소프트웨어 지적 재산권 보호 기술의 역공학 취약점 분석: B 소프트웨어를 기반으로

정원태*, 이경률*

*대구가톨릭대학교 컴퓨터소프트웨어학부

Vulnerability Analysis of Intellectual Property Rights using Reverse Engineering: Based on B Software

Wontae Jung*, Kyungroul Lee*

*School of Computer Software, Daegu Catholic University

요 약

소프트웨어의 불법적인 사용을 예방하기 위하여, 다양한 소프트웨어 지적 재산권 보호 기술이 등장하였지만, 역난독화 및 역공학과 같은 기술을 악용함으로써 소프트웨어의 지적 재산권을 보호받지 못하는 한계점이 존재한다. 이러한 한계점이 가지는 근본적인 원인을 규명하기 위하여, 소프트웨어의 불법적인 사용을 방지하는 지적 재산권 보호 기술 중 라이선스 인증 기술이 적용된 B 소프트웨어를 중점으로, 발생 가능한 취약점을 분석한다. 분석 결과, 단순히 인증 결과를 조작함으로써 라이선스 인증을 우회하는 취약점은 파일 개수를 제한하는 기능을 사용하지 못함으로써 실질적인 취약점으로 판단할 수 없지만, 분석한 파일 개수와 비교 명령어를 기반으로 파일 개수를 조작한 결과, 라이선스를 인증하지 않더라도 B 소프트웨어가 제한하는 기능을 우회하는 취약점을 실증하였다. 본 논문에서 규명한 라이선스 인증 기술이 가지는 근본적인 취약점을 제거한다면, 더욱 효과적으로 소프트웨어의 지적 재산권을 보호할 수 있을 것으로 판단된다.

키워드: 소프트웨어 저작권 보호 기술, 취약점 분석, 역공학, 라이선스 인증

I. 서론

사용자가 컴퓨터 하드웨어를 더욱 편리하게 사용하기 위하여, 프리웨어, 애드웨어, 셰어웨어와 같은 다양한 종류의 소프트웨어가 등장하였다 [9]. 그중, 셰어웨어는 사용자가 정당한 비용을 지불하여야만 소프트웨어가 제공하는 모든 기능을 사용할 수 있다. 하지만 많은 사용자들은 비용을 지불하지 않고 소프트웨어를 사용하기를 원하였으며, 이로 인하여 소프트웨어 불법 복제가 등장하였고 지금까지 지속적으로 악용되는 실정이다.

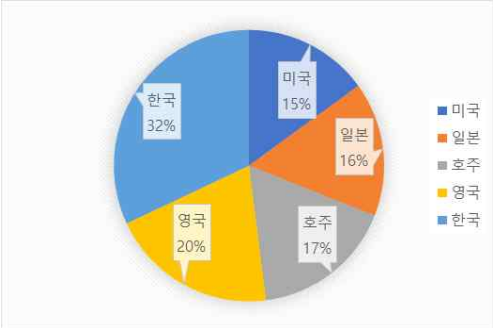
<그림 1>과 같이 국내의 소프트웨어 불법 복제 비율은 32%이며, 세계적으로는 평균 약 37%로 높은 수치를 가진다. 이와 같은 불법 복제로 인하여, 소프트웨어 업계들은 약 7,200억 원의 경제적 손실이 발생하는 심각한 문제점이 발생한다고 보고되었다 [1, 2].

소프트웨어의 불법적인 사용을 예방하기 위하여, 라이선스 인증, 소스코드 난독화, 그리고 역공학 방지, 템퍼링 방지, Anti-Piracy 도구들을 활용함으로써 소프트웨어의 지적 재산권을 보호하기 시작하였다.

그럼에도 불구하고, 역난독화 및 역공학과 같은 기술을 악용함으로써 소프트웨어의 지적 재산권을 보호받지 못하는 한계점이 존재한다. 따라서 이러한 한계점이 가지는 근본적인 원인을 규명하기 위하여, 현재 적용된 기술이 가지는 취약점 분석하는 연구가 요구되며, 분석 결과를 기반으로 더욱 효과적으로 소프트웨어의 지적 재산권을 보호할 수 있을 것으로 판단된다. 이러한 요구를 만족하기 위하여, 본 논문에서는 소프트웨어의 불법적인 사용을 방지하는 지적 재산권 보호 기술 중 라이선스 인증 기술이 적용된 B 소프트웨어를 중점으로, 해당 기술

의 취약점을 분석하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 소프트웨어 지적 재산권 보호 기술을 소개하고, 3장에서는 B 소프트웨어의 지적 재산권 보호 기술 및 취약점 분석 결과를 서술하며, 결론 및 향후 연구를 4장에 나타내었다.



<그림 1> 2017년 소프트웨어 불법 복제 비율

II. 관련 연구

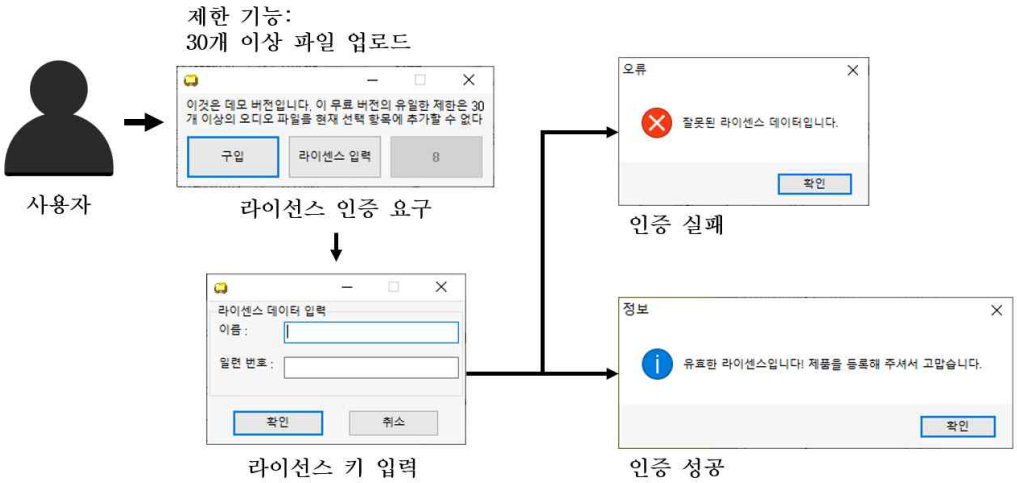
소프트웨어 지적 재산권 보호 기술은 라이선스 인증, 소스코드 난독화, 불법 복제 방지 기술이 있다. 라이선스 인증 기술은 저작권자로부터 소프트웨어의 사용을 허락받는 것으로, 인증을 위하여 문자열 형태의 유일한 라이선스 키를 활용한다 [3]. 소프트웨어 난독화 기술은 소스코드의 추가 및 수정을 통하여 분석가로 하여금 분석을 방해하거나 시간이 많이 소요되도록 가독성을 낮추는 기술이다 [4, 5]. 불법 복제 방지 기술은 역공학 방지, 템퍼링 방지, Anti-Piracy를 지원하는 도구들을 사용하여 프로그램의 분석 및 복제를 방지하는 기술이다 [7].

상기와 같이 소프트웨어의 지적 재산권을 보호하기 위한 다양한 기술이 존재하며, 본 논문의 분석 대상인 B 소프트웨어는 라이선스 인증 기술을 사용한다. 이 기술에서 인증과 관련된 핵심 정보는 문자열 형태의 유일한 라이선스 키이며, 키의 노출 및 탈취, 인증 우회와 같은 취약점을 집중적으로 분석함으로써 지적 재산권을 보호받지 못하는 근본적인 원인을 분석하고자 한다.

III. 분석 결과

3.1. B 소프트웨어 지적 재산권 보호 기술 분석

B 소프트웨어는 라이선스 인증 기술을 기반으로 지적 재산권을 보호하며, <그림 2>와 같이 라이선스 인증 과정을 통하여 소프트웨어의 사용을 허가받는다. 라이선스를 인증받지 못하면, 30개를 초과하는 파일을 편집하지 못하며, 라이선스 인증 요구 메시지가 지속적으로 출력된다. 즉, B 소프트웨어는 라이선스 인증을 통하여 30개를 초과하는 파일을 편집할 수 있도록 사용을 허가받는다. 따라서 라이선스를 구매하여 올바른 라이선스 키를 입력한 사용자만 B 소프트웨어가 제공하는 모든 기능을 사용할 권한을 부여받는다.



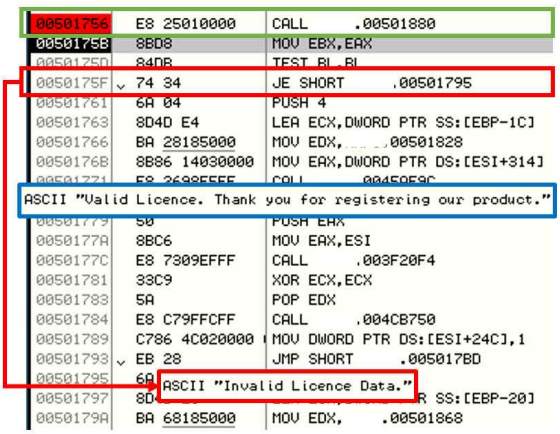
<그림 2> B 소프트웨어 라이선스 인증과정

하지만, 이처럼 문자열을 비교하는 인증은 구조적으로 올바른 라이선스 키와 입력한 라이선스 키를 비교할 수밖에 없으며, 비교 결과를 기반으로, 인증 성공 및 실패 메시지를 출력한다. 이 과정에서 인증 결과를 조작한다면, 인증에 실패하더라도 인증에 성공한 사용자로 위장이 가능한 취약점이 존재한다.

3.2. 취약점 분석 결과

상기 가정한 취약점을 기반으로 B 소프트웨어의 라이선스 인증 우회를 시도하였다. B 소프트웨어는 라이선스 인증을 위하여, <그림 2>와 같이 사용자의 이름과 라이선스 키에 해당하는 일련번호를 입력받는다. 입력된 일련번호는 0x00501756번지의 call에서 비교하며, 인증 결과에 따라, 0x0050175F번지에서 코드의 흐름이 달라진다.

<그림 3>에 표시된 문자열과 같이 키가 올바른 경우에는 0x00501761번지의 코드를 실행하며, 올바르지 않은 경우에는 0x00501795번지로 분기한다. 이는 코드상에 노출된 문자열인 “Valid Licence. Thank you for registering our product.”와 “Invalid Licence Data.”를 통하여 쉽게 코드를 분석할 수 있다.



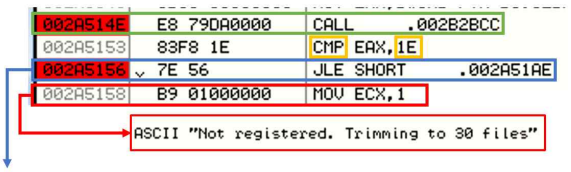
<그림 3> 라이선스 인증 함수 및 인증 결과 코드 일부

따라서 강제적으로 올바른 라이선스를 인증하는 주소인 0x00501761번지를 실행하도록 0x0050175F번지에서 실행 흐름을 조작한다면, 올바르지 않은 비밀번호를 입력하더라도, 라이선스를 정상적으로 인증하였을 때 출력되는 문자열인 “Valid Licence. Thank you for registering our product.”가 출력된다.

상기 코드 분석을 통하여, 라이선스 인증의 우회가 가능하지만, 인증을 우회하더라도, 30개를 초과하는 파일을 편집할 수 있는 제한 기능을 사용하지 못하는 한계점이 존재한다.

B 소프트웨어에서 제한하는 기능을 우회하기 위하여, 라이선스를 인증하지 않더라도, 30개를 초과하는 파일을 편집하는 취약점을 분석하였다. B 소프트웨어가 특정 코드에서 편집을 원하는 파일의 개수를 비교함으로써 기능을 제한하는 것으로 가정하였고, 10진수 30과 비교 명령어를 포함하는 코드를 중점적으로 분석하였다.

분석 결과, <그림 4>와 같이 특정 함수 호출 후, 10진수 30에 해당하는 0x1E와 비교하는 코드를 확인하였고, 비교 결과에 따라, 실행 흐름이 달라진다. 다시 말하면, 0x002A514E번지의 call은 업로드한 파일의 개수를 계산하여 EAX에 저장하며, EAX에 저장된 파일의 개수와 0x1E (30) 을 비교하여 작을 경우, 0x002A51AE번지로 분기하여 파일 업로드 기능을 정상적으로 실행한다. 하지만, 업로드한 파일의 개수가 30보다 큰 경우에는 0x002A5158번지를 실행하여 라이선스가 등록되지 않았다는 문자열인 “Not registered. Trimming to 30 files”를 참조하였다.

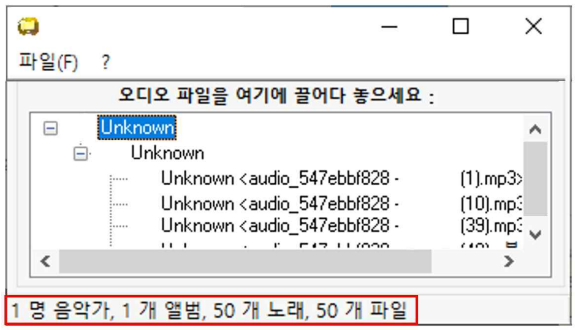


파일 업로드 기능 실행

<그림 4> 업로드하는 파일의 개수를 비교하는 코드 일부

따라서 0x002A5153번지의 0x1E를 30보다 더 큰 수, 예를 들면, 50으로 변경한다면, 제한 기능을 우회하여 50개의 파일을 업로드하고 편집할 수 있을 것으로 가정하였다. 이러한 가정

을 검증하기 위하여, 총 50개의 파일을 업로드한 후, 해당 코드의 0x1E를 0x32 (50) 로 수정하였으며, 그 결과, 그림 5와 같이 라이선스를 인증하지 않더라도 B 소프트웨어가 제한하는 기능을 우회하여 50개의 파일이 편집 가능한 취약점을 실증하였다.



<그림 5> B 소프트웨어 제한 기능 우회 결과

IV. 결론

소프트웨어의 지적 재산권을 보호하기 위한 다양한 기술들이 등장하였지만, 역난독화 및 역공학과 같은 기술을 악용함으로써 소프트웨어의 지적 재산권을 보호받지 못하는 한계점이 존재한다. 이러한 한계점이 가지는 근본적인 원인을 규명하기 위하여, 본 논문에서는 라이선스 인증 기술이 적용된 B 소프트웨어를 대상으로 취약점을 분석하였다.

분석 결과, 라이선스 인증 결과에 출력되는 문자열과 라이선스 키인 일련 정보를 비교하는 코드를 분석함으로써 인증을 우회하였지만, B 소프트웨어에 적용된 제한 기능을 사용하지 못하는 한계점이 존재하였다. 이를 우회하기 위하여, 파일 개수와 비교 명령어를 기반으로 제한 기능을 우회함으로써 라이선스 인증 기술을 무력화하는 것을 실증하였다.

본 논문에서 규명한 라이선스 인증 기술이 가지는 근본적인 취약점을 기반으로, 제한 기능과 관련된 정보를 하드코딩하지 않는 방안 및 라이선스 키와 연동함으로써 제한 기능과 관련된 정보를 노출시키지 않는 방안을 연구할 예정이다.

감사의 글

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학
지원사업의 연구결과로 수행되었음"(2019-0-01056)

[참고문헌]

[1] 소프트웨어 정책연구소(SPRI), “국가별 SW 불법 복제율,” https://stat.spri.kr/posts/view/22306?code=stat_sw_illegal_copy, 2019년 6월 28일 등록, 2021년 2월 5일 접속.

[2] 한국저작권위원회, “정정당당 브로슈어,” https://www.copyright.or.kr/kcc/itsam/licensedata/view.do?brdctsno=2058, 2021년 2월 5일 접속.

[3] 한국저작권위원회, “소프트웨어 관리가이드,” https://www.copyright.or.kr/information-materials/publication/education-and-promotion/view.do?brdctsno=40228&list.do?pageIndex=1&brdctsstatecode=&brdclasscode=&servicecode=06&nationcode=&searchText=&searchTarget=ALL#, 2017년 3월 7일 등록, 2021년 2월 5일 접속

[4] 서재훈, 유성민, 박유경, “소프트웨어 보호를 위한 기술 분석,” 한국정보기술학회 학회지, 제13권, 제1호, pp. 33-39, 2015년 6월.

[5] 이경률, 육형준, 임강빈, 유일선, “소프트웨어 보안을 위한 난독화 기술 동향,” 한국정보과학회 학회지, 제34권, 제1호, pp. 22-27, 2016년 1월.

[6] 조성제, 김동진, 박민규, “소프트웨어 저작권 보호 기술 동향,” 한국정보기술학회 학회지, 제11권, 제2호, 2013년 12월

[7] 장혜영, “역공학 공격에 대한 소프트웨어 보호 기법,” 단국대학교, 박사학위논문, 2010년 8월

[8] 강기봉, “컴퓨터프로그램의 리버스 엔지니어링에 관한 법정책적 소고”, https://www.moleg.go.kr/mpbleg/mpblegInfo.mo?mid=a10402020000&mpb_leg_pst_seq=133303, 한양대학교 법학연구소, 2014년 3월 25일 등록, 2021년 2월 5일 접속

[9] 유성민, “소프트웨어 저작권 보호를 위한 정부의 정책방향,” 한국정보기술학회 학회지, 제13권, 제1호, pp. 41-47, 2015년 6월