이 일 구 교수 지도

석사학위 청구논문

패스워드 매니저의 취약점 분석 및 보안성 평가 프레임워크

2024

성신여자대학교 대학원

미래융합기술공학과

장 지 원

패스워드 매니저의 취약점 분석 및 보안성 평가 프레임워크

이 일 구 교수 지도

이 논문을 석사학위논문으로 제출함

2023년 11월

성신여자대학교 대학원
미래융합기술공학과
장 지 원

인 준 서

장지원의 석사학위 논문으로 인준함

2023년 11월

심사위원장 임 연 섭 (서명 또는 인)

심 사 위 원 김 성 민 (서명 또는 인)

심 사 위 원 이 일 구 (서명 또는 인)

성신여자대학교 대학원

논 문 개 요

초고속 정보통신기술의 발달로 인터넷 사용자들은 다양한 웹 서비스를 이용할 때 회원 가입하고 발급받은 수많은 계정을 관리하게 되었다. 복잡하고 다양한 계정들을 효율적으로 관리하기 위해 웹 서비스 사용자들은 '패스워드 매니저'를 사용하고 있다. 패스워드 매니저는 계정 정보를 복잡하게 생성하거나 사이트 별로 저장하고, 사용자 인증이 필요할 때 정보를 손쉽게 불러올 수 있어서 사용자가 패스워드를 관리할 때 보안성과 편의성을 제공하는 소프트웨어이다. 그러나 패스워드 매니저에는 사용자의 개인정보 침해를 초래하는 보안 취약성이 잠재되어 있으며, 웹 서비스와 웹 애플리케이션에 널리 활용되고 있지만 최신의 연구는 부족한 실정이다. 따라서 본 연구에서는 데스크톱 환경에서 사용되는 웹 기반 패스워드 매니저의 전반적인 기능과 보안 취약성을 분석하고 상용 웹 서비스 환경에서 계정 정보 탈취 공격이 가능함을 입증한다. 그리고 실험 결과에 근거하여 취약점을 평가할 수 있는 패스워드 매니저 보안 평가 프레임워크를 제안한다. 제안한 패스워드 매니저 보안 평가 프레임워크를 활용하면 상용 패스워드 매니저의 보안 취약점을 분석하고 적절한 대응 방안을 도출할 수 있다.

목 차

표 차 례

그림 차례

Ⅰ. 서 론

1. 연구 배경

다양한 서비스를 제공하는 웹 애플리케이션을 이용하기 위해 웹 서비스 사용자는 회원가입을 통해 계정을 생성하고, 각자의 계정 정보로 인증하여 서비스의 인가 권한을 얻어서 웹 서비스를 이용한다. 정보통신기술과 웹 서비스 응용 기술의 발달로 다양한 웹 서비스가 등장하여 사용자들은 수많은 계정을 가지게 되었으나, 인간의 인지적인 한계로 모든 정보를 기억하기 어렵기 때문에 계정을 안전하고 손쉽게 관리하기 위해 웹 서비스 사용자들은 '패스워드 매니저(Password Manager)'를 사용하고 있다. 패스워드 매니저란 웹 사이트를 이용할 때 필요한 계정 정보인 패스워드를 생성할 때 저장한 정보를 활용하여 효율적으로 패스워드를 관리할 수 있는 소프트웨어이다. 브라우저에 기본으로 내재된 형태의 패스워드 매니저가 아니라면 별도로 설치해야 하는 번거로움이 있지만, 패스워드 매니저의 이용자 수와 산업의 최근 성장률은 웹 서비스 이용자들이 점차 패스워드 매니저를 디지털 환경에서 필수적인 애플리케이션으로 활용하고 있음을 시사한다.

그러나 패스워드 매니저는 안전한 패스워드 사용 습관을 위한 소프트웨어이지만, 패스워드 매니저의 취약점으로 인해 사용자의 계정 정보와 개인 정보가 탈취될 수 있다. 실제로 가장 많은 사용자 수를 기록한 상용 패스워드 매니저인 'LastPass'는 2022년 해킹 사고가 발생한 바 있고, 이외에도 다양한 패스워드 매니저의 취약점 식별 번호(CVE, Common Vulnerabilities and Exposures)가 최근까지 꾸준히 발급되고 있다.

서비스 별로 계정 정보를 다르게 설정하는 것이 이상적이지만, 많은 웹 서

비스 사용자가 패스워드를 유사하거나 동일하게 사용하고 있어서 한 곳에서 탈취된 계정 정보는 크리덴셜 스터핑(Credential Stuffing) 기법으로 다수의 웹 사이트나 도메인으로 피해가 확장되고 있다. 즉, 탈취된 계정 정보의 횟수나 개수와 상관없이 계정 정보 유출의 파급력이 커서 패스워드 매니저의 보안성 강화 연구를 지속적으로 수행해야 한다. 웹 해킹 기법의 발달로 웹 서비스를 구성하는 요소들의 보안성 강화 조치가 꾸준히 이루어지고 있으나 패스워드 매니저의 보안성은 비교적 이전의 상황에 비해 크게 개선되지 않았다.

최근 패스워드 매니저의 보안 취약점 분석과 보안 대응 방안에 관한 연구는 전 세계적으로 다양하게 수행되고 있지만 다음과 같이 몇 가지 한계점이 있다. 국내에서 수행된 연구의 경우, 물리적인 기기가 확보된 상황에서 단일 패스워드 매니저의 취약점을 이용하여 저장된 데이터를 복호화하는 연구와 디지털 포렌식 수사를 위한 연구에 한정되어 있다. 국외의 연구는 패스워드 매니저의 사용자 편의성(usability)에 관한 연구가 주로 수행되고 있고, 보안 취약점 관련된 연구는 2010년대 중반에 활발하게 진행되어 최신의 연구가 부족하다. 보안성과 사용성은 트레이드 오프(trade-off) 관계에 있으므로 사용성과 보안성 요구 조건을 동시에 분석하고 최적화하는 연구가 요구된다.

그러므로 본 연구에서는 사용자 수가 많은 패스워드 매니저의 사용 환경에서 유발되는 보안 취약점 벡터를 도출하고, 상용 패스워드 매니저의 보안 취약점을 분석하고 실험 및 공격 시연을 통해 보안 사고의 가능성을 입증한다. 또한, 문헌 조사와 실험을 통해 다양한 데스크톱 환경의 패스워드 매니저에 보편적으로 적용할 수 있는 보안성 평가 프레임워크를 활용하여 평가한다.

본 연구의 주요 기여점은 다음과 같다.

- 패스워드 매니저 보안성 연구를 메타 분석하여 상용 패스워드 매니저의 취약점 발생 공격 표면을 도출하였다.

- 공격 표면에 따라 계정 탈취가 발생할 수 있는 연계 공격을 구현하여 취약점을 증명하였다.
- 다양한 패스워드 매니저에 공통적으로 적용할 수 있는 보안성 평가 프레임워크를 제안하고 대표적인 상용 패스워드의 보안성을 평가했다.
- 향후 패스워드 매니저 사용에 있어 취약성을 줄일 수 있는 보안 요구사항과 대응 방법을 도출하였다.

본 논문의 구성은 다음과 같다. Ⅱ장은 연구 대상의 배경지식과 이전 연구 동향을 설명한다. Ⅲ장에서는 패스워드 매니저의 취약점 분석을 위해 공격 표면(Attack vector)을 식별하고, 이어진 Ⅳ장에서 식별한 취약성이 실제 내재하는지 상용 패스워드 매니저를 대상으로 실험한다. Ⅴ장에서는 실험 결과를 바탕으로 취약성 평가 항목을 도출하여 사용자 수 상위 5개의 확장 프로그램형 패스워드 매니저와 브라우저의 기본 내장형 패스워드 매니저 3개를 대상으로 보안성 점수를 측정한다. 이후 Ⅵ장에서 연구를 통한 시사점을 기반으로 보안 요구사항을 도출하여 안전한 패스워드 매니저 아키텍처를 제안한다. 마지막 Ⅶ장으로 향후 연구를 제시하며 결론을 맺는다.

2. 연구 목적

패스워드 매니저의 공격 표면은 웹 서비스 사용 시나리오에 따라 다양하게 발생할 수 있으므로 패스워드 매니저의 동작 구조와 특성을 고려해 공격 표면을 파악하고, 적절한 보안 조치가 필요한 지점을 사용성과 조율해야 한다. 동시에 수많은 패스워드 매니저 애플리케이션이 있지만 그중 어떤 것이 현재 기준에서 보안성이 가장 두드러지거나 미흡한지 체계적으로 평가할 수 있는 프레임워크가 없다. 따라서 이기종의 패스워드 매니저 간의 공통분모를 도출하고 다양한 패스워드 매니저에 통용할 수 있는 일반화된 체계를 정립하여 패스워드 매니저의 보안성을 평가할 수 있는 프레임워크가 요구된다.

이러한 필요성에 따라, 본 연구에서는 웹 서비스의 패스워드 매니저와 관련한 선행 연구를 기반으로 취약점이 발생할 수 있는 공격 표면을 식별하고, 보안 취약점 분석 실험을 하여 공격이 유효한지 파악한다. 이 실험을 통해 관련 연구의 공격 기법 중에 여전히 유효한 공격을 식별하고 공격의 파급력을 분석했다. 실험에 기반한 평가 지표를 도출하고 사용자 수 기준으로 내림차순하여 상위 5개의 상용 확장 프로그램형 패스워드 매니저와 3개의 브라우저 패스워드 매니저를 대상으로 보안성 수준을 평가한다. 그리고 패스워드 매니저 사용 환경을 위한 보안 요구사항을 도출한다.

Ⅱ. 배 경

패스워드 매니저의 서비스 제공사와 그 형태는 다양하지만, 사용 목적과 동작 구조는 유사하다. 그러므로 전체 패스워드 매니저를 관통하는 대표적인 특징을 도출하여 공통의 기준을 정립한 후 보안 취약점을 발생시키는 요인을 식별하여 보완하는 절차로 전체적인 패스워드 매니저의 보안성을 강화할 수 있다. 본 장에서는 패스워드 매니저란 무엇이고, 기본 기능과 특성에 대해 서비스를 제공하는 플랫폼 형식에 따른 특성과 차이점을 설명하며, 연구 대상과 범위를 설정한 기준과 패스워드 매니저 보안 연구 동향 전반에 관해 다룬다.

1. 패스워드 매니저

패스워드 매니저는 사용자 계정 관리 소프트웨어로서 종류에 따라 사용자 아이디나 메일주소, 신용카드 정보, 이름 등의 개인 정보도 다루지만 공통적으로 사용자가 웹 서비스를 이용할 때 주요 인증 정보로 사용하는 ‘패스워드’를 대상으로 동작한다. 패스워드 매니저는 데스크톱과 모바일 환경에서 사용되는데, 본 연구는 데스크톱 환경에서 사용하는 패스워드 매니저를 대상으로 수행하였다.

1) 패스워드 매니저의 기능

패스워드 매니저의 주요 기능은 크게 세 가지로 분류할 수 있다. 첫째로 추측이 어려운 강도의 암호를 ‘생성’하는 기능이다. 패스워드를 생성할 때는 문자 및 숫자, 특수 기호로 구성된 데이터 셋에서 임의로 정해진 길이 만큼의 난수를 선택하여 패스워드로 사용할 것을 권고한다. 그러나 이전 연구에서는

이러한 암호에 일정한 생성 규칙이 있어서 보안성이 낮다고 평가하기도 하였다 [1].

둘째로 사용자가 지속적으로 사용할 계정 정보를 '저장' 하는 기능이 있다. 저장 방식은 소프트웨어 종류에 따라 다르지만, 사용자 PC 내부의 신뢰 영역에 암호화하여 저장하거나 원격 클라우드 서버에 저장한다. 저장된 계정 정보 데이터베이스는 '볼트(vault)' 라고 칭하며, 이 볼트는 '마스터 패스워드(Master Password)' 로 접근한다. 저장 방식에서도 보안 취약점이 발생한다. 이전 연구에서는 사용자 PC에 저장된 볼트를 복호화하는 데 성공하였으며, 네트워크를 통해 원격 클라우드 저장소에 전송하는 과정에서 암호화 통신을 수행하지 않아서 정보가 노출된 바 있다.

세 번째는 '자동 입력' 기능으로, 패스워드가 필요한 상황이 되면 볼트에 저장하였던 패스워드를 불러와서 입력해야 할 필드에 제안하거나 패스워드 매니저가 직접 정보를 작성하는 기능이다.

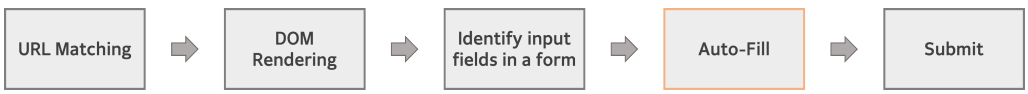

FIGURE 1 Auto-Fill Process of Password Manager

Fig. 1은 패스워드 매니저의 자동 입력 절차를 나타낸다. 사용자가 새로운 웹 페이지를 로딩하는 이벤트를 발생시켰을 때, 첫 번째로 해당 페이지의 URL을 확인하여 저장된 계정 정보 목록에서 이전에 저장했던 이력이 있는지 파악하여 자동 입력이 필요한 페이지인지 구분한다. 로그인이 필요한 페이지라고 판별하면 페이지 내 DOM(Document Object Model) 렌더링을 기다리고 Form 형식과 알맞은 입력 필드를 식별하여 계정 정보를 자동으로 입력한다.

자동 입력 정책은 패스워드 매니저마다 다르게 채택하고 있다. 정책의 기본

활성화 여부부터 상이한 설정을 보이는데, 설치한 이후 사용자가 직접 환경 설정 변경을 하여야 자동 입력 기능이 활성화되는 서비스와 별도로 설정을 수정하지 않아도 자동 입력을 지원하는 서비스로 나뉜다. 자동 입력의 사용성 수준으로는 사용자가 패스워드 매니저의 제안 팝업을 누르는 등의 '상호작용(interaction)'을 수행해야만 입력되는 반자동 입력 방식과 편의성을 극대화하기 위해 로그인 페이지가 렌더링 되면 패스워드 매니저가 곧바로 계정 정보를 입력하는 자동 입력 방식이 있다. 그리고 자동 입력 후 연속적으로 제출 이벤트까지 발생시키는 서비스도 있으나, 이는 사용자의 사용성을 개선하지만, 사용자가 알아채기 어렵게 계정 정보가 탈취될 수 있어 보안성의 측면에서 굉장히 위험한 정책이다. 자동 입력 정책은 실제로 보안 취약점이 발생하게 되는 공격 표면의 많은 비중을 차지하여 많은 패스워드 매니저가 해당 기능을 점차 기본 비활성화 하도록 보안 업데이트 하였다.

정리하면, 패스워드 매니저는 생성, 저장, 자동 입력의 기능을 공통적으로 지원하고, 일련의 기능들은 패스워드 매니저의 '라이프 사이클(Lifecycle)'이라고도 표현한다 [1]. 각 단계에서는 작동 방식에 따른 취약성이 내재되어 있다.

2) 패스워드 매니저의 종류

데스크톱 환경의 패스워드 매니저는 '서비스를 제공하는 플랫폼'이라는 기준에 따라 브라우저 기반에 기본 내재된 패스워드 매니저와 브라우저 확장 프로그램 형태의 패스워드 매니저, 그리고 개별 소프트웨어 형태로 제공되는 패스워드 매니저로 분류할 수 있다.

① 브라우저 기본 패스워드 매니저

데스크톱 환경에서 사용하는 브라우저 애플리케이션은 사용자가 현재 접하

고 있는 웹 페이지 내 계정 정보가 입력되거나 form 작성되는 것을 인식하고, 입력한 정보를 저장하는 기능을 가진 패스워드 매니저를 기본적으로 내재하고 있다. 교육 플랫폼 W3Schools가 조사한 사용자 점유율 5대 주요 브라우저인 Google의 Chrome(78.8%), Microsoft의 Edge(10.3%), Linux 시스템의 Firefox(4.8%), Apple의 Safari(3.9%), Opera 브라우저(2.3%)와 같이 절대다수의 사용자를 가진 브라우저뿐만 아니라 Samsung 브라우저, Naver Whale 브라우저, Brave, 심지어는 다크 웹 전용 브라우저인 Tor도 패스워드 매니저를 기본적으로 지원한다.

브라우저 애플리케이션에 탑재된 패스워드 매니저는 별도 설치의 과정이 없고 기본적으로 활성화되어 있어서 접근성이 좋으므로 사용자가 많다. 실제로 과거 사례 연구에 따르면 브라우저 사용자들은 높은 편의성으로 내장된 패스워드 매니저를 사용하고 있다고 보고했다 [2]. 따라서 높은 사용률을 보이는 브라우저의 패스워드 매니저에 적절한 보안 정책이 적용되었는지 확인해야 한다. 그러나 종래 연구에서 브라우저의 패스워드 매니저는 다양한 취약성을 내재하고 있는 것으로 발표하였으나 그 후 수행된 연구에서 여전히 동일한 공격 표면이 유효한 것을 입증하였다. 그리고 가장 최근의 브라우저 패스워드 매니저의 취약점 분석을 수행한 연구에서 다룬 크롬 브라우저의 패스워드 매니저의 경우 71 버전을 대상으로 하였으나 [1], 2023년 11월 기준 크롬 브라우저는 119 버전을 채택하고 있으므로 최신 버전의 크롬 브라우저 패스워드 매니저에 관한 연구가 필요하다.

② 브라우저 확장 프로그램 유형의 패스워드 매니저

브라우저 확장 프로그램이란, 브라우저 제공자가 직접 구현한 것은 아니지만 사용자들에게 보다 편리한 브라우징 환경을 제공하기 위해 지원하는 일종의 브라우저 커스터마이징(customizing)을 가능하게 하는 연결 프로그램을

말한다. 크롬의 익스텐션(extension), 파이어 폭스의 애드온(add-on), 사파리와 엣지의 확장 등 수많은 브라우저가 확장 프로그램을 통한 브라우저 커스터마이징을 지원한다.

이 브라우저 확장 프로그램의 형식으로 제공되는 패스워드 매니저의 종류가 굉장히 다양하고, 각 서비스의 높은 사용자 수를 공식 확장 스토어와 통계 자료로 확인할 수 있다. 브라우저 확장 프로그램은 서로 다른 확장 프로그램 간의 간섭을 일으킬 수 있고, 내부 렌더링 엔진을 통해 권한 상승 공격에 성공할 수 있어 시스템적인 문제에도 노출된다는 문제가 있다. 그리고 웹 서비스 사용자가 현재 보고 있는 페이지의 DOM을 수정할 수 있어 공인된 사이트를 해킹하지 않더라도 쉬운 해킹 기법으로 사용자의 계정 정보를 노출시킬 수 있어서 큰 피해가 예상된다.

그러므로 본 연구는 확장 프로그램을 다운로드한 수에 비례하여 확장 프로그램 유형의 패스워드 매니저를 선정하기 위해 사용자 수 분석을 수행하였다. 2023년 11월 기준, 각 브라우저의 확장 프로그램 정보를 분석하는 사이트를 기반으로 사용자 수가 많은 순으로 주요 3대 브라우저의 확장 프로그램 기반의 패스워드 매니저를 나타내면 다음 TABLE 1에 정리하였다.

프로그램의 다운로드 수는 실시간으로 변동되는 특성을 가지고 있기 때문에 본 연구에서는 천의 자리 수 이하는 버림하고, 플러스 기호(+)로 표기하였다. 그리고 RapidIdentity 패스워드 매니저는 크롬 브라우저에서만 지원하여 엣지와 파이어폭스 집계에서는 마이너스 기호(-)로 표기하였다.

TABLE I

패스워드 매니저 확장 프로그램의 사용자 수

사용자 수 순위	서비스 명	Chrome	Edge	Firefox
1	LastPass	10,000,000+	3,960,000+	570,000+
2	Avira	6,040,000+	4,260,000+	18,000+
3	Norton	4,310,000+	4,500,000+	320,000+
4	1Password	4,120,000+	1,010,000+	300,000+
5	Bitwarden	3,630,000+	1,410,000+	660,000+
6	Dashlane	2,190,000+	860,000+	140,000+
7	RapidIdentity	1,460,000+	–	–
8	Keeper	1,210,000+	710,000+	42,000+
9	RoboForm	654,000+	500,000+	60,000+
10	DualSafe	518,000+	350,000+	180+

크롬 브라우저의 점유율이 높은 비중을 차지하고 있으므로 브라우저 별 사용자 수의 차이가 크게 벌어졌으나, 브라우저 별 확장 프로그램의 순위는 비교적 유사한 양상을 띤다. 크롬과 엣지 브라우저에서 사용자 수가 가장 많은 확장 프로그램은 LastPass이고, Firefox 브라우저는 Bitwarden인데, 이는 LastPass와 Bitwarden은 유료 구독 서비스 형태가 아니기 때문에 많이 활용되고 있는 것으로 보인다. 종합적으로 합산하여 본 연구에서 보안성 평가를

목표로 하는 패스워드 매니저는 상위 5개의 패스워드 매니저인 LastPass, Avira, Norton, 1Password, Bitwarden이다.

본 연구는 사용자 수 상위 5개의 확장 프로그램 유형의 패스워드 매니저와 점유율 상위 3개의 브라우저 기본 제공 패스워드 매니저, 총 8종의 애플리케이션을 대상으로 수행하였다. 대상 패스워드 매니저의 특징과 지원하는 기능은 다음 TABLE II와 같다.

TABLE II

연구 대상 패스워드 매니저의 기능 분석

	서비스 명	패스워드 생성 기능	마스터 패스워드 체제	자동 입력 정책	최신 버전
1	LastPass	○	○	자동	4.123.0
2	Avira	○	◎	자동	2.19.14.4461
3	Norton	○	◎	수동	8.1.1.39
4	1Password	○	◎	수동	2.17.1
5	Bitwarden	○	○	수동	2023.10.2
6	Chrome	○	×	자동	119.0.6045.160
7	Edge	○	×	자동	119.0.2151.72
8	FireFox	○	×	자동	120.0

연구 대상 패스워드 매니저 모두 패스워드 생성 기능을 제공하였다. 그러나 이중 LastPass, Avira, Norton, 1Password, Bitwarden, Chrome, Edge 패스워드 매니저는 문자와 숫자, 특수문자를 적절히 섞은 높은 강도의 암호를 추천하였지만, Firefox는 문자와 숫자 데이터 셋 안에서만 한정되어 패스워드를 생성하여 추천하였다.

마스터 패스워드란 패스워드 매니저를 활성화하고 볼트에 접근하기 위해 사용하는 패스워드 매니저의 계정 정보를 의미한다. 5개의 확장 프로그램 기반 패스워드 매니저는 모두 마스터 패스워드 체제를 사용하였지만, 브라우저 기본 내장형 패스워드 매니저는 브라우저의 설정에서 부가적인 인증 절차 없이 패스워드의 평문을 확인할 수 있다. 확장 프로그램 기반 패스워드 매니저 사이에도 차이가 있으며, 이는 표의 중첩 원 모양과 단일 원 모양으로 구분하였다. 단일 원 모양은 마스터 패스워드로 패스워드 매니저를 활성화하는 것을 의미하고, 중첩 원모양은 암호 저장소인 볼트(vault)에 접근하기 위해 추가적인 볼트용 마스터 패스워드로 인증을 수행해야 하는 경우이다.

자동 입력 정책은 패스워드 매니저마다 다른 양상을 보였다. 브라우저 기본 내장형 패스워드 매니저는 편의성과 접근성을 극대화하여 모두 기본 설정값이 자동 입력이나, 브라우저 패스워드 매니저는 대부분 수동 입력으로 사용자의 상호작용을 거쳐 자동 입력하였다.

버전은 2023년 11월 기준 가장 최신 업데이트 버전을 의미하며, 실험 및 평가 또한 최신 버전을 대상으로 수행하였다.

③ 데스크톱 로컬 설치 유형의 패스워드 매니저

확장 프로그램으로 제공되는 형태가 아닌, 사용자 데스크톱에 직접 다운로드하여 사용하는 소프트웨어를 본 연구에서 데스크톱 로컬 설치 유형의 패스워드 매니저라고 분류하였다. 이러한 유형의 패스워드 매니저는 컴퓨터에 개별 프로세스를 가져 자원을 할당받고 평소 서비스 백그라운드(background) 측에서 대기 상태를 유지하다가 계정 정보를 입력해야 할 때가 감지되면 포그라운드(foreground)로 전환되어 작동한다. 이와 같은 동작 구조는 모바일 환경에서 사용되는 패스워드 매니저와 유사하며, 데스크톱 클라이언트를 PC에 설치하여 앞서 언급한 패스워드 매니저와 같이 웹앱으로 접근하는 것이 아니

라 개별 소프트웨어 실행을 통해 사용할 수 있다.

웹 기반 패스워드 매니저가 실질적으로 웹 상에서 통신하고, 저장하는 구조와는 다른 점이 많아서 같은 기준으로 보안성을 평가하기 어렵다고 판단되어 본 연구의 범위에서 제외하였다.

2. 위협 모델 및 연구 범위

본 연구에서 가정하는 위협 모델은 패스워드 탈취를 목적으로 하는 공격자가 보안이 취약한 공인 웹사이트를 점유하거나, 일반 확장 프로그램의 일부를 조작하여 사용자가 로그인을 시도하는 웹 페이지에 영향을 줄 수 있는 상황을 가정한다. 또한 사용자는 수동적으로 일반 포털 사이트만 사용하는 것이 아니라 메일 서비스나 확장 프로그램 등 다양한 애플리케이션을 활용하는 것을 가정한다.

패스워드 매니저의 기능인 생성, 저장, 자동 입력 중 본 연구에서는 생성과 저장 과정의 보안성 실험은 제외하였고, 생성 정책과 저장 형태와 저장소 접근 방식 등의 저장 정책을 검토하였다. 자동 입력 기능의 경우에는 공격 표면을 식별하고 실질적인 보안성 실험을 수행하였으며 연계 공격을 구현하여 실현 가능성을 입증하였다.

3. 패스워드 매니저 보안 연구 동향

전 세계적으로 패스워드 매니저의 보안 연구가 많이 발표되었다. 국내 연구의 경우 포렌식 수사 관점의 연구와 패스워드 저장소의 취약성을 활용한 연구가 2020년 전후로 수행되었고, 해외에서는 다양한 웹 해킹 기법이나 웹 브라우저의 요소들과 결합된 취약점 분석 관점의 연구가 2014년 전후로 활발히 수행되었다. 해외는 국내에 비하여 오래전부터 웹 브라우저 취약점 기반의 웹 해킹에 관한 연구를 수행하였음을 알 수 있고, 최근에는 보안성 연구보다 사용성을 향상하기 위한 연구가 수행되고 있다. 이러한 경향은 사용자들이 더욱 편리한 웹 환경을 선호하고 있어서 패스워드 매니저를 선택할 때 사용성과 편의성이 가장 중요한 요소인 것으로 밝혀졌다[2]. 그리고 패스워드 매니저의 보안성 강화에 기여한 대표적인 연구는 다음과 같다.

Huaman, N. et al. [3] 연구진이 수행한 연구는 가장 최신의 수행된 패스워드 매니저에 관한 연구이며, 사용성의 관점에서 패스워드 매니저와 웹 사이트 간 상호작용(interaction)이 실패하는 요인을 분석한 연구이다. 상호작용이란 올바른 패스워드 입력 란에 저장된 패스워드를 붙여 넣는 이벤트인 자동 입력을 의미한다. 이 자동 입력이 필요할 때 수행되지 않으면 사용자들은 불편함을 느끼게 되어 사용자 이탈률이 증가한다. 따라서 이 상호작용 문제 패턴에 관한 평가 항목을 도출하여 15개의 패스워드 매니저를 평가하였다. 평가 항목은 이용자 수 상위 30개 패스워드 매니저에 기록된 사용자 리뷰를 메타 분석하여 정리하였다. 실 사용자들의 리뷰를 중심으로 상호작용이 실패하게 되는 케이스를 많이 수집하여 체계적인 평가 항목을 도출하였는데, 이는 본 연구에서 다중 입력 필드 공격 표면 부분에서 참고할 수 있는 케이스였다. 그러나 사용성을 고려한 연구는 그 이면에

보안성 저하를 크게 발생시키는 요인이 많기 때문에 보안 연구의 필요성이 더 크게 부각되었다.

Oesch, S. et al. [1] 연구진이 수행한 연구는 패스워드 매니저의 수명 주기(lifecycle)를 생성, 저장, 자동 입력으로 규정하고 각 단계에서 나타날 수 있는 취약성을 식별하였다. 생성의 경우 패스워드 매니저가 제안하는 무작위 패스워드를 머신러닝 알고리즘 기반으로 분석해 발생하는 문자의 분포도가 적합한 복잡도를 가지는지 판별하였고, 저장 및 자동 입력의 경우에는 이전 연구에서 식별하였던 공격 기법들이 여전히 유효한지 실험하는 방식으로 연구 진행하였다. 최초로 '생성'에 대한 보안성 분석을 수행했다는 것에 큰 이점이 있다.

Silver, D. et al. [4] 연구진이 수행한 연구는 패스워드 매니저를 대상으로 한 대표적인 취약점 분석 관점의 연구로 꼽힌다. 이 연구는 네트워크 환경에서 중간자 공격(Man-in-the-Middle Attack)이 가능한 공격자를 주 위협 모델로 가정하고 느슨한 자동 입력 정책에 따라 발생할 수 있는 보안 취약성을 분석했다. 카페 등의 공공 오픈 Wi-Fi를 통해 접속한 피해 PC는 준비된 악성 랜딩 페이지로 스윕 공격을 입게 되며, 이 과정에서 네트워크 트래픽을 변조하는 방법으로 공격의 시작점을 가정하였다. 소개된 다양한 공격 기법들이 10년 전의 인터넷 환경과 웹 브라우저의 보안 정책에 종속된 방법이므로 더 이상 공격이 유효하지 않다는 한계가 있다. 즉, 패스워드 매니저의 보안성 강도에 따른 공격 기법 성공률이 아닌, 오랜 시간 흐름에 따른 웹 브라우저 보안성의 강화로 공격이 실패하게 되는 것이므로 최신화해야 한다. 본 연구에서도 느슨한 자동 입력 정책에 따른 취약성을 줄이는 것을 주 보안 목표로 다루는 만큼 밀접한 연구라고 할 수 있다.

Li, Z. et al. [5]가 수행한 연구는 패스워드 매니저에서 나타날 수 있는

북마클릿, 취약한 웹 사이트에 따른 취약점, 협업 상황에서 발생하는 미흡한 권한 부여 취약점, 피싱을 목적으로 한 사용자 인터페이스 취약점, 총 4가지 상황에 관해 연구를 수행하였다. 본 연구는 자세한 사례 분석을 통해 구체적인 취약점 발현 과정을 공유하여 이후 수행된 많은 패스워드 매니저 연구의 참고가 되고 있다. 그러나 2014년에 수행된 연구임에 분석 대상이었던 5개의 패스워드 매니저 중 3개가 현재 서비스가 종료되어 사용되지 않는 소프트웨어이며, 약 10년의 기간 동안 급변한 웹 환경에 맞춘 최신의 연구가 필요하다는 단점이 있다.

Table III은 국내에서 수행된 패스워드 매니저 대상 연구의 기여점을 요약했다. 국내의 연구는 공통적으로 패스워드 매니저의 저장 기능에서 발생하는 취약점을 다루었고, 사용자가 분실 등의 사유로 공격자가 물리적으로 기기를 확보하여야 계정 탈취가 가능한 시나리오라는 점을 한계로 꼽을 수 있다. 또한 LastPass나 Chrome 등 특정 패스워드 매니저에 국한한 연구를 수행하여 보편적으로 적용하기 어렵다는 한계점이 있다.

TABLE III

국내 패스워드 매니저 연구 동향

Category	Refs	Contribution	Year
저장방식	[6]	- Chrome 패스워드 매니저, 데스크톱 LastPass, 모바일 LastPass를 대상으로 패스워드 매니저 동작 로직을 분석하여 변조에 성공함 - 패스워드를 저장하는 데이터베이스 접근에 암호화가 적절하지 않다는 점을 연구하여 평문 노출에 성공함	2018
저장방식	[7]	- 데스크톱 LastPass를 대상으로 원격 클라우드 서버와 통신할 때 사용하는 마스터 패스워드를 로컬에서 탈취하여 저장한 패스워드를 복호화함	2018
네트워크	[8]	- 데스크톱 LastPass와 KeePass를 대상으로 원격 클라우드 서버와 통신할 때 네트워크를 분석하여 암호화 과정의 결함을 연구함	2020
포렌식	[9]	- Chrome 패스워드 매니저를 대상으로 포렌식 수사에 있어 재현성과 관리 연속성의 원칙을 준수하여 피의자가 범죄 혐의를 부인할 때 저장된 계정 정보로 적법한 흔적을 얻는 방법을 고안함	2021

Ⅲ. 패스워드 매니저 취약점 분석 방법론

패스워드 매니저는 웹 서비스 사용자가 현재 보고 있는 페이지 내의 form 객체를 추적하며 제출(submit) 이벤트가 발생하는지 포착한다. 그 시점에 현재 페이지의 URL(Uniform Resource Locator)과 form 객체에 입력된 아이디, 패스워드 정보의 저장 여부를 사용자에게 묻고, 저장하겠다는 의사 표현을 받게 되면 패스워드 매니저가 가리키는 저장소에 저장한다. 이후 저장된 URL과 일치하는 페이지가 렌더링 되었을 때 form 객체를 파싱(parsing)하면 자동으로 채우거나, 자동 입력이 필요한지 사용자에게 질의한 후 처리한다. 실제 패스워드 매니저 상에 저장되는 형태는 Fig. 2와 같다.

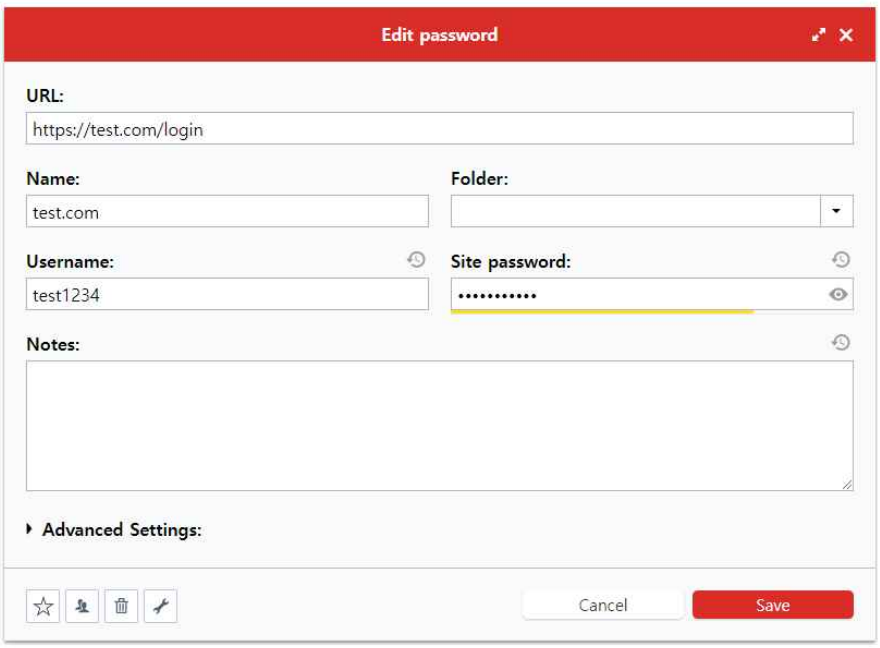

FIGURE 2 LastPass 패스워드 매니저의 저장된 계정 정보

패스워드 매니저의 이러한 로직은 보편적인 웹 로그인 페이지 구성에 맞추어 구현되었다. 그러므로 일반적인 웹 페이지처럼 보이더라도 패스워드 매니저의 기본적인 특성이 악용되도록 변형을 가하면 패스워드 매니저를 속여 사용자의 정보를 탈취할 수 있다. 패스워드 매니저는 변형된 웹 페이지 상에서 적절한 로그인 필드에 자동 입력을 수행하거나 자동 입력을 수행하지 않는 반응, 악성 변조에 잘못된 자동 입력을 수행하거나 악성 변조를 인지하여 자동 완성을 수행하지 않는 4가지 반응을 보일 수 있다. 그러므로 본 연구에서는 변형된 웹 페이지를 구성하고, 패스워드 매니저의 반응성 확인을 통해 패스워드 매니저의 보안성을 분석하였다.

그리고 패스워드 매니저는 사용자의 사용성을 고려하여 느슨한 정책을 채택한 경우가 많다. 이 영향으로 현재 사용자가 인가된 사용자인지, 또 해당 암호가 이전에 저장된 사이트와 동일한지, 무결성이 훼손된 사이트는 아닌지 부차적으로 검증하지 않아서 취약성이 발생한다. 본 장에서는 패스워드 매니저의 취약점이 발생하였던 이전 연구 및 사례를 검토하여 취약성 공격 표면을 도출하고, 이 취약성으로 계정 정보가 유출될 수 있는 시나리오를 구성하여 실제 유출 사고 가능성을 입증하는 방식으로 취약점 분석하였다.

1. 변형된 웹 페이지 처리 방식에 따른 취약성

웹 서비스 사용자가 웹 페이지 내의 로그인 폼을 이용할 때 로그인 정보를 자동으로 완성시키는 기능인 자동 입력 기능은 패스워드 매니저의 편의성을 증진시키는 기능이자 패스워드 매니저의 주요 기능이다. 최근 수행된 패스워드 매니저를 대상으로 한 연구는 패스워드 매니저의 사용성을 더 높여 사용자의 유입을 늘리기 위해 이전보다 엄격하지 않은 자동 입력 정책

을 장려하는 연구가 수행되고 있으나 이 정책은 공격 표면을 확장 시켜 보안 취약성을 유발한다. 본 장에서는 사용성 향상을 위한 느슨한 보안 정책 때문에 발생할 수 있는 공격 표면에 관해 서술한다.

공격 표면 산출을 위해 최근 10년 동안 수행된 패스워드 매니저의 자동 입력 취약점을 다룬 논문을 모두 수집하여 분석하였으며, 이전 연구에서 공개된 취약점 중 공통적인 기능을 통해 유발되어 보편적으로 적용하고 평가할 수 있는 항목을 Table Ⅳ에 정리하였다. 취약점이 공개됨에 따라 보안성 강화 조치가 이루어진 공격 표면들은 제외하였다.

TABLE Ⅳ

취약한 자동 입력 정책에 따라 발현되는 취약성 리스트

Clauses	Refs.
1 취약한 URI 식별 정책	[3], [4], [10]
2 다중 도메인에 대한 반응 정책	[1], [4]
3 DOM 수정 확인 정책	[1], [4]
4 다중 Input Field 처리 정책	[3], [11]
5 비표준 로그인 필드 처리 정책	[1]

1) 취약한 URI 식별 정책

패스워드 매니저가 최초로 계정 정보를 저장할 때 계정 정보뿐 아니라 도메인 주소, 프로토콜 등 현재 페이지를 기억할 수 있는 부차적인 정보도 함께 저장한다. 이 정보를 기반으로 이후 동일한 URI를 식별했을 때 저장한 계정 정보를 불러와서 자동 입력할 수 있다. 본 연구에서는 이 URI를

저장하고, 감지하는 정책을 ‘패스워드 매니저의 URI 식별 정책’이라고 칭하였다.

패스워드 매니저 자체의 URI 식별 정책이 엄격하지 않은 경우에는 계정 정보를 입력해야 할 URI가 아닌 유사한 URI인 경우에도 자동 입력을 할 수 있다. 가장 흔하게 사용되는 느슨한 정책은 회원 가입 페이지인 https://test.com/signup에서 저장한 계정 정보를 로그인 페이지인 https://test.com/signin에 붙여 넣는 경우이다. 회원 가입과 동시에 계정 정보를 저장하여 사용하는 것은 사용자에게 큰 사용성 효과를 주기 때문에 대부분의 패스워드 매니저는 프로토콜과 포트, 하위 페이지 주소까지 정확히 대응하지 않고 도메인 이름(Domain name) 일치 정책을 사용한다. 대형 사이트의 경우, 수많은 하위 사이트가 연결되어 있기 때문에 이러한 정책은 편리할 수 있지만, 만약 큰 사이트 내에 가장 약한 하위 사이트가 공격자에게 해킹된 상황이라면 계정 정보 탈취 사고가 발생할 수 있다.

2) 다중 도메인에 대한 반응 정책

다중 도메인이란 웹 페이지 내에 다른 HTML 페이지를 삽입하여 서로 다른 도메인의 요소로 한 페이지가 구성된 경우이다. 본 연구에서 언급하는 다중 도메인이란 iframe(inline frame) 객체를 통해 구성된 페이지를 의미한다. 이러한 다중 도메인 페이지는 단일 HTML로 구성된 페이지와 비교하여 풍성한 페이지를 구성할 수 있다는 장점이 있지만, 패스워드 매니저의 자동 입력 기능의 오작동으로 사용자의 중요 정보가 노출되는 공격 표면이 발생한다. iframe을 통해 포함된 페이지의 URI는 사용자가 직관적으로 확인할 수 없기 때문에 공격 성공 가능성이 높다. 가령, Cross-origin 정책을 허용된 웹 사이트라면 iframe을 이용하여 공격자의 사이트에서 탈취하고 싶은 공인된 사이트의 로그인 폼 객체만을 가져와서 패스워드 매

니저에게 혼란을 줄 수 있다 [4]. 이와 같이 iframe에 로딩된 페이지를 단일 페이지로 인식하고 자동 입력을 하는 경우에는 보안 취약성을 불러일으킬 수 있기 때문에 유의해야 한다. 최근에는 웹 브라우저 보안성 강화로 서로 다른 도메인의 iframe 사용을 엄격하게 관리하고 있다. 현대 브라우저에서는 CORS(Cross-Origin Resource Sharing) 정책이 도입된 이후로 iframe으로 페이지를 구성하는 것이 어려워졌지만, 이는 브라우저의 보안 정책이므로 패스워드 매니저가 적절한 보안 반응성을 보이는 지는 확인이 필요하다.

3) DOM 변조 확인 정책

DOM(Document Object Model)이란 웹 페이지의 구조를 프로그래밍적으로 조작할 수 있게 해주는 인터페이스로, HTML이나 XML 형식으로 구성된 문서의 요소를 구조화해서 하나의 객체로 접근할 수 있도록 하는 모델을 말한다. 이 DOM은 JavaScript를 통해 조작할 수 있으며 웹 페이지를 동적으로 만들 수 있다.

그러나 웹 페이지를 구성하는 DOM이 일부 수정되었음에도 패스워드 매니저는 이를 정밀하게 검증하지 않고 자동 입력을 하여 보안 취약점이 발생할 수 있다. 예를 들어, 입력 필드의 값을 변수로 저장한 form 객체의 input id가 수정되거나 class 명이 수정되더라도 패스워드 매니저는 input field만을 식별하고 자동 입력한다. 또한 페이지 내에 form 객체를 추가 삽입하며 정상적인 필드(field)가 아닌 공격자가 임의로 삽입한 필드에 자동으로 입력하도록 하는 공격도 가능하다. 사소한 DOM 수정에도 사용자 계정 정보가 유출될 수 있기 때문에 패스워드 매니저의 위험한 정책이 현재까지 유효한지 확인하고 보완해야 한다.

4) 다중 Input Field 처리 정책

패스워드 매니저는 웹 페이지 로딩에 따라 로그인이 필요한 페이지라는 것을 인식하게 되면, 페이지를 구성하고 있는 모든 Input field를 식별한다. Input field는 일반적인 아이디나 패스워드 정보 뿐 아니라 OTP(One-Time Password), 이름, 생년월일과 같이 추가적인 인증 정보를 요구할 수 있다. 이와 같은 다양한 계정 정보를 요구하는 페이지를 접하게 된다면 패스워드 매니저는 혼란을 겪어 작동하지 않거나, 계정 정보를 붙여 넣어야 할 곳이 아님에도 오동작한다. 이러한 반응은 침해된 사이트에 공격자가 악성 form 객체를 숨겨둔 페이지라면 사용성 문제로 그치지 않고, 계정 정보가 유출될 수 있다.

5) 비표준 로그인 필드 처리 정책

비표준 로그인 필드란 직접적인 로그인 역할을 하는 form 객체는 아니지만, 각 input type=text와 input type=password 속성을 가진 두 개의 입력 필드를 가진 객체를 말한다. 패스워드 매니저는 사람의 인지 처리와 달리 페이지의 코드를 분석하여 보편적인 로그인 속성을 가진 두 개의 입력 필드가 포함되어 있으면 로그인 양식이라고 판단하는 로직을 가진다. 따라서 실제 로그인 역할을 하는 필드가 아닌 비표준 로그인 필드를 오탐하여 자동 입력 기능을 수행할 수 있는데, 이는 취약점과 연계될 수 있다.

2. 취약점 구현 방법

본 장에서는 패스워드 매니저가 변형된 웹 페이지를 처리하는 취약한 정책에 따라 발생하는 보안 취약성을 기반으로 실제 보안 사고로 이어질 수 있는 연계 공격 구현 방안을 도출한다. 취약성이 실제 취약점으로 발현하기 위해서는 여러 절차가 필요하므로 단일의 취약성을 모두 방지하는 것은 비효율이 따를 수 있다. 그러나 본 연구에서 입증한 취약점은 침해 사고 발생 가능성이 높기 때문에 적절한 보안성 패치를 권고한다.

1) 메일 서비스를 통한 비표준 로그인 필드 구현

여러 포털 사이트는 사용자와 사용자 간 메시지 및 자료를 주고받을 수 있는 메일 서비스를 지원한다. 폭넓은 메일 내용을 지원하기 위해서 많은 메일 서비스는 HTML을 담아 주고받을 수 있도록 서비스한다. 이 점을 활용해서 전달하는 메시지의 가독성을 좋게 만들거나 영상과 이미지와 같은 미디어 매체를 첨부할 수 있지만, form 태그와 같이 일반적인 로그인 페이지에 사용되는 문법도 지원하는 경우 보안 사고로 이어질 수 있다. 예를 들어, 공격자는 수신자에게 비표준 로그인 필드를 노출시키는 이메일 피싱 공격을 수행할 수 있다. 피싱 메일을 통해 수신자의 화면 내 배치된 비표준 로그인 필드에 패스워드 매니저가 반응할 수 있어 메일 서비스를 제공하는 도메인의 계정 정보를 탈취할 수 있다.

2) 악성 확장 프로그램을 통한 평문 노출

HTML의 input 객체 중 type 태그는 입력되는 값의 속성을 매칭하는 역할을 한다. 속성은 입력되는 값의 성질이나 형식을 지정할 수 있어 많은 웹페이지에서 사용된다. 예를 들어, 로그인 폼의 아이디에 해당하는 부분

은 input type=text이고 패스워드에 해당하는 부분은 input type=password인데, text 타입인 경우에는 입력된 문자의 평문이 노출되고, password 타입인 경우에는 입력한 문자가 큰 점으로 표시되며 복사 및 붙여넣기 기능도 불가하다. 따라서 모든 로그인 필드의 패스워드 입력란은 input type=password로 구성되어 있다. 그러나 이 속성 값을 변조한다면 패스워드 매니저가 로그인 폼의 형태를 인식하더라도 자동 입력을 수행할 것인지 확인하여야 한다. 평문으로 노출된 패스워드는 스크린 샷 기능이나 클립보드, 악성 자바스크립트 삽입을 통해 유출시킬 수 있다.

본 연구에서는 취약점 구현을 위해 악성 확장 프로그램을 가정하였다. 확장 프로그램은 프로그램의 자세한 명세서가 되는 manifest 파일과 사용자가 브라우저를 사용하는 동안 백그라운드에서 원하는 이벤트가 발생할 때까지 대기하는 Background 스크립트, 현재 페이지의 DOM을 읽거나 수정하여 확장 프로그램의 직관적인 기능 구현을 담당하는 Content 스크립트로 구성된다. 각 스크립트의 동작 구조는 Fig. 3에서 확인할 수 있다. 이러한 성질을 고려해 악성 확장 프로그램을 구현하여 현재 보이는 페이지의 성질을 인식하여 DOM 속성 값을 수정하는 방식을 채택하였다.

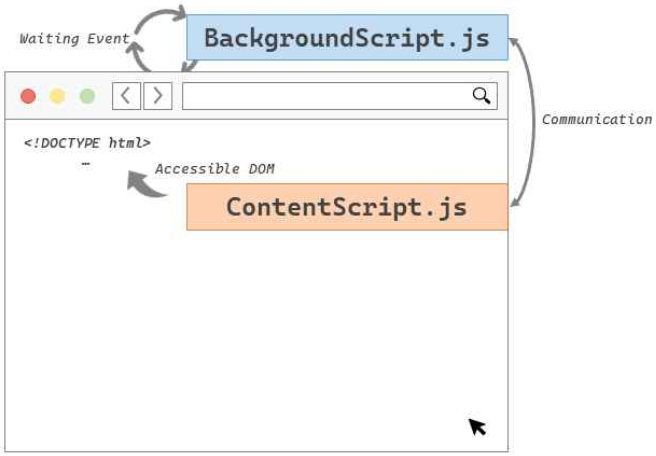

FIGURE 3. 웹 페이지 상에 영향을 줄 수 있는 크롬 확장 프로그램의 주요 스크립트 모식도

3) 악성 확장 프로그램을 통한 DOM 변조 구현

확장 프로그램 간에는 브라우저에서 제공하는 API를 사용하여 통신한다. 브라우저 제공사는 확장 프로그램이 시스템에 영향을 크게 미치지 않도록 권한을 분리하여 계층 별로 스크립트를 분리하여 작동하도록 복잡한 아키텍처를 설계하였다. 예를 들어, Chrome 확장 프로그램의 경우, Content Script는 DOM을 수정할 수 있어 가장 사용자에게 크게 영향을 끼칠 수 있는 스크립트지만, 취약성을 최소화하기 위해 굉장히 제한된 권한만 부여되어 있다. 시스템 단까지 미칠 수 있는 스크립트인 Background Script는 사용자가 체감할 수 있는 이벤트가 발생하지 않더라도 Back-end 단에서 브라우저의 라이프 사이클 내에서 발생하는 수많은 이벤트를 감지하고 데이터를 주고 받을 수 있다. 그렇지만 보안성을 크게 고려한 아키텍처에도 허점이 존재한다.

페이지 내의 form 객체를 구성하는 태그는 input type, action url 등 다양하며, 이와 같은 태그에 매핑되는 변수에 따라 사용자들이 입력한 정보의 행선지가 정해진다. 본 연구에서는 확장 프로그램의 ContentSciprt가 DOM을 수정할 수 있다는 점을 이용하여 action URI을 수정해 탈취하는 방식을 보인다. Content Script의 기능으로 랜더링 된 페이지 내의 DOM을 수정하여 사용자의 계정 정보가 원래 인증 서버가 아닌 공격자의 서버에 전송되도록 하였다. Fig. 4는 그 공격 기법의 동작 방식을 보여준다.

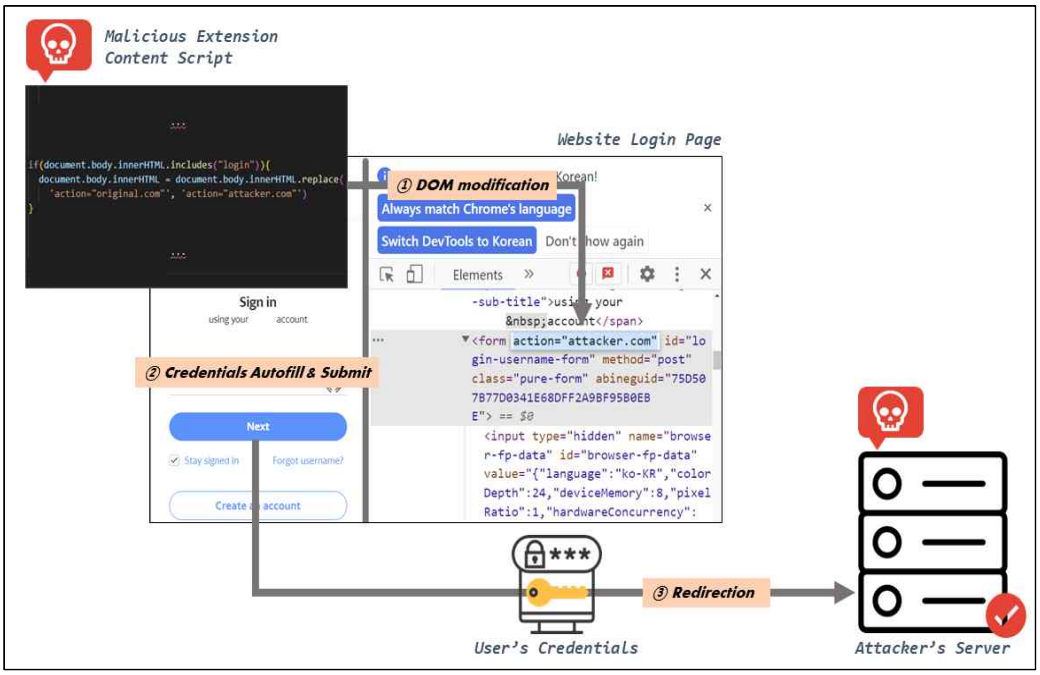

FIGURE 4. 크롬 확장 프로그램을 통해 구현할 수 있는 DOM 수정 및 계정 탈취 방식 모식도

제안하는 방식은 시스템 계층까지 위/변조 수행해야 하지 않고 단 몇 줄의 JavaScript 코드만으로 구현하여 적은 노력으로 파급력이 큰 공격을 성공시킬 수 있음을 증명한다.

Ⅳ. 취약점 분석 결과

본 장에서는 3장에서 도출한 취약점 유발 벡터를 기반으로 실험적으로 취약점의 실현 가능성이 있는 공격인지 파악한다.

1. 변형된 웹 페이지 처리 방식에 따른 취약성

첫 번째 공격 표면인 변형된 웹페이지에 따른 패스워드 매니저의 반응성 평가를 위해서 각 항목에 부합하는 취약한 웹페이지를 구현하였다. 실험 환경은 가상머신을 활용한 Ubuntu 20.04 운영체제에서 Python3의 Flask 모듈을 사용해 로컬 서버를 구동하여 각 페이지가 연계될 수 있도록 구성하였다.

1) 취약한 URI 식별 정책

Fig. 5와 같이 회원 가입 페이지를 구현하고, 아이디와 패스워드를 입력하여 패스워드 매니저가 로그인 정보를 저장하도록 한다.

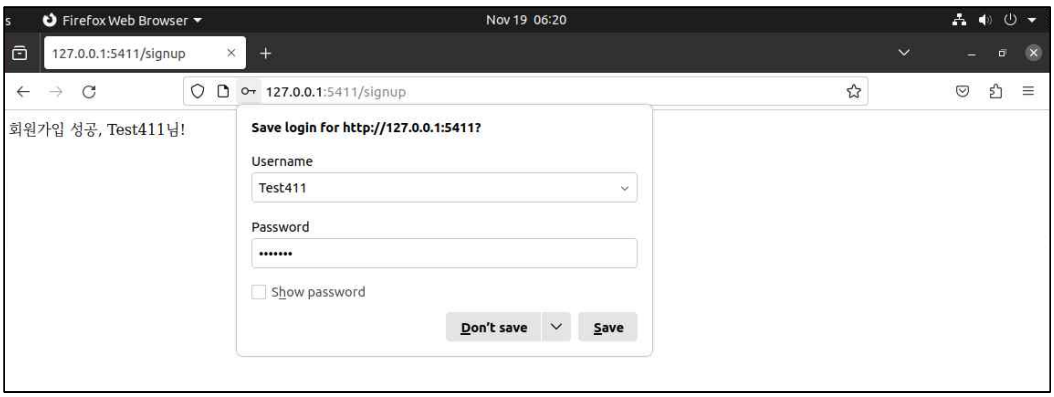

FIGURE 5. 저장 시의 URI와 정확히 일치하지 않음에도 자동 입력 기능이 발현된 결과

이후 도메인은 같지만 다른 하위 페이지의 로그인 페이지에서 로그인을 시도하였을 때 패스워드 매니저의 반응을 확인하였고, 저장된 계정 정보를 아무런 경고 메시지 없이 웹 서비스 사용자가 입력 창을 인지하기도 전에 자동 입력한 것을 확인하였다. 패스워드 매니저가 사용성 증진을 목적으로 회원가입-로그인 관계의 이름을 가진 페이지에만 취약하게 동작되도록 설계되었는지 확인하기 위해 관련 없는 하위 페이지를 구현하였으나 Fig. 6에서 보이는 것처럼 동일한 반응을 보였다.

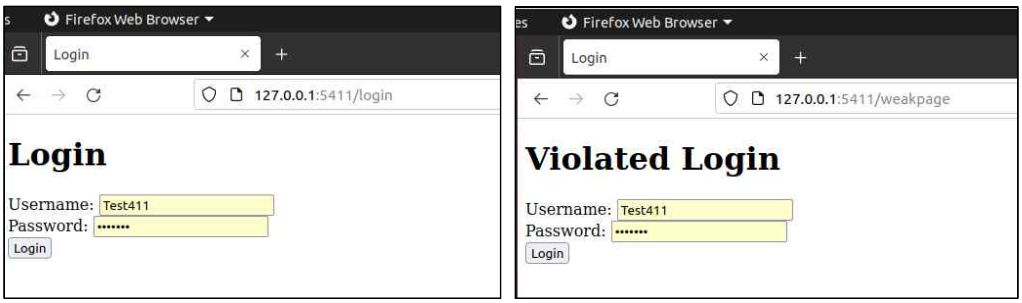

FIGURE 6. 로그인과 직접적인 관련이 없는 이름을 가진 하위페이지에서도 자동 입력 기능이 발현된 결과

2) 다중 도메인에 대한 반응 정책

다중 도메인 환경을 구성할 수 있도록 iframe 태그를 활용해서 Fig. 7과 같이 여러 개의 도메인이 한 페이지에 혼재하도록 구현하였다. 웹 페이지는 로그인 기능을 가진 페이지와 기존 로그인과 iframe으로 불러온 다른 사이트의 로그인 페이지가 함께 있는 페이지를 구성하였고, 패스워드 매니저가 기존 로그인 페이지에서 저장한 계정 정보를 기존 로그인과 iframe 로그인 창이 함께 있는 페이지에서 어떤 반응을 보이는지 확인하였다.

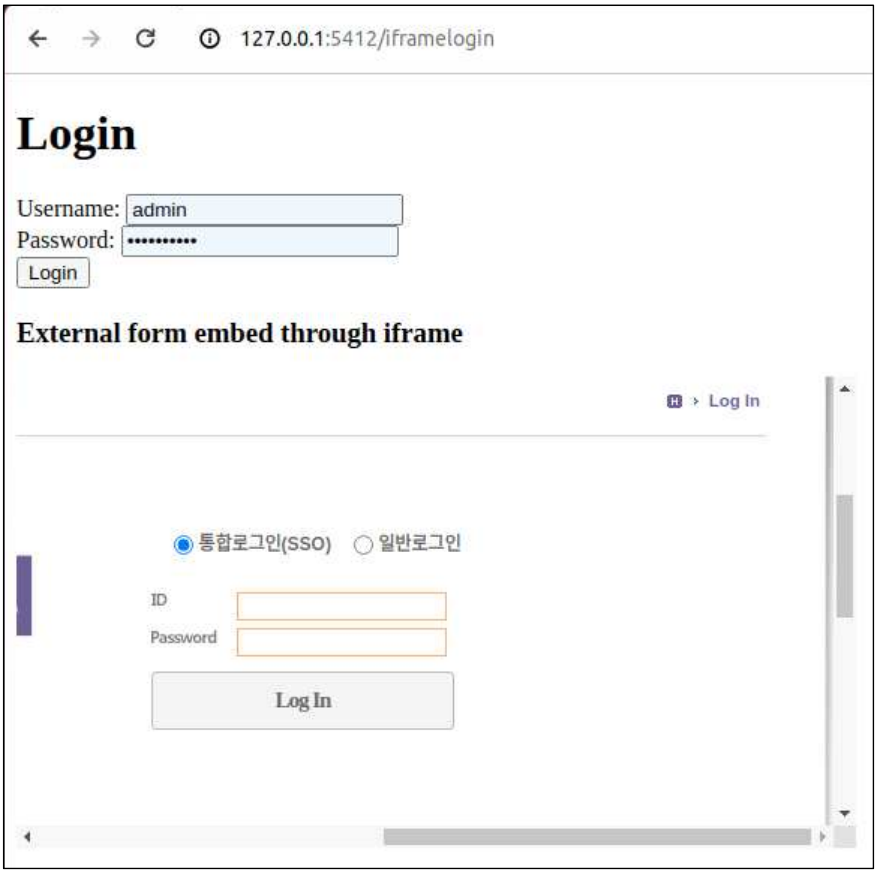

FIGURE 7. iframe을 통해 한 페이지 내에 다른 도메인의 요소를 불러온 결과

각 사이트마다 채택하고 있는 보안 정책이 달라 iframe의 호출을 받는 사이트가 다른 도메인에서 사용할 수 없도록 구현한 경우는 페이지에 배치할 수 없었다. 그러나 보안 정책이 미흡한 사이트의 경우 iframe을 통해 렌더링 할 수 있었으며, 패스워드 매니저는 껍데기 페이지의 계정 정보를 불러오는 것이 아닌 iframe 속의 탈취하고자 하는 사이트의 계정 정보를 입력하는 것을 확인할 수 있었다.

3) DOM 변조 확인 정책

Fig. 8에서 보이는 것처럼 겉에 보이는 페이지 상에서 달라진 점은 확인할 수 없으나, TABLE V에서 보이는 것과 같이 DOM이 변경된 페이지를 로드하여 패스워드 매니저의 반응성 테스트를 진행하였다. 그 결과로 변경 전에 저장한 패스워드를 웹 페이지가 변경되었음에도 자동 입력한 것을 확인하였다.

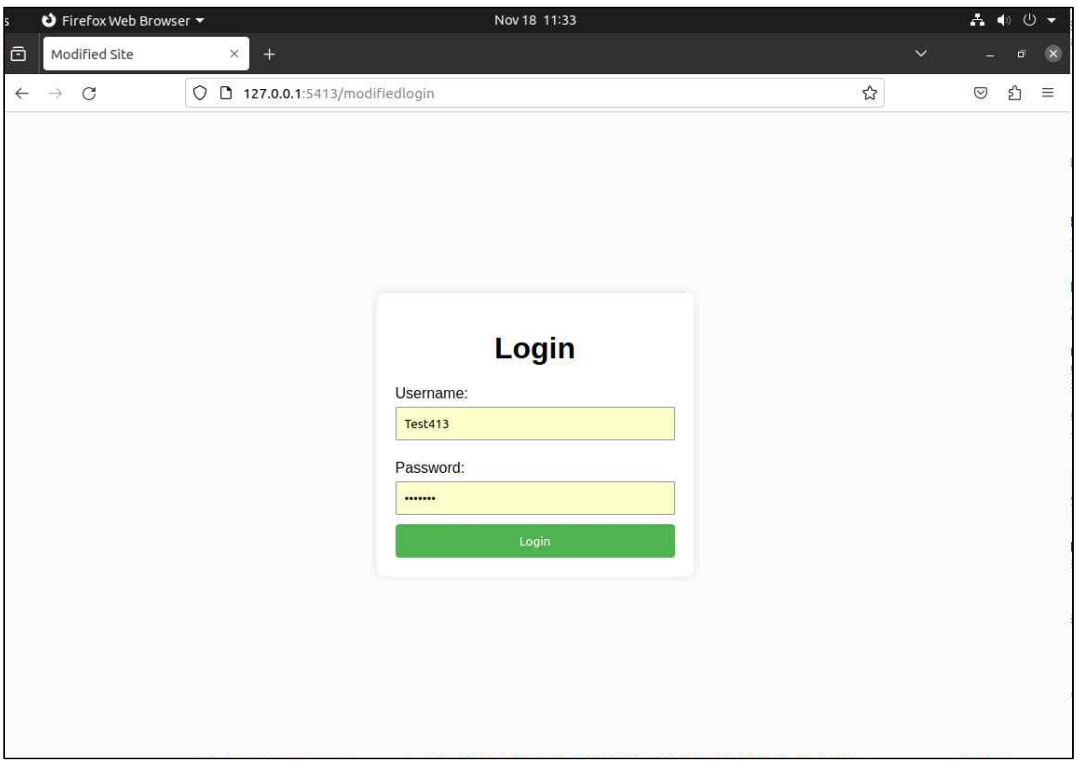

FIGURE 8. 변경된 DOM을 검증하지 않고 자동 입력을 수행한 모습

TABLE V

패스워드 매니저가 저장할 때와 비교하여 달라진 페이지 코드 상세

변경 전 (login.html)	변경 후 (modifiedlogin.html)
<form method="POST" action="/login"> <label for="username">Username:</label> <input type="text" id="username" name="username" required> <label for="password">Password:</label> <input type="password" id="password" name="password" required> <button type="submit">Login</button> </form>	<form method="POST" action="/doublelogin"> <label for="username">Username:</label> <input type="text" id="userid" name="userid" required> <label for="password">Password:</label> <input type="password" id="userpw" name="userpw" required> <button type="submit">Login</button> </form>

4) 다중 Input Field 처리 정책

다중 Input Field에 따른 패스워드 매니저의 처리 정책을 확인하기 위해 일반적인 로그인 페이지와 로그인 페이지에 다른 로그인 페이지가 삽입된 웹 페이지를 구현하였다. 실험 결과의 가독성을 확인하기 위해 Fig. 9에서 보이는 것과 같이 눈에 띄는 색으로 구성하였으나, 실제 공격 상황에서는 CSS를 활용하여 육안으로 확인할 수 없도록 투명도 조절이 가능하다. 실

험 결과에 따르면 패스워드 매니저는 동시에 로그인 페이지와 삽입된 페이지 모두 로그인 정보를 자동 입력하였다.

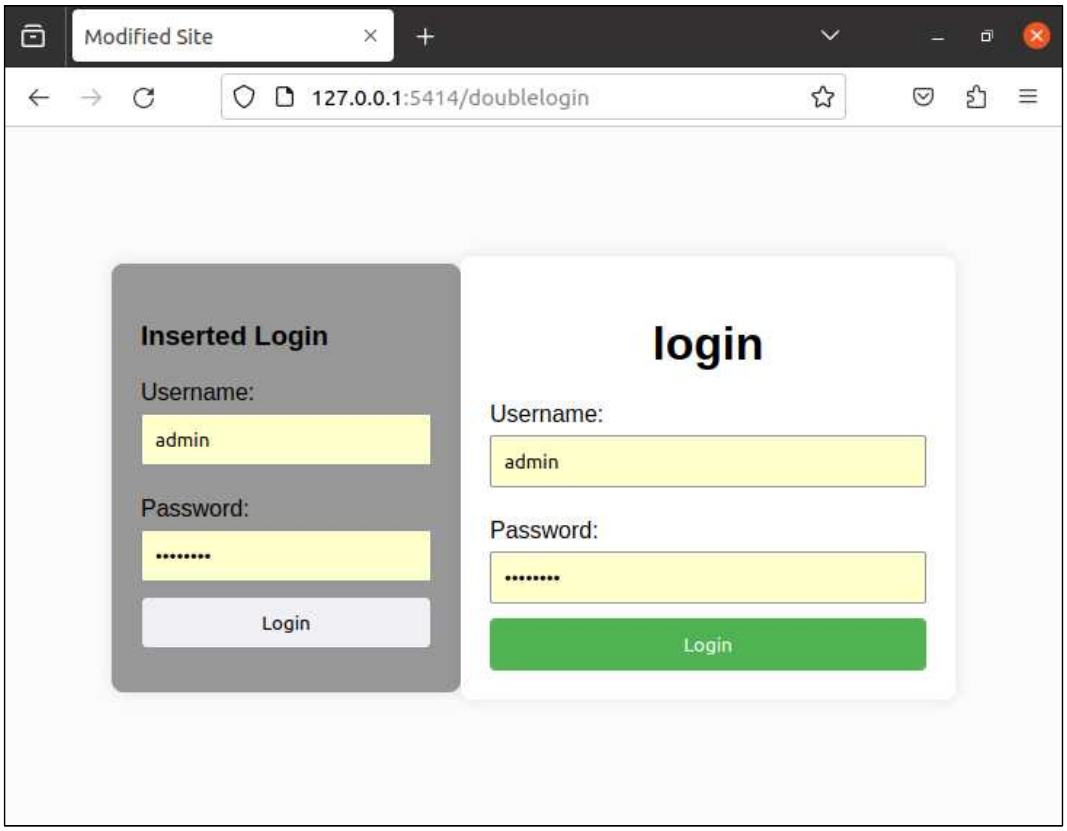

FIGURE 9. 정상 로그인 폼 뿐만 아니라 삽입된 로그인 폼에도 현재 도메인에 저장된 같은 계정 정보를 자동 입력한 결과

5) 비표준 로그인 필드 처리 정책

비표준 로그인 필드에 따른 패스워드 매니저의 자동 입력 반응을 확인하기 위해 일반적인 로그인 페이지와 같이 text와 password 속성을 가진 입력 필드를 Fig. 10과 같은 모습으로 제작하였다. 실험 결과, 일반적인

로그인 페이지에서 저장한 로그인 정보를 유사 로그인 페이지에 자동 입력하는 것을 확인하였다.

FIGURE 10. 정상적인 로그인 폼 형식이 아니더라도 자동 입력을 수행한 결과

2. 취약점 구현

1장에서 분석한 공격 표면으로 실제 보안 취약점 연계 사고를 입증하기 위해 본 연구에서는 취약점 발생 시나리오 세 가지를 구현하였다. 본 장의 실험 환경은 웹 서비스 사용자들이 대부분 사용하는 GUI 환경을 위하여 Windows 10 운영체제에서 수행하였으며, 웹 서비스 플랫폼은 Alexa Top Site와 사용자 다운로드 수를 종합적으로 고려하여 많은 사용자 수가 방문하는 것으로 판단되는 사이트를 대상으로 실험하였다.

1) 메일 서비스를 통한 비표준 로그인 필드 구현

앞서 도출한 공격 표면 중 취약한 URI 식별 정책과 DOM 수정 확인 정책, 비표준 로그인 필드 처리 정책을 메일 서비스와 연계하여 실제 발생할 수 있는 보안 취약점을 증명하였다. 실험 대상 선정 기준으로는 메일 서비스를 제공하는 여러 포털 사이트 중 사용자 수가 500만 명이 넘고, 메일 송신 시 HTML을 첨부할 수 있는 플랫폼으로 10가지를 도출하여 수행하였다. HTML 첨부 기능을 통해 로그인 폼을 유사하게 구현하여 전송하여 패스워드 매니저의 반응을 살폈다. 보내는 측에서는 HTML 작성을 위한 창을 지원하는 메일 서비스의 경우 해당 기능을 통해 로그인 폼을 전송하였고, 공식적으로 지원하지 않은 플랫폼의 경우 Fig. 11과 같이 브라우저의 개발자 도구에서 ‘Edit as HTML’ 기능을 통해 삽입할 수 있었다. Fig. 12는 Google G-Mail 환경에서 보내는 측과 받는 측을 캡쳐한 모습이다. 패스워드 매니저는 메일 서비스를 통해 전송된 비표준 로그인 필드를 로그인 필드로 인식하여 자동 입력을 수행하였으며, 그 화면은 Fig. 13과 같다.

FIGURE 11. 개발자 도구의 Edit as HTML 옵션으로 HTML을 임베드

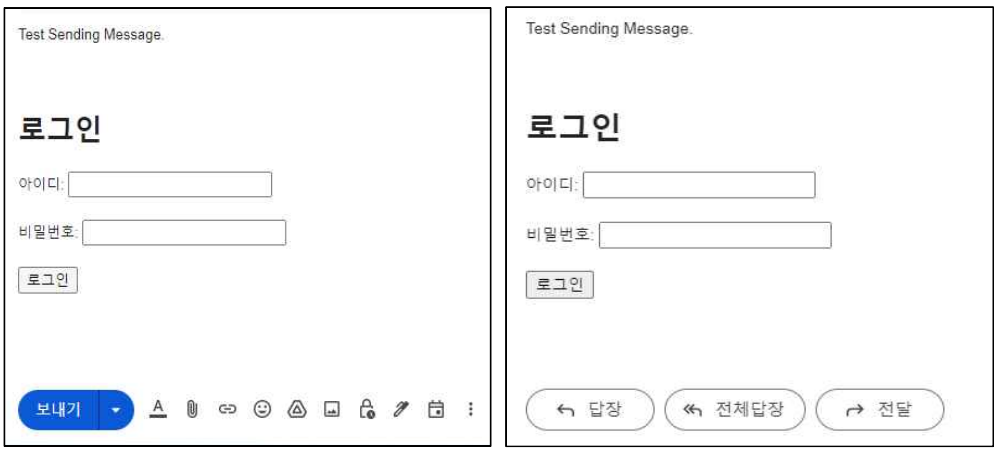

FIGURE 12. HTML을 포함하여 보낸 메일(좌)과 받은 메일(우)

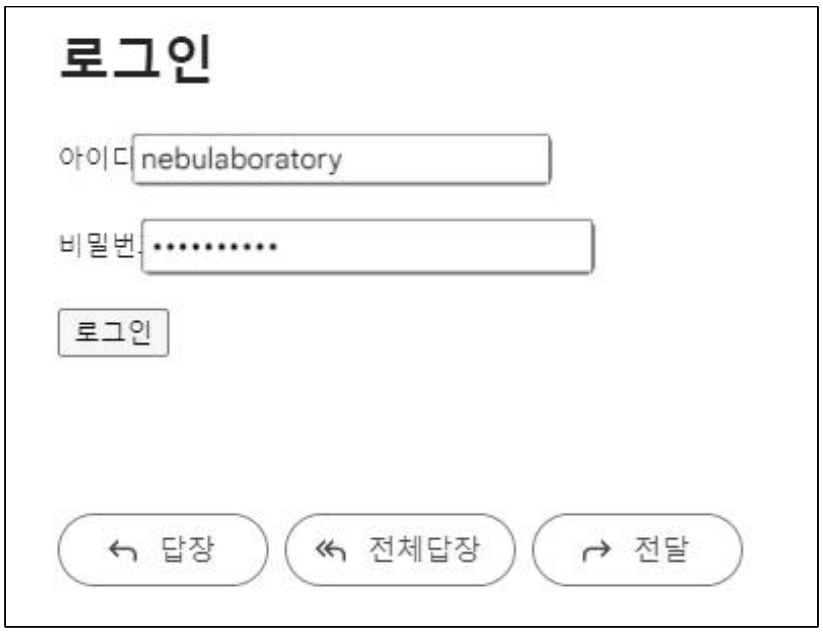

FIGURE 13. 메일 서비스의 비표준 로그인 필드에 자동 입력을 하는 모션

결과적으로, 국내외 주요 메일 서비스인 Naver, Daum/Kakao, G-mail, Outlook, iCloud, Yahoo!-AOL, GMX, Proton, Yandex, mail.ru 총 열 가지를 대상으로 실험을 수행하였고, 결과를 요약한 표는 Table VI에 정리하였다.

삽입된 로그인 폼은 공격자의 서버에 제출되도록 구현하였고 패스워드 매니저가 자동 입력을 한 후 전송되는 경우, 공격자의 서버에 계정 정보가 유출되는 것을 확인하였다. 최종 사용자 계정 탈취까지 성공하면 공격에 성공했다고 간주하고, 자동 입력까지만 수행한 경우는 별도 기록하였다.

TABLE VI

주요 메일 서비스에 연계한 취약점 분석 결과

	플랫폼	자동 입력 여부	비고
1	Naver	가능	
2	Daum/Kakao	가능	
3	G-mail	불가	password manager autofilling
4	Outlook	불가	password manager autofilling
5	iCloud	불가	
6	Yahoo!/AOL	불가	
7	GMX	불가	
8	Proton	불가	
9	Yandex	불가	
10	mail.ru	불가	password manager autofilling

사용자 계정 탈취까지 성공하지는 못하였지만, 패스워드 매니저가 자동 입력을 수행하는 반응을 Table VI 3, 4, 10번의 플랫폼에서 볼 수 있었다. 5-9번의 플랫폼에서 패스워드 매니저가 작동하지 않은 이유는 각 메일 서버에서 보안 취약점 예방을 위해 전송 내용에 HTML 객체를 필터링하는 과정을 거쳐 폼 객체가 갖추어야 할 최소한의 문법을 삭제한다. 그러므로 수신지에 도착한 컨텐츠는 패스워드 매니저가 로그인 폼이라고 인식할 수 없었기 때문으로 확인하였다. 이는 패스워드 매니저는 취약성을 가

지고 있으나 웹 사이트 내에서 보안 조치를 한 것으로 해석할 수 있으며, 이는 패스워드 매니저의 미흡한 보안성에 대한 시사점을 남긴다.

2) 악성 확장 프로그램을 통한 평문 노출

DOM 수정 확인 정책을 확인하기 위해 상용 웹 서비스의 로그인 필드를 조작하여 패스워드 매니저의 반응성 테스트를 수행하였다. 첫 번째로 렌더링 된 페이지의 패스워드 입력란의 input type=password를 input type=text로 수정되도록 확장 프로그램을 구현하였다. 실험 결과, Fig. 14에서 보이는 것처럼 모든 패스워드 매니저가 변경된 필드에 자동 입력하지 않았고, 로그인 필드라고 인식하는 것으로 보기 어려웠다. 이를 통하여 패스워드와 직결되는 속성은 엄격하게 확인한다고 볼 수 있다.

FIGURE 14. 패스워드 입력 속성을 바꾼 후 패스워드 매니저의 반응성 확인

본 항목에 대한 연장 공격으로, 필드에 값이 채워지는 것이 탐지되면

input type이 변경되도록 확장 프로그램을 구현하였다. 웹 페이지가 랜더링되고 브라우저의 확장 프로그램보다 패스워드 매니저가 먼저 작동을 하게 되면 가정 대로 평문 노출이 가능할 것이다. 이를 실제 실험한 결과 Fig 15에서 보이는 것처럼 평문이 노출되는 것을 확인할 수 있었다.

FIGURE 15. 패스워드 입력 속성 변조를 후처리한 결과

필드 위치를 알면 원격 값 추출도 가능하기 때문에 패스워드 매니저의 자동 입력 후 속성이 바뀌는 것은 Fig. 15와 같이 평문이 노출되므로 보안 사고가 발생할 수 있다. 이는 연계 공격 성공을 위해 제안하는 방식으로 1장에서 도출한 공격 표면과도 무관하여 모든 패스워드 매니저가 취약한 것으로 확인하였다. 이러한 타이밍 공격은 패스워드 매니저 차원에서 막기는 어려운 특징으로, 패스워드 매니저 보안 아키텍처 설계에서 중요하게 다루어야 할 시나리오이다.

3) 악성 확장 프로그램을 통한 DOM 변조 구현

세 번째 취약점 시나리오는 악성 확장 프로그램으로 로그인 페이지가 랜더링 될 때 계정 정보가 전송되는 URI를 수정된 상황에서 발생하는 계정 탈취 방법이다. 본 시나리오 구현을 위하여 확장 프로그램의 Content Script를 통해 DOM 요소가 변조되도록 프로그램을 작성하였고, 패스워드 매니저의 반응성 테스트를 진행하였다. 실제 악성 행위가 발생되는 코드는 TABLE VII와 같다.

TABLE VII

구현한 악성 확장 프로그램의 기능 구현부

ContentScript.js

if login in page:
innerHTML.replace('action="기존 URI"', 'action="악성 URI"');
innerHTML.replace('method="POST"', 'method="GET"');

본래 로그인 필드에 입력된 값은 제출 시 인증 서버에 도달하는데, 이를 공격자의 서버로 제출되게 조작하여 반응성 테스트를 진행한 결과 사용자의 계정 정보가 수정된 서버로 전송되어 로그인 정보를 탈취하는데 성공하였다.

1번 메일 서비스를 통한 비표준 로그인 필드 구현 항목과 유사하게 본 항목에서도 로그인 사이트의 플랫폼마다 제출 이벤트가 비활성화 되거나 경고 알림을 노출하는 등 다양한 조치를 보였다. 그러나 이는 웹 서비스 제공자가 제공 서비스의 완결성을 높이기 위해 수행한 보안 조치이며, 패스워드 매니저는 이와 같은 공격에 무방비하게 노출됨을 확인하였다.

3. 실험 결과 분석

변형된 웹 페이지에 발생하는 패스워드 매니저의 취약성과 이 취약성을 연계하여 실제 취약점으로 이어질 수 있도록 구현하여 8개의 패스워드 매니저를 대상으로 반응성 실험을 수행하였다. 표 VIII은 체크할 항목을 정리한 것이다.

TABLE VIII

도출한 공격 표면 및 취약점 시나리오 반응성 평가 지표

	분류	항목 명	자동 입력 여부
1		취약한 URI 식별 정책	
2		다중 도메인에 대한 반응 정책	
3	취약성	DOM 수정 확인 정책	
4		다중 Input Field 처리 정책	
5		비표준 로그인 필드 처리 정책	
6		메일 서비스를 통한 비표준 로그인 필드 구현	
7	연계 공격	악성 확장 프로그램을 통한 평문 노출	
8		악성 확장 프로그램을 통한 DOM 수정 구현	

각 항목에 맞추어 패스워드 매니저의 반응성 실험 결과는 TABLE IX에 요약하였다.

TABLE IX

반응성 평가 결과 요약

	브라우저			확장프로그램				
	Chrome	Firefox	Edge	Last-Pass	Avira	Norton	1Pass-word	Bitwar-den
1	◎	◎	◎	○	◎	○	○	◎
2	×	○	×	○	◎	×	×	×
3	◎	◎	◎	○	◎	○	○	◎
4	◎	◎	◎	○	◎	×	○	◎
5	◎	◎	◎	○	×	○	○	◎
6	○	◎	◎	◎	×	○	○	◎
7	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○

* ◎ : 사용자 상호작용 없이 자동 입력 수행, 취약

** ○ : 사용자 상호작용 후 자동 입력 수행, 취약

*** × : 자동 입력 기능 비활성화, 안전

동그라미 두 개는 사용자의 상호작용 없이도 자동 입력 동작을 수행한 것을 의미하고, 단일 동그라미는 상호작용은 필요했지만 사용자가 요청하였을 때 자동 입력 기능을 의도대로 수행한 것을 의미한다.

1번과 3번 항목은 모든 패스워드 매니저가 자동 입력 동작을 수행하였다. 2번 항목은 연구 대상 절반 이상의 패스워드 매니저가 반응하지 않았다. 이전 연구에서 Chrome을 비롯한 많은 패스워드 매니저가 취약한 반응을 보여 문제점으로 제기되었으나 보안성 조치가 이루어진 것으로 확인

된다.

1Password는 4번 항목에서 삽입된 악성 필드에 상호작용 후 자동 입력 정책을 보였다. 그러나 패스워드 매니저 관리 인터페이스에서 자동 채우기 버튼을 클릭했을 때는 메인 로그인 필드만을 채웠다. 다시 말해서, 기존의 로그인 필드와 새로 추가된 필드를 구분하는 모습을 여덟 가지 패스워드 매니저 중에 유일하게 보였다. Avira 패스워드 매니저는 4번 항목을 테스트할 때 모든 필드에 자동 입력 하지 않았다. 또한 Avira 패스워드 매니저는 5번 비표준 로그인 필드를 처리할 때 유일하게 새로운 로그인 필드로 인식하여 보안성을 신경 쓴 모습을 확인할 수 있었다.

본 실험은 패스워드 자동 입력 정책의 취약성을 파악하기 위한 반응성 평가이므로, 자동 입력 옵션을 설치 후 별도로 설정해주어야 하는 패스워드 매니저는 자동 입력되도록 설정한 후 실험 수행하였다.

V. 패스워드 매니저 보안성 평가 프레임워크

본 장에서는 패스워드 매니저 연구를 메타 분석하고, 본 연구에서 수행한 취약성 평가 결과를 기반으로 패스워드 매니저에 공통의 기준으로 적용할 수 있는 보안성 평가 프레임워크를 제안한다.

1. 보안성 평가 프레임워크

실험을 통해 도출한 항목과 이전 연구를 통해 밝혀진 취약성을 유발하는 기본 패스워드 매니저의 설정 값을 기반으로 패스워드 매니저의 전체적인 보안성 평가를 수행할 수 있는 프레임워크를 제안한다. 최근 10년 동안 수행된 패스워드 매니저 보안성 연구를 메타 분석하여 이전 연구에서 공격 표면으로 식별한 항목은 매핑하였다.

TABLE X

보안성 평가 프레임워크

	Category	Clauses	Refs.
1	자동 입력 취약점	느슨한 URI 식별 정책	[3], [4], [10]
2		다중 도메인에 대한 반응 정책	[1], [4]
3		DOM 수정 확인 정책	[1], [4]
4		다중 Input Field 처리 정책	[3], [11]
5		비표준 로그인 필드 처리 정책	[1]
6	기본 설정	마스터 패스워드 체제 지원	[12]
7		자동 입력 옵션 기본 비활성화 여부	[1], [4], [13]
8		계정 잠금 세션 시간	[13]
9	생성 취약점	생성 패스워드 복잡도	[1], [12]
10		사용자 패스워드 강도 분석 기능 지원 여부	[12], [13]

위 항목 중 실제 취약점 분석 수행 시 직접적으로 계정 탈취에 활용되었던 항목은 가중치를 부여하고, 기본 설정과 생성 취약점 항목에 대해서는 직접적인 기능적 취약성과 직결된다고 보기 어려우므로 낮은 중요도 값을 반영한다. 구체적인 평가 기준은 이어지는 다음 세부 항목과 같다.

① 자동 입력 취약점

점수	평가 기준
3	악의적인 필드에 반응하지 않음
2	악의적인 필드를 구분할 수 있음
1	상호 작용 후 자동 입력
0	자동 입력

패스워드 매니저의 자동 입력 기능의 보안성 강도를 측정하기 위한 항목으로, 악의적으로 생성하거나 변조된 필드에 반응하지 않은 패스워드 매니저에는 보안성 점수 3점을 부여하였고, 상황에 따라 자동 입력을 수행하긴 하나 구분하는 경우는 보안성 점수 2점을 부여하였다. 그리고 프로그램 사용자가 상호 작용을 수행한 후 자동 입력하는 경우 보안성 점수 1점을 부여하였고, 웹 페이지가 랜더링 되자마자 자동 입력하는 경우 0점을 부여하였다.

② 기본 설정

점수	평가 기준
1	기능 지원
0	기능 미지원

패스워드 매니저의 보안성에 영향을 미치는 정책에 대한 항목으로, 각 정책이 기능적으로 지원하는 경우 보안성 점수 1점을 부여하였고, 기능적으로 지원하지 않아 프로그램 사용자가 보안 강도를 조절할 수 없는 경우 0점을 부여하였다.

③ 생성 취약점

점수	평가 기준
1	기능 지원 및 숫자, 문자, 특수문자 등 적합한 복잡도 구성
0	기능 미지원

패스워드 매니저의 생성 및 저장 기능에서 지원하는 기능에 대한 항목으로, 기능을 적합한 복잡도로 구현하는 경우 보안성 점수 1점을 부여하였고, 그렇지 않은 경우 0점을 부여하였다.

상기 10개의 항목에서 모두 최고점을 받게 되면 20점이 책정된다.

2. 상용 패스워드 매니저 평가

도출한 보안성 평가 프레임워크를 적용하여 연구 대상인 8개의 패스워드 매니저를 평가하였다. 결과는 다음 TABLE XI와 같다.

TABLE XI

보안성 평가 프레임워크 적용 결과

	항목	점수							
		1	2	3	4	5	6	7	8
1	느슨한 URI 식별 정책	0	0	0	1	0	1	1	0
2	다중 도메인에 대한 반응 정책	3	1	3	1	0	3	3	3
3	DOM 수정 확인 정책	0	0	0	1	0	1	1	0
4	다중 Input Field 처리 정책	0	0	0	1	0	3	2	0
5	비표준 로그인 필드 처리 정책	0	0	0	1	3	1	1	0
6	마스터 패스워드 체제 지원	0	0	0	1	1	1	1	1
7	자동 입력 옵션 기본 비활성화 여부	0	0	0	0	0	1	1	1
8	계정 잠금 세션 시간	0	0	0	0	1	1	1	1
9	생성 패스워드 복잡도	1	0	1	1	1	1	1	1
10	사용자 패스워드 강도 분석 기능 지원 여부	1	0	0	1	1	1	1	1
	총점	5	1	4	8	7	14	13	8
	순위	6	8	7	3	5	1	2	3

* 2행은 순서대로 Chrome, Firefox, Edge, LastPass, Avira, Norton, 1Password, Bitwarden을 뜻함.

브라우저에 기본 내재된 패스워드 매니저 3종은 전반적으로 보안성이 아주 낮은 것으로 평가되었으며 그중 Firefox가 특히 취약한 것으로 나타났다. 확장 프로그램 기반의 패스워드 매니저는 Norton 패스워드 매니저가 가장 높은 점수를 기록하였고 1Password가 뒤따른다. 그러나 모든 패스워드 매니저가 아직 계정 탈취에 대한 취약성을 내재하고 있어 보안성 조치가 필요하다.

Ⅵ. 보안성 강화 모델 제안

1. 보안 요구 사항

본 연구에서 제안한 보안성 평가 프레임워크의 자동 입력 공격 표면과 실제 취약점 실현 가능성을 보인 악성 확장 프로그램을 통한 공모 공격을 바탕으로 패스워드 매니저가 갖추어야 할 보안 요구 사항을 도출하였다. Google과 같은 브라우저 제공자들은 확장 프로그램을 웹 스토어에 등록할 때 뿐 아니라 사후 조치를 엄격하게 수행하여 악성 확장 프로그램으로 보이는 것을 추적하고 삭제하고 있다. 그러나 본 연구에서 제안한 확장 프로그램을 통한 취약점 공격은 확장 프로그램이 기본적인 기능 구현을 위해 가지고 있는 최소 권한과 특성을 악용하여 공격 기법을 구상하고 성공할 수 있음을 보였다. 따라서 확장 프로그램 검수에 의존하는 것이 아닌 패스워드 매니저의 보안성 패치를 통해 원천적인 위협 요인을 축소해야 한다.

본 연구에서 제안한 기법 외에도 이전 연구에서 공개된 대부분의 자동 입력 취약점은 변형된 웹 페이지에 따라 발현되는 경향을 보인다. 따라서 웹 페이지가 공격자에 의해 훼손되었는지 확인하는 기능이 패스워드 매니저에 탑재된다면 공격 성공률을 크게 낮춰줄 것으로 예상된다.

자동 입력 정책의 경우 기본 설정의 여부는 크게 중요하지 않다. 사용자들은 편의성을 위해 기본 옵션이 아니더라도 활성화 하여 사용할 것이기 때문에 자동 입력 정책을 이전보다 엄격하게 설계하는 것이 필요하다.

패스워드 매니저의 보안성 강화를 위한 보안 요구사항을 정리하면 다음과 같다.

1) 로그인 페이지의 무결성 검증 기능을 탑재

확장 프로그램은 연결된 브라우저의 API(Application Programming Interface)를 사용할 수 있는 권한이 있다. 이를 통하여 로그인 이벤트 수행 시 새 탭을 실행시켜 로그인을 시도하는 페이지가 악성 공격자에 의해 변조되지 않았는지 무결성을 검증할 수 있다. 이는 DOM 변조와 비표준 로그인 필드 공격 표면에 대응할 수 있는 보안 요구사항이다.

2) 계정 정보 저장 시 필드 정보 기억

최초 로그인을 수행하여 패스워드 매니저에 계정 정보를 저장할 때 입력 값이 위치한 필드의 식별 정보를 함께 저장하는 것이 필요하다. 이를 통하여 다중 Input Field가 존재하는 페이지라고 하더라도 올바른 위치에 자동 입력 기능을 수행할 수 있다.

3) 저장된 URI와 다른 하위 페이지에 대한 정책 수정

패스워드 매니저가 사용자의 계정 정보를 최초 저장할 때 함께 기록되는 URI 정보를 엄격하게 구분하여야 한다. 같은 도메인 이름을 가지더라도 여러 개의 로그인 필드를 가질 수 있는데, 패스워드 매니저는 이러한 페이지 구조에 취약함을 보였다. 큰 규모의 사이트는 수많은 하위 페이지로 이루어져 있으나 모든 페이지에 강도 높은 보안성 조치가 되어 있다고 보기 어렵다. 따라서 도메인에 의존한 종래의 판단 정책을 수정하여 취약한 URI 식별 정책, 다중 도메인에 대한 반응 정책에 발생하는 공격 표면에 대응해야 한다.

2. 제안하는 보안성이 강화된 패스워드 매니저

본 장에서는 1장에서 도출한 보안 요구사항을 구현할 수 있는 방안을 서술하며, 이를 기반으로 향후 보안성을 강화한 패스워드 매니저 개발에 기여할 수 있는 아키텍처를 설계한다.

첫 번째 '로그인 페이지의 무결성 검증 기능을 탑재' 요구 사항은 현재 페이지와 상용 페이지의 DOM 일치 여부를 확인하는 방안으로 충족할 수 있다. 이 기능을 구현하기 위해서 새 탭에서 로그인 페이지의 Form 객체 전문을 해싱(hashing)하여 다이제스트를 도출하고, 현재 로그인하려는 페이지의 Form 객체의 다이제스트와 비교하여 DOM 변조 여부를 검출하는 방법을 제안한다. 제안하는 방법은 브라우저 API로 사용자가 접하고 있는 웹 페이지와 격리된 공간에서 새 탭을 실행시키므로 DOM 변조 악성 행위에 영향을 받지 않는다. 그리고 확장 프로그램의 권한을 사용하므로 다른 응용 애플리케이션을 설치하지 않고 패스워드 매니저에 부가 기능으로 구현할 수 있다.

두 번째 '계정 정보 저장 시 필드 정보 기억' 요구 사항은 현재의 패스워드 매니저의 저장 체계를 보다 견고하게 보완하는 방식으로 구현할 수 있다. 현재 패스워드 매니저는 Fig. 2와 같이 프로그램 사용자가 입력한 아이디, 패스워드 정보와 URL 세 가지만 저장한다. 이 종래의 정책에 더하여 input 객체의 id와 name, class 정보도 함께 저장하여 다른 Form 객체에 자동 입력하는 보안 문제를 예방할 수 있다.

세 번째 '저장된 URI와 다른 하위 페이지에 대한 정책 수정' 요구 사항은 패스워드 매니저의 저장 및 자동 입력 체계를 보완하는 방식으로 구현할 수 있다. 종래 저장 및 자동 입력 체계는 계정 정보를 저장하는 페이지의 URI를 상세히 기록한다고 하더라도, 로그인 페이지를 식별할 때 스

킴(Scheme), 호스트(Host), 경로(Path)를 엄격하게 구분하지 않는다. 이러한 로직에 따른 취약성을 낮추기 위해서는 계정 정보 저장 시 폼 제출이 수행되는 URI를 하위 페이지까지 정확하게 기록하고, 같은 도메인의 다른 하위 사이트라면 추가적인 검증을 수행하여 부가적으로 저장하도록 저장 및 자동 입력 정책을 수정하여야 한다.

위 보안 요구 사항에 대응하는 패스워드 매니저 보안 아키텍처 모식도는 Fig. 16과 같다. 패스워드 매니저가 자동 입력을 수행한 후 제출 이벤트를 발생시켰을 때 패스워드 매니저는 개별 인터넷 프로세스를 실행시켜서 현재 제출하려는 페이지와 실제 돌아가는 상용 서비스의 웹 페이지 DOM을 비교하여 무결성 체크를 완료한 후에 제출한다. 이는 공격자에 의해 실시간으로 웹 페이지 내 DOM이 수정된 상황이나 악성 사이트에 유인된 뒤 계정 정보를 입력하는 피해 상황에 대응할 수 있다.

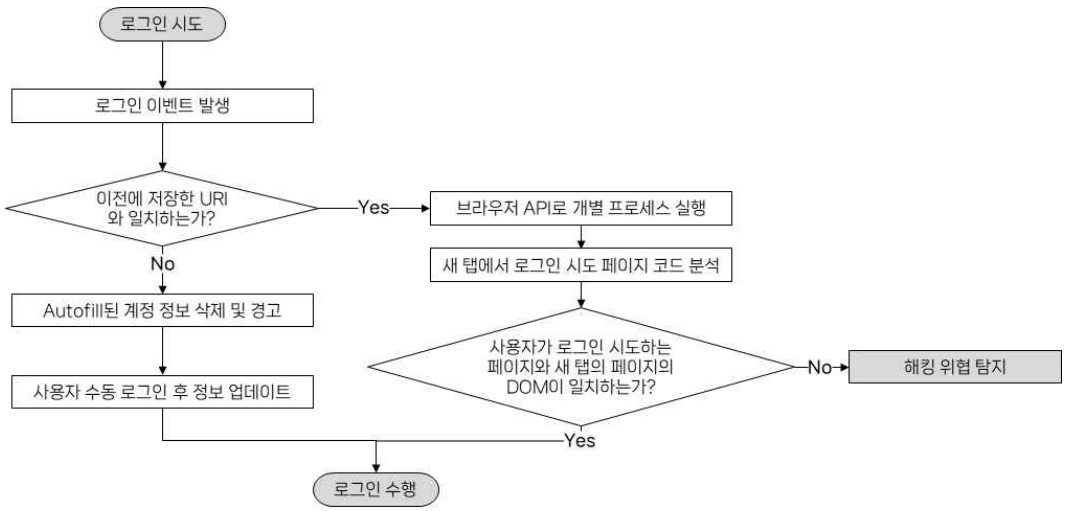

FIGURE 16. 제안하는 보안성 강화 패스워드 매니저 아키텍처

Ⅶ. 결론

1. 논의 및 향후 연구

본 연구를 통해 종래 발견되었던 보안 취약점의 사후 조치 여부를 확인할 수 있었다. 북마클릿의 형태로 제공되는 패스워드 매니저의 취약점은 오래전부터 제기되어왔던 고질적인 문제점을 가지고 있었으나 보안 취약점의 파급력이 큼에 따라 현재는 북마클릿 형식의 패스워드 매니저는 지원하지 않게 되었다. 그러므로 해당 공격 표면으로는 취약점이 발생하지 않게 됨을 확인할 수 있었다. 반면에 이전 연구에서 현재 유효하지 않은 것으로 결론지었던 공격 벡터는 패스워드 매니저의 보안성 강화가 아닌 웹 브라우저 및 사이트의 보안 정책 강화로 무력화된 것이므로 본질적인 문제가 해결되었다고 보기 어렵다. 따라서 패스워드 매니저의 취약한 정책을 보완할 필요가 있다.

패스워드 매니저의 자동 입력 정책이 종래 공개되었던 취약성과 비교하여 보안성이 강화되고 있음을 실험을 통해 확인하였다. 그러나 패스워드 매니저의 자동 입력 기능 사용에 있어 여전히 공격 표면이 다양한 방법으로 발생하고 있으며, 본 연구에서 공개한 취약점에 대응할 수 있도록 보안 조치가 필요하다.

실험 과정에서 LastPass 패스워드 매니저의 자동 입력 정책이 유료 지원 버전과 무료 지원 버전 간의 보안 강도 차이를 발견하였다. 이는 보안성 조치가 현실적으로 어려운 지점이 아님을 시사한다.

본 연구에서는 악성 확장 프로그램을 통해 계정을 탈취하는 시나리오를

제안하였다. 이를 예방하기 위해서는 확장 프로그램을 등록하거나 업데이트 하는 절차를 엄격하게 관리해야 하며, 사용자 또한 브라우저 제공사에 의존하는 것이 아니라 확장 프로그램을 다운로드 받을 때는 어떤 권한을 같이 요구하는지 확인하는 자세가 필요하다.

본 연구에서는 데스크톱 환경에서 사용되는 브라우저 기본 내재형과 확장 프로그램형 패스워드 매니저를 대상으로 취약점을 분석하고 보안성 평가 프레임워크를 제안하였다. 이에 그치지 않고 본 연구에서 다루지 않은 설치형 패스워드 매니저와 모바일 패스워드 매니저에도 적용할 수 있는 보안성 평가 프레임워크가 필요하다. 본 연구 수행 방식과 같이 각 서비스의 보편적인 특성을 도출하여 취약점 분석 평가를 수행하는 연구 방법론으로 향후 연구를 수행할 수 있을 것으로 예상된다.

클라우드 시대가 도래함에 따라 패스워드 매니저의 저장소 또한 원격 클라우드에 저장되는 방식을 채택하고 있는 추이이므로 이 클라우드 통신 상의 취약성에 관한 연구도 주요 과제이다.

2. 결론

본 연구에서는 현대 패스워드 매니저의 보안성 강화를 위해 취약점 분석을 수행하였고 보안성 평가 프레임워크를 제안하였다. 취약점 분석 결과 종래에 공개되었던 보안 취약점이 여전히 유효함을 보였고, 구상한 연계 공격을 통해 내재된 취약점으로 실제 계정 정보 탈취 사고가 발생할 수 있음을 보였다. 제안하는 보안성 평가 프레임워크로 상용 패스워드 매니저를 평가하였으며, 그 결과로 브라우저 기본 내장형 패스워드 매니저가 특히 보안성 수준이 낮고, 확장 프로그램형 패스워드 매니저는 비교적 보안 조치가 수행되고 있으나 여전히 취약하다고 분석하였다. 끝으로 향후 인터넷 환경에서 안전한 패스워드 관리를 할 수 있는 보안 요구사항을 도출하였으며, 이를 통하여 보안성 강화에 이바지할 수 있기를 기대한다.

참고문헌

[1] Oesch, S., & Ruoti, S. (2020, August). That was then, this is now: A security evaluation of password generation, storage, and autofill in browser-based password managers. In Proceedings of the 29th USENIX Conference on Security Symposium (pp. 2165-2182).

[2] Pearman, S., Zhang, S. A., Bauer, L., Christin, N., & Cranor, L. F. (2019). Why people (don't) use password managers effectively. In Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019) (pp. 319-338).

[3] Huaman, N., Amft, S., Oltrogge, M., Acar, Y., & Fahl, S. (2021, May). They would do better if they worked together: The case of interaction problems between password managers and websites. In 2021 IEEE Symposium on Security and Privacy (SP) (pp. 1367-1381). IEEE.

[4] Silver, D., Jana, S., Boneh, D., Chen, E., & Jackson, C. (2014). Password managers: Attacks and defenses. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 449-464).

[5] Li, Z., He, W., Akhawe, D., & Song, D. (2014). The {Emperor's} new password manager: Security analysis of web-based password managers. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 465-479).

[6] Nam, G., Seok, B., Gong, S., Kim, Y., & Lee C. (2018). Method of hijacking a user account using a password manager vulnerability. Journal of Digital Forensics, 12(1), 9-18. https://doi.org/10.22798/KDFS.2018.12.1.9

[7] Jeong, H., & So, J. (2018). Security of Password Vaults of Password Managers. Journal of the Korea Institute of Information Security & Cryptology, 28(5), 1047-1057. https://doi.org/10.13089/JKIISC.2018.28.5.1047

[8] Hong, S., So, J., & Jeong, H. (2020). Security Vulnerabilities of Client-S

erver Communications of Password Managers. Journal of the Korea Institute of Information Security & Cryptology, 30(1), 17-27. https://doi.org/10.13089/JKIISC.2020.30.1.17

[9] LEE, S., & Park, J. (2021). A Study on Acquisition of Google Password Manager Data in Chrome Browser. Journal of Digital Forensics, 15(2), 148-159. https://doi.org/10.22798/KDFS.2021.15.2.12

[10] Carr, M., & Shahandashti, S. F. (2020). Revisiting security vulnerabilities in commercial password managers. In ICT Systems Security and Privacy Protection: 35th IFIP TC 11 International Conference, SEC 2020, Maribor, Slovenia, September 21-23, 2020, Proceedings 35 (pp. 265-279). Springer International Publishing.

[11] Stajano, F., Spencer, M., Jenkinson, G., & Stafford-Fraser, Q. (2015). Password-manager friendly (PMF): Semantic annotations to improve the effectiveness of password managers. In Technology and Practice of Passwords: International Conference on Passwords, PASSWORDS'14, Trondheim, Norway, December 8-10, 2014, Revised Selected Papers 7 (pp. 61-73). Springer International Publishing.

[12] Arias-Cabarcos, P., Marín, A., Palacios, D., Almenárez, F., & Díaz-Sánchez, D. (2016). Comparing password management software: toward usable and secure enterprise authentication. IT Professional, 18(5), 34-40.

[13] Simmons, J., Diallo, O., Oesch, S., & Ruoti, S. (2021, December). Systematization of password manageruse cases and design paradigms. In Annual Computer Security Applications Conference (pp. 528-540).

ABSTRACT

Vulnerability Analysis and Security Assessment Framework of Password Managers

Jiwon Jang
Department of Future Convergence
Technology Engineering
Graduate School of Sungshin University

With the rapid advancement of high-speed information and communication technologies, Internet users find themselves required to manage numerous accounts acquired through membership registrations while utilizing various web services. To efficiently manage the complexity of diverse accounts, users have increasingly turned to the use of password managers. A password manager is software designed to generate and store account information. It offers users both security and convenience by enabling easy retrieval of information when needed, either by generating complex passwords or storing them separately for each site. Despite being widely adopted, there is a lack of recent research in comparison to the high utilization of password managers, particularly in addressing this concern for enhanced security. Therefore, in this study, a comprehensive analysis of the functionalities of web-based password managers used in desktop environments is conducted. The aim is to analyze vulnerabilities that may arise from the characteristics of password managers and demonstrate the potential for account information theft attacks in commercial web service environments. Furthermore, based on experimental results, a password manager security assessment

framework is established to evaluate vulnerabilities. Using the proposed password manager security assessment framework allows us to analyze security vulnerabilities in commercial password managers and identify necessary countermeasures.