저와 연동할 때 발생하는 보안 취약점을 다루었다. 그리고 M. Vigo and A. Garcia는 Black Hat Europe 2015에서 LastPass 프로그램의 저장소 취약점과 클라이언트-서버 간의 통신 메시지를 분석하였다[12].

대표적인 오픈 소스 패스워드 매니저인 KeePass 프로그램에 대한 보안 취약점 연구가 진행되었다. 논문 [13]은 컴퓨터 메모리가 부족할 때 하드 디스크에 마스터 패스워드를 암호화하지 않고 저장하는 문제, 데이터베이스를 출력할 때 마스터 패스워드를 재확인하지 않는 문제, 암호화하지 않은 데이터베이스를 임시 폴더에 저장하는 문제를 지적하였다. 논문 [14]는 KeePass 데이터베이스 구조를 분석하고 전수 조사 방식으로 마스터 패스워드를 찾는 복잡도를 분석하였다.

본 논문은 패스워드 매니저의 클라이언트-서버 통신 취약점을 Fig. 1과 같이 사용자 컴퓨터에서 동작하는 패스워드 매니저와 사용자 비밀 정보 데이터베이스가 저장된 서버와의 통신 과정을 다룬다. 본 논문의 공헌은 다음과 같다. 첫째, 상용 패스워드 매니저 프로그램인 LastPass의 클라이언트-서버 통신 과정을 메시지 별로 상세히 분석하고, 메시지 내용이 암호화되지 않은 상태로 메모리에 저장되어 메모리 공격에 취약하다는 것을 보였다. 그리고 데이터베이스를 파싱하고 복호화하는 구체적인 과정을 보였다. 기존 논문 [4-6, 12]은 로컬 컴퓨터에 저장된 패스워드 데이터베이스 파일을 분석하는데 그쳤고, 논문 [12]은 클라이언트-서버 간의 통신 메시지를 분석하는데 그쳤지만, 본 논문은 메모리 공격을 통해 통신 메시지 내용을 탈취하여 LastPass의 비밀번호가 저장된 데이터베이스를 복호화할 수 있다는 것을 보였다. 둘째, 대표적인 오픈 소스 프로그램인 KeePass의 클라이언트-서버 통신 과정을 분석하고, 메시지 스니핑과 메모리 공격에 모두 취약하다는 것

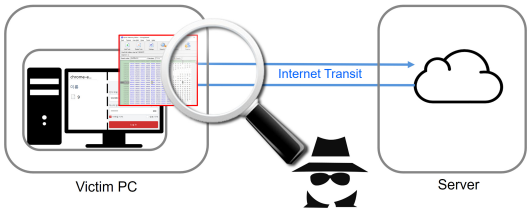


Fig. 1. Security vulnerabilities of client-server communications of a password manager

을 보였다. 기존의 KeePass 보안 취약점 연구는 논문 [13, 14]에서와 같이 비밀번호가 저장된 로컬 데이터베이스 파일의 포렌식 분석을 기반으로 하는데 비해, 본 논문은 웹 브라우저와 KeePass 프로그램 간의 클라이언트-서버 통신에 메시지 스니핑 및 메모리 공격 취약점이 있음을 보였다.

III. LastPass 클라이언트-서버 통신 취약점

3.1 LastPass 개요

대표적인 상용 패스워드 매니저 프로그램들은 LastPass, Dashlane, Keeper Security, 1Password, Roboform 등이 있다[15-17]. 이 중에서 널리 사용하고 있는 상용 패스워드 매니저 프로그램들은 Lastpass, Keeper Security, Dashlane 이며[18, 19], 그 중에서 LastPass 프로그램은 61,000개의 기업에서 사용하고 있고, 2020년 1월 기준 크롬 웹스토어에서 가장 많이 다운로드된 패스워드 매니저이며, 구글 플레이에서 가장 많은 리뷰 숫자를 보인 프로그램이다[19-21]. 본 논문에서는 LastPass 패스워드 매니저의 서버와의 통신 취약점을 분석한다. 2019년 12월 기준으로 국내 데스크탑 환경에서 웹브라우저 점유율은 크롬 브라우저가 70.35%, 인터넷 익스플로러가 15.63%이다[22]. 크롬 브라우저 또는 인터넷 익스플로러 사용자가 85.98%이므로 본 논문에서는 윈도우 데스크탑 환경에서 크롬 브라우저 또는 인터넷 익스플로러를 사용하는 환경을 고려한다. 그러나 파이어폭스 브라우저에서도 동일한 보안 취약점이 나타난다.

LastPass는 Fig. 2와 같은 로그인 화면에서 "암호 기억" 옵션에 따라 사용자의 비밀 정보 데이터베이스 저장 위치를 달리한다[23]. 옵션을 선택하면 사용자 로컬 컴퓨터에 저장하고, 옵션을 선택하지 않으면 서버에 저장한다. 본 논문에서는 옵션을 선택하지 않는 것으로 가정한다.

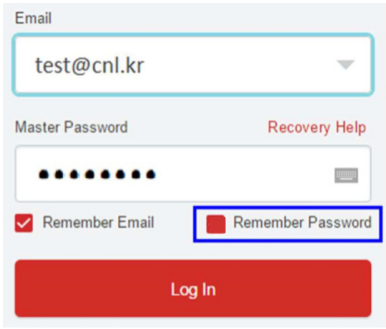


Fig. 2. Login of LastPass

LastPass의 동작은 Fig. 3과 같다. 사용자가 로그인 과정을 수행하면, 서버로부터 암호화된 사용자의 비밀 정보 데이터베이스(vault)를 수신받아 로컬 컴퓨터에 저장한다. 그리고 응답 메시지로 암호화된 데이터베이스를 복호화할 수 있는 정보를 수신한다.

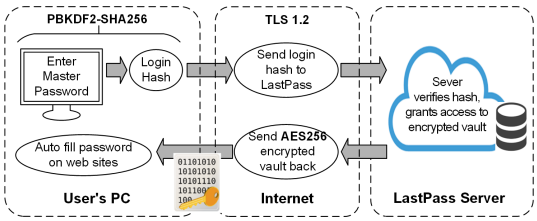


Fig. 3. Overall operation of LastPass

3.2 LastPass 클라이언트-서버 송수신 과정 및 복호화 분석

Fig. 4는 LastPass와 서버 간의 송수신 과정을 보인다. 사용자 컴퓨터에 설치된 LastPass는 서버로 보내는 요청 메시지에 사용자의 아이디와 마스터

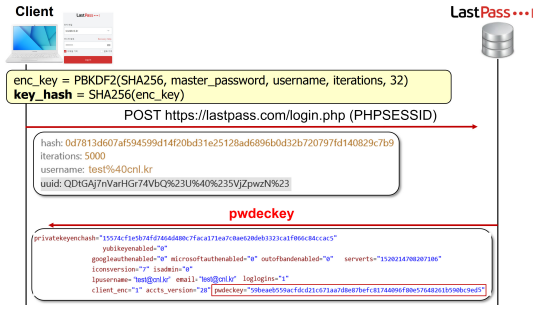


Fig. 4. Client-server communications of LastPass

패스워드의 해쉬값을 포함하여 전송한다. 서버는 클라이언트가 보낸 요청 메시지의 아이디와 패스워드를 확인한 후, 클라이언트에 보내는 응답 메시지에 사용자의 비밀 정보 데이터베이스를 복호화할 수 있는 "pwdeckey"를 포함하여 전송한다.

LastPass 클라이언트 프로그램은 로그인이 완료되면, 서버로부터 사용자 비밀 정보 데이터베이스를 수신하여 로컬 컴퓨터에 저장한다. 데이터베이스가 저장되는 파일 경로는 브라우저에 따라 다르며 크롬, 인터넷 익스플로러, 파이어폭스 브라우저는 Table 1에 기술한 바와 같다. Table 1의 경로에 데이터베이스 파일명은 확장자가 없는 0~9까지의 숫자이다. 숫자는 LastPass 클라이언트 프로그램을 설치할 때마다 변경된다. 데이터베이스 파일 포맷은 SQLite 데이터베이스 형식이다.

사용자의 비밀 정보 데이터베이스를 "DB Browser for SQLite" 프로그램[24]을 사용하여 열면 Fig. 5와 같다. 데이터베이스 type이 "key"에 해당하는 data가 "encryptedVaultKey"이며, 이는 암호화된 vault를 복호화할 수 있는 키가 암호화된 것이다. 데이터베이스 type이 "accts"에 해당하는 data가 "encryptedVault"로, 여기에 사용자의 웹사이트 로그인 정보 및 패스워드들이 암호화되어 있다.

암호화된 데이터베이스의 복호화 과정은 Fig. 6과 같다. LastPass는 서버로부터 수신한 응답 메시지에 있는 pwdeckey의 해쉬값을 AES256-CBC 암호키로 하여 encryptedVaultKey를 복호화한다. 그리고 복호화된 정보를 Vault Key로하여 로컬 컴퓨터에 저장된 암호화된 비밀 정보 데이터베이스 encryptedVault를 복호화한다.

Table 1. Location and format of encrypted vault

Location	• Chrome Browser 〈user profile dir〉 \AppData\Local\Google\Chrome\ User Data\Default\ databases\chromeextension_hdoki ejnpimakedhajhdlcegeplioahd_0 • IE Browser 〈user profile dir〉\AppData\ LocalLow\LastPass • FireFox Browser 〈user profile dir〉\AppData\Local Settings\Applicaion Data\LastPass
Format	SQLite format 3

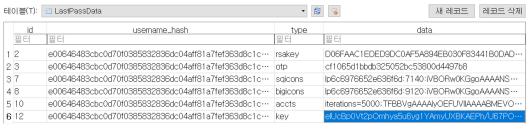


Fig. 5. Format of LastPass encrypted valut

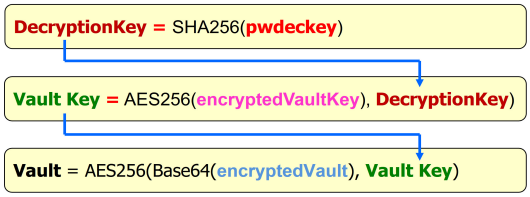


Fig. 6. Vault decryption process of LastPass

3.3 LastPass 클라이언트-서버 통신 취약점

LastPass의 암호화된 데이터베이스를 복호화하기 위해서는 Fig. 4에서 서버가 LastPass 클라이언트로 보내는 메시지에 포함되어 있는 pwdeckey 값을 추출해야 한다. 그러나 LastPass 클라이언트-서버 통신은 TLS 기반으로 데이터를 암호화하여 송수신하므로 메시지를 스니핑하더라도 pwdeckey 값을 추출하기 어렵다. 그러나 LastPass 클라이언트 프로그램은 메시지 송수신 내용을 로컬 컴퓨터 메모리에 남겨두는 심각한 취약점이 있다. 공격자는 pwdeckey 값을 탈취하기 위해 사용자가 Fig. 4의 LastPass 로그인 과정을 종료하였을 때 메모리 공격을 통해 사용자 컴퓨터의 메모리 헥사 데이터를 탈취한다. 본 논문에서는 LastPass 로그인 과정이 종료된 후에 "Quick Memory Editor" 프로그램[25]를 이용하여 사용자 컴퓨터의 메모리를 분석하였으며, Fig. 7과 같이 LastPass 서버로부터 수신한 응답 메시지가 메모리에 저장되어 있음을 확인하였다. Fig. 4의 로그인 과정에서 보듯이 사용자가 LastPass 서버로 보내는 로그인 메시지는 사용자 아이디와 loglogins="0" 값을 포함한다. 이후 로그인 메시지를 수신한 서버는 LastPass 클라이언트에 pwdeckey를 포함하는 응답 메시지를 전송한다. 따라서, "Quick Memory Editor" 프로그램에서 사용자 로그인 아이디와 loglogins="0"을 키워드로 메시지를 검색한 후 다음 메시지를 열면 Fig. 7의

우측과 같이 로그인 메시지에 대한 응답 메시지의 헥사 데이터를 확인할 수 있다. 서버로부터 수신한 응답 메시지를 바이트 단위로 분석하면, ASCII 값으로 "pwdeckey="를 발견할 수 있고, '=' 뒤의 비트 값이 Fig. 6의 encryptedVault 복호화 과정에 사용되는 pwdeckey 값이다. 이후 pwdeckey를 사용하여 Fig. 6의 방법으로 Table 1에 저장된 사용자의 비밀 정보 데이터베이스를 복호화할 수 있다.

3.4 LastPass 클라이언트-서버 통신 취약점 시험

3.4.1 시험 시나리오

시험 시나리오에서 LastPass 사용자는 Table 2와 같이 아이디를 "test", 패스워드를 "test1234*"로 하여 해당 URL에 접속한다.

LastPass의 클라이언트-서버 통신 취약점 공격 시나리오를 Fig. 8에 도시하였다. 공격자는 피공격자 컴퓨터에 메모리 공격을 하여 메모리상에 남아 있는 pwdeckey를 탈취한다. 메모리 공격 프로그램은 [26]과 같이 오픈 소스 프로그램을 참조하여 개발할

Table 2. Login Information of the LastPass vulnerability test

Item	Value
URL	http://c**.******.ac.kr/mycloud/
Username	test
Password	test1234*

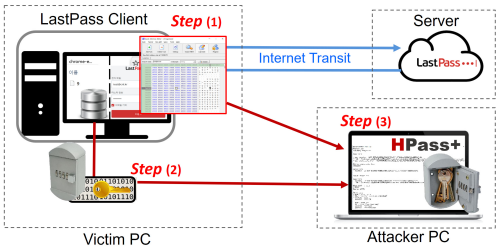


Fig. 8. Attack scenario of LastPass

수 있다. 그리고 Table 1과 같이 피공격자 컴퓨터의 특정 위치에 저장되어 있는 데이터베이스 파일을 탈취한다. 그리고 본 연구를 위해 개발한 HPassPlus 프로그램을 이용하여 피공격자의 암호화된 데이터베이스를 복호화한다.

3.4.2 암호화된 데이터베이스 복호화

사용자의 비밀 정보 데이터는 Fig. 9와 같이 데이터베이스 type이 "accts"에 해당하는 data에 저장된다. accts 데이터는 반복 횟수 정보 iteration=x;로 시작하며, 이후 사용자 비밀 정보를 AES256으로 암호화한 것을 Base64 인코딩한 것이다.

Fig. 10은 accts 데이터를 Base64 디코딩한 것이다. HEX 데이터에서 ASCII 값 "ACCT" (0x41434354) 이후에 "!!" (0x2121)가 나오는 곳에 각각 ① name, ② URL, ③ username, ④ password 정보가 담겨 있다. "name" 값은 사용자가 사이트별 로그인 정보를 저장할 때 지정하는 이름

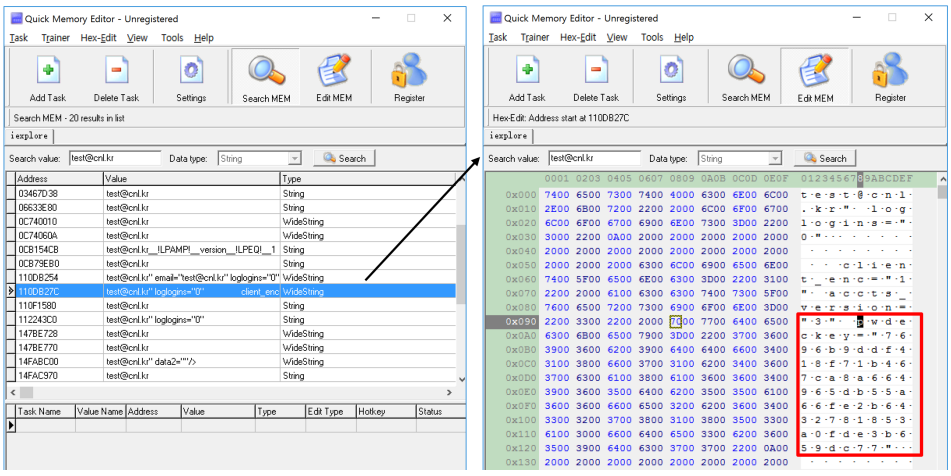


Fig. 7. Memory after a LastPass login

Fig. 9. LastPass 데이터베이스의 "accts" 데이터

으로 설정하지 않으면 사이트의 도메인으로 지정된다. URL 정보는 암호화되어 저장되어 있지 않으므로 HEX 데이터를 ASCII 문자로 변환하여 얻을 수 있다. name, username, password는 각각 AES256-CBC로 암호화되어 있으며, 길이는 16 바이트 IV 값과 16 바이트 암호문으로 구성된다. 따라서 password 복호화를 일례로 들면 Fig. 11과 같이 32 바이트 데이터에서 앞 16 바이트를 IV로, 뒤 16 바이트를 암호문으로 하고, Vault Key는 Fig. 6에서 얻은 값을 사용하면 로그인에 사용한 패스워드를 복호화할 수 있다. 동일한 방법으로 name, username을 복호화한다.

본 논문에서는 accts 데이터로부터 "name, URL, username, password" 정보를 복호화하는 프로그램을 C# 언어로 개발하였으며, 이를 HPassPlus라고 지칭한다. Fig. 12는 개발한 프로그램을 사용하여 사용자의 비밀 정보를 복호한 결과이다.

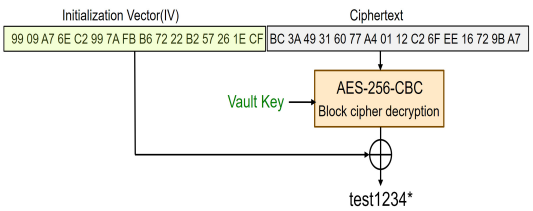


Fig. 11. Decryption of an encrypted password

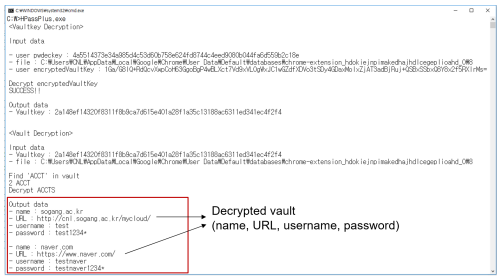


Fig. 12. Execution result of HPassPlus

IV. KeePass 클라이언트-서버 통신 취약점

4.1 KeePass 개요

KeePass 패스워드 매니저는 2003년에 배포된 대표적인 오픈 소스 프로그램이며 여러 기관에서 상을 받았다[27, 28]. 2019년 CSO 기사에 가장 인기 있는 6개 패스워드 매니저들 중에 KeePass가 포함되었다[16]. 그러나 KeePass 프로그램은 독립적으로 동작하는 로컬 프로그램으로 웹 브라우저와 연동되지 않는다. 따라서, 사용자는 로컬 컴퓨터에서 KeePass 프로그램을 실행한 후, KeePass 프로그램에서 웹사이트 주소를 키워드로 검색하여 사용자의 아이디와 패스워드를 복사하여 웹 브라우저의 해당 입력란에 각각 붙여넣어야 하는 불편함이 있다. 이러한 불편함을 해소하기 위해 사용자는 KeePass 프로그램에 KeePassHttp 서버 플러그인을 설치하고[29], 웹 브라우저에 확장 프로그램을 설치하여 사용한다.

KeePass 프로그램과 웹 브라우저 간은 Fig. 13과 같이 HTTP 통신으로 데이터를 주고 받는다. 크롬 또는 파이어폭스 브라우저를 사용할 때, 확장 프로그램은 ChromeIPass와 KeePassHttp-Connector가 있다[30, 31].

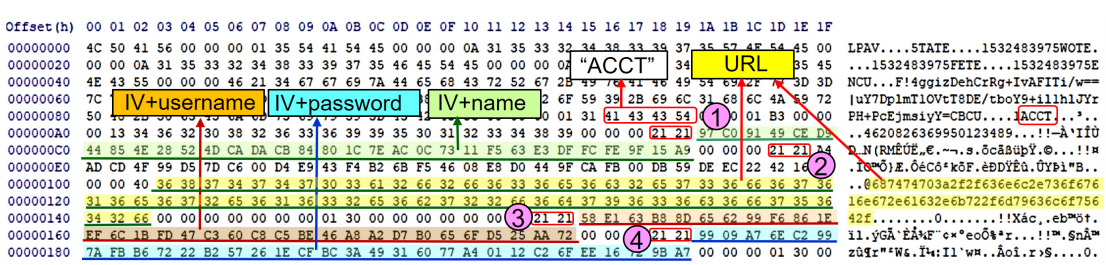


Fig. 10. Encrypted vault of LastPass

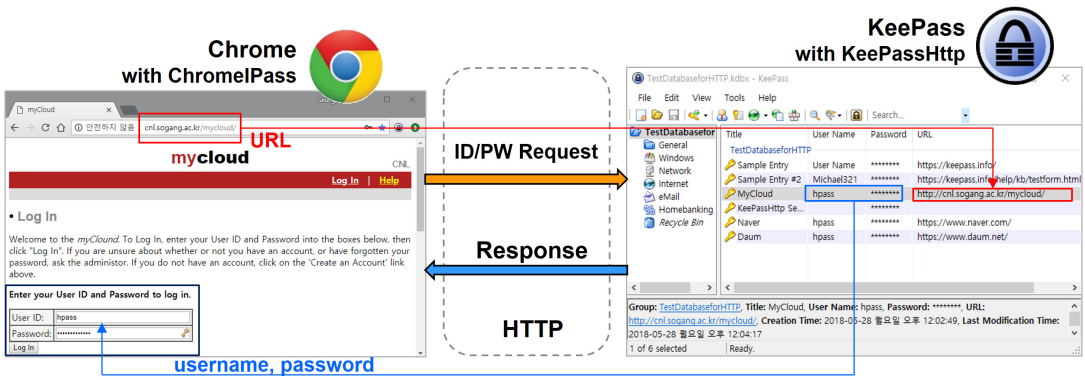


Fig. 13. Overall operation of ChromeIPass and KeePassHttp

KeePassHttp-Connector는 ChromIPass를 기반으로 개발된 프로그램으로 동작은 ChromeIPass와 동일하다. 본 논문에서는 시장 점유율이 높은 크롬 브라우저를 기준으로 확장 프로그램은 ChromeIPass을 사용하는 환경을 고려한다.

사용자가 웹사이트를 방문하면 ChromeIPass 클라이언트는 로그인 정보를 HTTP 프로토콜을 사용하여 KeePassHttp 서버에 요청한다. KeePassHttp 서버는 KeePass 패스워드 매니저가 저장한 로그인 정보를 검색하여 응답 메시지로 전송한다.

4.2 KeePass 클라이언트-서버 송수신 과정 및 복호화 분석

크롬 브라우저 플러그인으로 설치된 ChromeIPass 클라이언트와 KeePassHttp 서버 간의 통신은 연결 수립 과정과 로그인 정보를 전달하는 과정으로 구분된다. 먼저, Fig. 14는 연결 수립 과정을 보여준다. ChromeIPass는 서버로 보내는 요청 메시지에 연결 수립을 나타내는 "RequestType":"associate"와 "Key" 값을 포함한다. 이때, "Key" 값은 추후 암호문을 해독하는 암호키로 사용된다. 서버는 클라이언트에 보내는 응답 메시지에 클라이언트의 연결을 구별할 수 있도록 "Key"에 대응하는 "Id"를 할당한다.

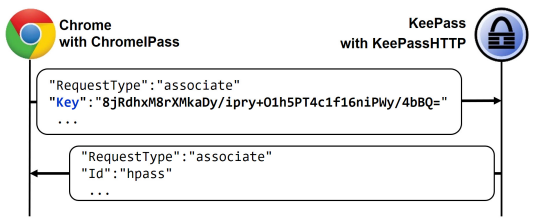


Fig. 14. Client-server association of KeePass

연결이 수립된 이후, 로그인 정보를 전달하는 과정은 Fig. 15와 같다. 사용자가 로그인 정보를 요구하는 웹사이트를 방문하면 ChromeIPass는 서버로 보내는 요청 메시지에 〈Nonce, 암호화된 URL〉을 포함한다. 여기서, URL 암호화 알고리듬은 AES256-CBC가 사용된다. 이때 IV는 함께 전송되는 Nonce 값이 사용되고, 암호키는 접속 과정에서 전송한 Key가 사용된다. 서버는 클라이언트가 보낸 암호화된 URL을 복호화한 후, 클라이언트에 보내는 응답 메시지에 〈None, Entries〉을 포함하여 전송한다. 여기서, Entries는 암호화된 로그인 아이디와 패스워드 정보를 포함하고 있다. 암호화 알고리듬은 AES256-CBC가 사용된다. 이때 IV는 함께 전송되는 Nonce 값이 사용되고, 암호키는 접속 과정에서 전송된 Key가 사용된다.

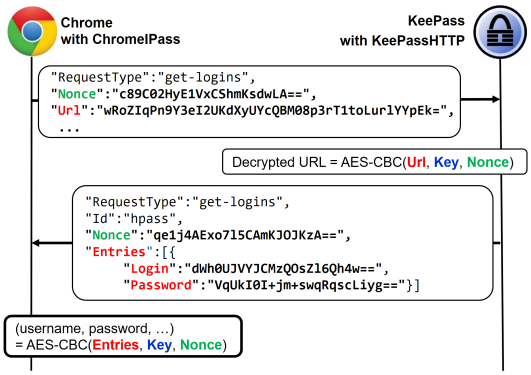


Fig. 15. Client-server association of KeePass

4.3 KeePass 클라이언-서버 통신 취약점

사용자의 암호화된 로그인 정보를 복호화하기 위해서는 연결 수립 과정에서 클라이언트가 서버로 전송하는 “Key” 값과 사용자가 로그인 정보를 요구하는 웹사이트를 방문하였을 때 서버가 클라이언트로 전송하는 〈Nonce, Entries〉 데이터가 필요하다. 그런데 클라이언트-서버 통신 과정에 암호화되지 않은 HTTP 프로토콜을 사용하기 때문에 공격자는 메시지 스니핑 공격을 통해 해당 정보들을 모두 탈취할 수 있다. 또한 메모리 상에 해당 내용을 남겨두기 때문에 메모리 공격으로도 해당 정보들을 모두 탈취할 수 있는 심각한 취약점이 있다.

4.4 KeePass 클라이언트-서버 통신 취약점 시험

4.4.1 시험 시나리오

시험 시나리오에서 KeePass 사용자는 Table 2의 정보를 사용한다. KeePass의 클라이언트-서버 통신 취약점 공격 방법은 2가지이다. 첫 번째 공격 방법은 HTTP 메시지 스니핑 공격이다. KeePass 서버가 원격에 위치하면서, 사용자가 암호화하지 않은 Wi-Fi를 이용하는 경우 무선 스니핑을 통해 HTTP 메시지를 탈취할 수 있다. 일례로 Fig. 16은 HTTP 스니핑을 통해 연결 수립 과정에서 “Key”를 탈취한 화면이다.

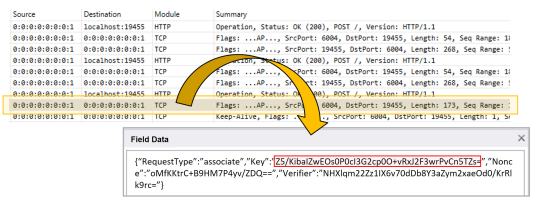


Fig. 16. HTTP message sniffing

두 번째 공격 방법은 컴퓨터 메모리 공격이다. 메모리 공격으로 공격자는 메모리에 남겨진 HTTP 메시지 내용을 탈취할 수 있다. 본 논문에서는 메모리 공격으로 사용자의 비밀 정보를 복호화하는 과정을 보인다.

4.4.2 메모리 공격

ChromeIPass와 KeePassHttp 서버 간 연결 수립 후 메모리 내용을 “HxD” 프로그램[32]을 사용하여 공격하면 Fig. 17과 같이 메모리에 “Key” 값이 남아 있음을 확인할 수 있다. 또한 사용자가 로그인 정보를 요구하는 웹사이트 방문후 메모리 공격을 취하면 Fig. 18과 같이 메모리에 〈Nonce, Entries〉 데이터를 확인할 수 있다. 해당 데이터들은 문자열 시작과 끝에 큰 따옴표(0x22)를 기준으로 파싱할 수 있다. 메모리 있는 데이터들을 Base64 디코딩 후에 AES256-CBC 복호화를 수행한다. 따라서 password 복호화를 일례로 들면 Fig. 19와 같다.

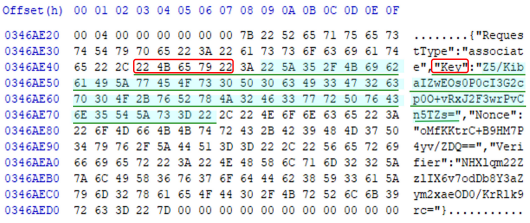


Fig. 17. Key in the memory

V. 결 론

본 논문에서는 널리 사용하고 있는 상용 패스워드 매니저인 LastPass와 대표적인 오픈 소스 패스워드 매니저인 KeePass의 클라이언트-서버 통신 취약점을 분석하였다. LastPass 프로그램은 서버와의

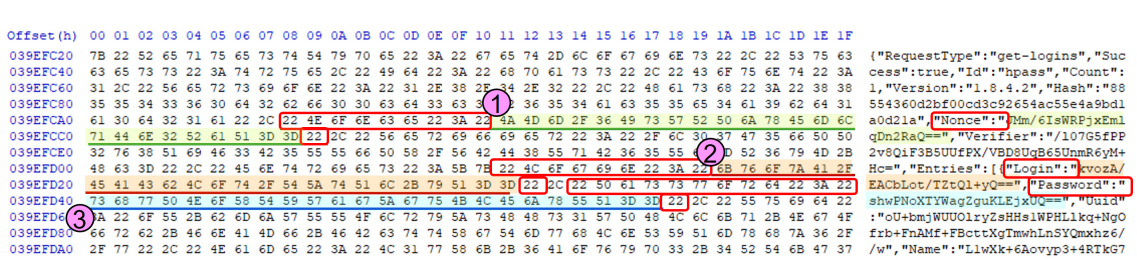


Fig. 18. Encrypted entries in the memory

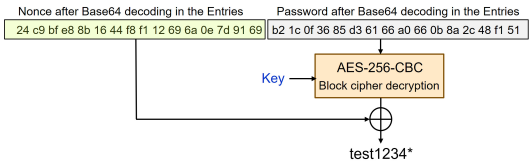


Fig. 19. Decryption of a password after Base64 decoding

통신에서 암호화된 HTTPS 프로토콜을 사용함으로써 인터넷 스니핑 공격을 방어하였지만, 컴퓨터 메모리에 메시지 내용을 남겨둠으로써 메모리 공격에 취약하였다. 공격 시나리오를 제시하고 공격 시험을 수행하였다. 메모리 공격을 통해 사용자 비밀 정보 데이터베이스를 복호화하는데 필요한 암호키를 탈취하였으며, 사용자 비밀 정보 데이터베이스 파일의 데이터 파싱을 통해 웹사이트 로그인 패스워드를 복호화할 수 있었다.

많은 KeePass 이용자들은 편의를 위해 ChromeIPass와 KeePassHttp 서버를 함께 사용한다. 그러나 ChromeIPass와 KeePassHttp 서버간의 통신은 암호화되지 않는 HTTP 프로토콜을 사용함으로써 인터넷 스니핑 공격에 취약하다. 또한 컴퓨터 메모리에 메시지 내용을 남겨둠으로써 메모리 공격에도 취약하다. 본 논문에서는 메모리 공격을 통해 웹사이트 로그인 패스워드를 복호화할 수 있음을 보였다.

본 논문은 널리 사용하고 있는 상용 패스워드 매니저와 오픈 소스 패스워드 매니저 한 개씩을 대상으로 취약점을 분석하였지만, 해당 취약점은 다른 패스워드 매니저들에도 공통적으로 적용될 수 있는 일반적인 것이다. 본 논문에서 확인한 패스워드 매니저의 클라이언트-서버 통신 취약점을 해결하기 위해서는 첫째, 통신 메시지 내용이 공격자에게 노출되지 않도록 스니핑 및 중간자 공격에 안전한 전송 계층 보안 프로토콜을 사용하여야 하며, 둘째, 통신 메시지 내용을 메모리에 저장할 때는 암호화하여 저장하거나 메모리 내용을 즉각 삭제하는 등 메모리 공격에 안전하여야 한다. 본 연구 결과는 향후 패스워드 매니저의 취약점을 보완하고 설계하는데 도움이 될 것으로 판단된다.

References

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in Proc. IEEE Symposium on Security and Privacy, pp. 553-567, May 2012.

[2] ResearchAndMarcket, "Global password management market to 2024," https://www.researchandmarkets.com/reports/4773624, Jun. 2019.

[3] ResearchAndMarcket, "Password management market - Growth, trends, and forecast (2020-2025)," https://www.mordorintelligence.com/industry-reports/password-management-market, Jan. 2020.

[4] P. Gasti and K. B. Rasmussen, "On the security of password manager database formats," in Proc. European Symposium on Research in Computer Security, pp. 770-787, Sep. 2012.

[5] S. Kim and H. Kim, "Security analysis of password managers," Review of Korea Institute of Information Security and Cryptology, vol. 28, no. 1, pp. 36-42, Feb. 2018.

[6] H. Jeong and J. So, "Security of password vaults of password managers," Journal of the Korea Institute of Information Security and Cryptology, vol. 28, no. 5, pp. 1047-1057, Oct. 2018.

[7] M. Golla, B. Beuscher, and M. Dürmuth, "On the security of cracking-resistant password vaults," in Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1230-1241, Oct. 2016.

[8] D. Silver, S. Jana, D. Boneh, E. Chen, and C. Jackson, "Password managers: Attacks and defenses," in Proc. USENIX Security Symposium, pp. 449-464, Aug. 2014.

[9] Z. Li, W. He, D. Akhawe, and D. Song, "The emperor's new password mana

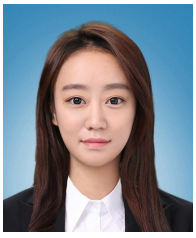
ger: Security analysis of web-based password managers," in Proc. USENIX Security Symposium, pp. 465-479, Aug. 2014.
[10] X. Li and Y. Xue, "A survey on server-side approaches to securing web applications," ACM Computing Surveys, vol. 46, no. 4, pp. 1-29, Apr. 2014.
[11] R. Zhao and C. Yue, "All your browser-saved passwords could belong to us: A security analysis and a cloud-based new design," in Proc. ACM conference on Data and Application Security and Privacy, pp. 333-340, Feb. 2013.
[12] M. Vigo and A. Garcia, "Even the LastPass will be gone, deal with it," Black Hat Europe 2015.
[13] J. Gray, V. N. L, Franqueira, and Y. Yu, "Forensically-sound analysis of security risks of using local password managers," in Proc. International Requirements Engineering Conference Workshops, pp. 1-8, Sep. 2016.
[14] H. Zhang, J. Hong, and J. Hu, "Analysis of encryption mechanism in KeePass Password Safe 2.30," in Proc. IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 43-46, Sep. 2016.
[15] Asecurelife, "Best password manager for stroing, secure passwords," https://www.asecurelife.com/best-password-manager/, Dec. 2019.
[16] T. Ferrill, "The 6 best password managers," CSO news, https://www.csoonline.com/article/3198507/the-6-best-password-managers.html, Jul, 2019.
[17] N. J. Rubenking, "The best password managers for 2020," PC Reviews, https://www.pcmag.com/roundup/300318/the-best-password-managers, Dec. 2019.
[18] Tom's guide, "Best password managers 2020," https://www.tomsguide.com/us/best-password-managers,review-3785.html, Dec. 2019.
[19] Google Play, "LastPass password manager," https://play.google.com/store/apps/details?id=com.lastpass.lpandroid&hl=ko, Jan. 2020.
[20] LastPass, "LastPass Homepage," https://www.lastpass.com/, Jan. 2020.
[21] Chrome Web store, "LastPass: Free password manager," https://chrome.google.com/webstore/detail/lastpass-free-password-ma/hdokiejnpimakedhajhdlcegeplioahd?hl=ko, Jan. 2020.
[22] Statcounter, "Desktop browser market share in republic of korea - December 2019," https://gs.statcounter.com/browser-market-share/desktop/south-korea, Jan. 2020
[23] LastPass, "Technical whitepaper," http://enterprise.lastpass.com, pp. 1-20, Mar. 2018.
[24] SQLite, "DB browser for SQLite," http://sqlitebrowser.org, Mar. 2018.
[25] Softcows, "Quick memory editor," http://softcows.com, Mar. 2018.
[26] G. McDonald, "Proecess dump," GitHub, https://github.com/glmcdona/Process-Dump, Apr. 2019.
[27] KeePass Password Safe, "KeePass password safe," https://keepass.info, Apr. 2018.
[28] KeePass, "Awards/ratings - Keepass," http://keepass.info/ratings.html, Sep, 2015.
[29] KeePassHttp, "KeePass plugin to expose password entries securely over HTTP," https://github.com/pfn/keepasshttp/, May 2017.
[30] PassIFox and chromeIPass, "Extensions to allow Chrome and Firefox," https://github.com/pfn/passifox, Feb. 2018.
[31] KeePassHttp-Connector, "Extension to allow Chrome and Firefox," https://git

hub.com/smorks/keepasshttp-connecto
r, Aug. 2019.
[32] mh-nexus, "HxD - Freeware Hex edit
or and disk editor," https://mh-nexus.
de/en/hxd, May 2018.

〈저자소개〉

홍 승 희 (Seunghui Hong) 학생회원
2017년 8월: 서강대학교 전자공학과 졸업
2019년 8월: 서강대학교 전자공학과 석사
2019년 9월~현재: ㈜삼성전자, 네트워크사업부 연구원
〈관심분야〉 머신러닝, 5G 액세스네트워크, 무선 보안

소 재 우 (Jaewoo So) 정회원
1997년 2월: 연세대학교 전자공학과 졸업
1999년 2월: 한국과학기술원 전기 및 전자공학과 석사
2002년 8월: 한국과학기술원 전기 및 전자공학과 박사
2001년~2005년: ㈜아이피원, 수석연구원
2005년~2007년: ㈜삼성전자, 책임연구원
2007년~2008년: Stanford University, 전기공학과 박사후연구원
2014년~2015년: UIUC, ECE 방문교수
2014년~2017년: 한국연구재단 ICT·융합연구단 전문위원
2008년 9월~현재: 서강대학교 전자공학과 교수
〈관심분야〉 머신러닝, 5G/6G 커넥티비티, 무선 보안



정 혜 라 (Hyera Jeong) 학생회원
2017년 2월: 연세대학교 미래교육원 소프트웨어개발학과 졸업
2017년 9월~현재: 서강대학교 정보통신대학원 석사과정
〈관심분야〉 보안취약점 분석, 암호 분석, 머신러닝