소프트웨어 불법복제 및 역공학 취약점 분석: A 소프트웨어를 기반으로

이재혁*, 이경률*

*대구가톨릭대학교 컴퓨터소프트웨어학부

Vulnerability Analysis of Software Piracy and Reverse Engineering: Based on A Software

JaeHyuk Lee* Kyungroul Lee*

*School of Computer Software, Daegu Catholic University

요 약

정보통신 기술의 발전으로 인하여, 다양한 소프트웨어들이 개발되고 있으며, 소프트웨어 시장 및 산업이 활성화되었다. 시장 및 산업이 활성화됨에 따라, 소프트웨어의 소스코드와 같은 저작권을 보호하기 위한 기술들이 등장하였다. 그럼에도 불구하고, 소프트웨어 시장 및 산업에 큰 걸림돌이 되는 소프트웨어 불법복제가 지속해서 시도되는 실정이다. 따라서 소프트웨어 저작권 보호 기술의 안전성 향상을 위한 방안이 요구됨에 따라, 불법복제 소프트웨어의 사용을 탐지하고 방지하기 위하여, A 소프트웨어를 중점으로 소프트웨어 저작권 보호 기술이 가지는 취약점과 보안 위협을 분석하고 실증한다.

키워드: 소프트웨어 저작권 보호 기술, 소프트웨어 불법복제, 역공학, 라이선스 인증

I. 서론

IT (Information Technology) 기술이 비약적으로 발전함에 따라, 패키지 소프트웨어, IT 서비스, 게임 소프트웨어와 같이 사용자 개인의 목적에 초점을 맞춘 소프트웨어가 많이 개발되는 실정이다 [1].

소프트웨어 시장 및 산업이 활성화됨에 따라, 상용 소프트웨어 (Commercial Software), 프리웨어 (Freeware), 셰어웨어 (Shareware)를 비롯한 다양한 소프트웨어 종류가 등장하였다 [2]. 그중, 소프트웨어 저작권 보호를 위하여, 프리웨어를 제외한 대부분 소프트웨어는 소프트웨어 저작권자에게 정당한 비용을 지급함으로써 사용하며, 이러한 기술이 소프트웨어 저작권 보호 기술이다 [4].

소프트웨어 저작권 보호 기술이 적용되었음에도 불구하고, 소프트웨어 역공학을 통하여, 정품 미보유, 라이선스 위반, 기간 초과 사용과 같은 소프트웨어 불법복제 및 불법 사용이 지속해서 시도되는 현실이다 [3]. 이러한 불법복제 소프트웨어의 사용은 소프트웨어 개발자의 개발 의욕을 감소시킬 뿐만 아니라, 전반적인 소프트웨어 시장 및 산업에 큰 걸림돌이 된다 [4]. 따라서 불법복제 소프트웨어의 사용을 탐지하고 방지하기 위하여, 소프트웨어 저작권 보호 기술이 가지는 취약점 및 보안 위협을 분석할 필요성이 있으며, 그 결과를 기반으로 소프트웨어 저작권 보호 기술의 안전성 향상 방안이 요구된다.

이러한 요구사항을 만족하기 위하여, 본 논문에서는 소프트웨어 저작권 보호 기술 중 라이선스 인증이 적용된 A 소프트웨어를 기반으로, 라이선스 인증 방식을 분석하고 불법복제 취약점을 도출한다.

논문의 구성은 다음과 같다. 2장에서는 A 소프트웨어의 라이선스 인증 취약점을 분석하기

위하여 인증과정을 분석하며, 3장에서는 분석한 결과를 기반으로 신규 발굴한 A 소프트웨어 취약점을 서술한다. 마지막으로 결론 및 향후 계획을 4장에 서술한다.

II. A 소프트웨어의 소프트웨어 저작권 보호 기술 분석

일반적으로 소프트웨어 저작권 보호를 위한 기술은 라이선스 키를 기반으로 정품을 인증하는 라이선스 인증기술 [5], 소프트웨어에 특정 정보를 삽입하여 무결성을 탐지하는 워터마킹 기술 [6], 마지막으로 평문을 해독 불가능한 형태로 변형하는 난독화 및 암호화 기술이 있다 [4, 7]. 본 논문에서의 분석 대상인 A 소프트웨어는 라이선스 인증기술을 사용하여 소프트웨어 저작권을 보호한다.

A 소프트웨어의 라이선스 인증기술 취약점을 분석하기 위하여, 라이선스 인증과정을 분석하였으며, 이를 그림 1에 나타내었다. A 소프트웨어의 라이선스 인증 방식은 크게 소프트웨어 사용 단계, 라이선스 인증 단계, 라이선스 인증 결과 단계인 총 3단계로 구성된다.

첫 번째 단계인 소프트웨어 사용 단계는, 사용자가 소프트웨어를 설치하고, 실제로 사용하는 단계이다. 이 단계에서 라이선스가 인증되지 않으면, 특정 기능이나 사용 기간이 만료되어 소프트웨어를 정상적으로 사용하지 못한다.

두 번째 단계인 라이선스 인증 단계는 특정 기능이나 사용 기간의 제한을 해제하기 위하여 라이선스 키를 인증하는 단계이다. 라이선스 키를 인증하기 위하여 사용자는 정당한 비용을 지불하여 라이선스 키를 발급받으며, 발급된 키를 인증함으로써 제한 기능 및 기간 만료 없이 정상적으로 프로그램을 사용한다.

A 소프트웨어의 제한 기능 및 기간은 표 1과 같으며, 제한 기간은 없고 특정 기능만 제한한다. 특정 기능으로는 첫 세션을 기준으로 200회의 명령을 사용할 수 있으며, 그 이후부터는 20회의 명령마다 라이선스 구매 유도 메시지를 출력한다. A 소프트웨어의 최대 명령 횟수 제한은 2,500회이며, 그 이후부터는 1회의 명령마다 라이선스 구매 유도 메시지가 출력된다. 따라서 2,500회를 초과하는 경우에는 소프트웨어를 구매하지 않고서는 정상적으로 소프트웨어를 사용하기 어렵다.

마지막 단계인 라이선스 인증 결과 단계는 사용자가 입력한 라이선스 키를 기반으로 인증결과를 출력하며, 인증에 성공한 경우에는 최대 명령 횟수 제한이 없는 정식 버전을 사용할 수 있고, 인증에 실패한 경우에는 계속 평가 버전을 이용한다.

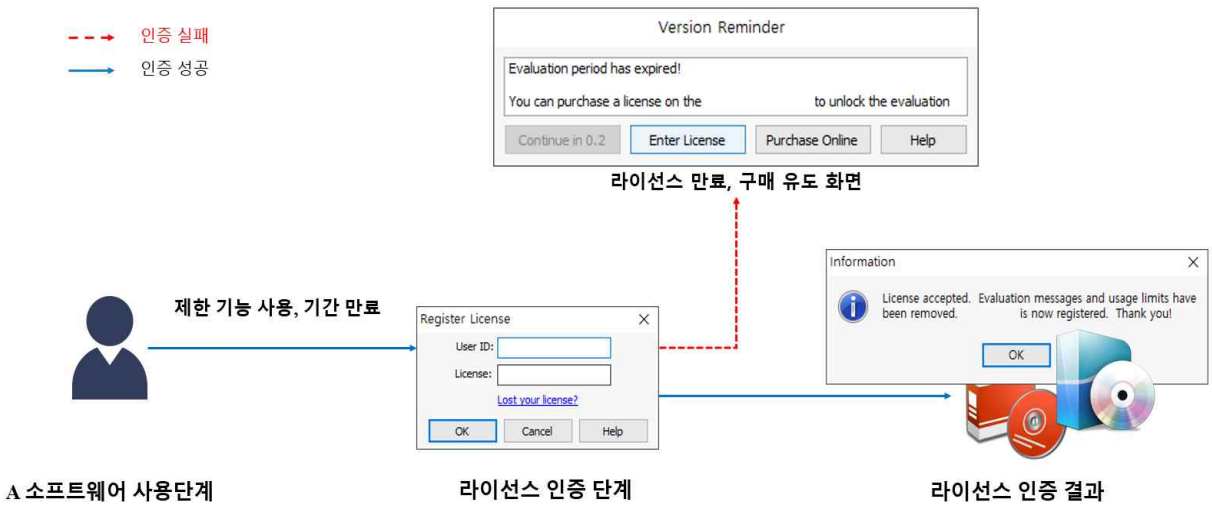

그림 1 A 소프트웨어 인증과정 및 결과

표 1. A 소프트웨어 기능 및 기간 제한

제한 항목	설명
기능 제한	최대 입력 가능 횟수 제한 (총 2,500회 제한, 첫 세션 당 200회, 이후 20회)
기간 제한	없음

III. A 소프트웨어 불법복제 취약점 분석

3.1. 라이선스 인증 취약점

A 소프트웨어에 적용된 소프트웨어 저작권 보호 기술은 라이선스 인증기술이며, 이 기술은 소프트웨어 내부에 유효한 라이선스 키와 관련된 정보가 노출되는 근본적인 문제점이 존재한다. 즉, 사용자가 입력한 키를 검증하기 위해서는 유효한 라이선스 키가 내부에 존재하여야만 하며, 해당 키들을 비교할 수밖에 없는 구조로 인하여 유효한 라이선스 키가 노출된다.

이러한 문제점으로 인하여, 불법복제가 가능한 취약점 및 보안 위협이 발생한다. 따라서 라이선스 키를 비교하는 코드를 분석한다면, 유효한 라이선스 키를 탈취할 수 있으며, 탈취된 키를 기반으로 소프트웨어를 구매하지 않고도 정당한 사용자로 인증하도록 우회하는 것이 가능하다.

A 소프트웨어의 라이선스 인증 방식 역시 이러한 구조적인 문제점으로 인하여 소프트웨어 저작권 보호 기술의 취약점이 발생하고, 불법적인 사용이 가능하다. 본 논문에서는 A 소프트웨어를 대상으로 라이선스 인증기술의 취약점을 분석한다.

라이선스 키를 비교하는 코드를 분석한 결과, 입력한 라이선스 키를 인자로 전달되는 함수를 확인하였고, 이를 그림 2에 나타내었다. 입력한 User ID는 1234이고, License는 QWER이다. 이러한 구조는 해당 함수가 User ID와 License를 비교하기 위한 함수일 가능성이 매우 높다.

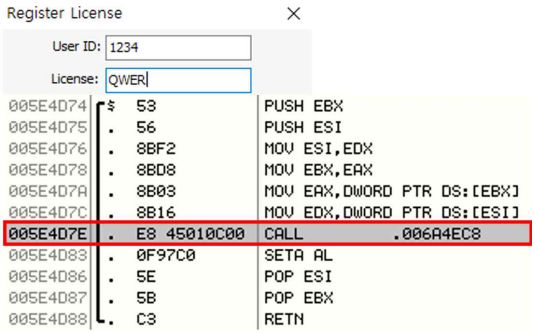

그림 2. 분석한 라이선스 키 비교 함수

해당 함수가 라이선스 키를 비교하는 함수임을 검증하기 위하여, 해당 함수가 호출될 때의 인자를 확인하였으며, 입력한 User ID와 License 정보가 그림 3과 같이 EAX에 저장되는 것을 확인하였다.

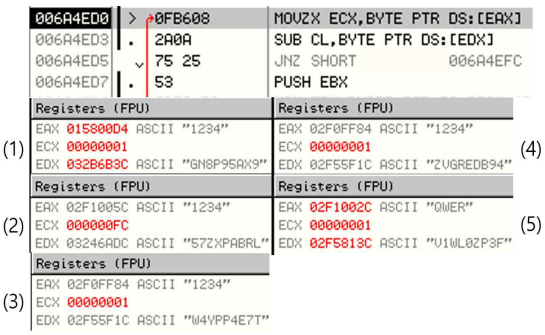

그림 3. 분석한 하드코딩된 유효한 User ID 및 License 정보

상기 인증정보를 비교하는 함수는 총 5번 호출되고, 첫 번째 호출부터 네 번째 호출까지는 EAX에 User ID가 저장되었으며, 마지막 다섯 번째 호출에서는 EAX에 License가 저장되었다. 같은 호출과정에서 EDX에는 임의의 9자리 문자열이 저장되는 것을 확인하였다. 따라서 이 함수를 인증정보를 비교하는 함수라고 가정한다면, EDX에 저장된 정보는 유효한 User ID와 License라고 판단할 수 있다.

이러한 가정을 바탕으로, 노출된 유효한 User ID와 License를 검증하기 위하여, 다섯 번째 호출에서 EDX에 저장된 문자열을 그림 4와 같이 입력한 결과, 정상적으로 라이선스가 인증되었으며, 제한 기능이 제거된 것을 확인하였다.

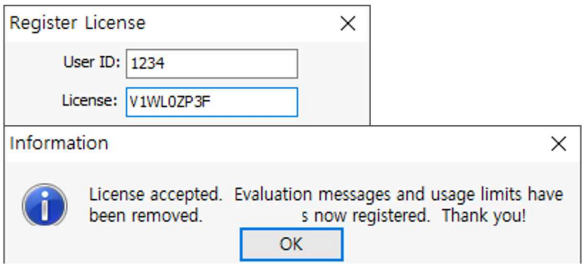

그림 4. 라이선스 인증 우회 결과

본 논문에서 분석한 결과를 통하여, A 소프트웨어는 소프트웨어 내부에 라이선스 인증정보를 하드코딩한다는 것을 실증하였으며, 노출된 라이선스 인증정보를 통하여 인증을 우회하는 것을 검증하였다.

하지만, 이 정보를 활용하는 경우, 라이선스가 정상적으로 등록되었다는 메시지가 출력되지만, 실제로 제한된 기능이 제거되지는 않는다. 다시 말하면, 상기 취약점을 활용하여 라이선스 인증을 우회하더라도, 최대 명령 횟수 제한인 2,500회를 초과하면, 평가 기간이 만료되었다는 메시지가 출력되며, 기능 사용에 제약이 따른다.

3.2. 기능 제한 취약점

상기 라이선스 인증을 우회하더라도 제한이 제거되지 않은 기능을 무력화하기 위하여, 기능 제한 취약점을 분석하였다. A 소프트웨어는 기능 제한을 위하여 라이선스 구매 유도 메시지를 반복적으로 출력한다. 이는 명령 횟수를 계산하고, 기준 횟수를 초과할 경우, 메시지를 출력하는 것으로 판단된다. 따라서 기준 횟수를 초과할 경우, 메시지를 출력하기 위하여 호출되는 함수가 존재할 것으로 가정하고, 해당 함수를 분석하였다.

분석 결과, 그림 5와 같이 0x004EBCA7 위치에서 특정 함수를 호출하는 것을 확인하였고, 해당 함수에서 라이선스 구매 유도 메시지가 출력되었다. 자세하게는, 해당 함수 내부에서 출력되는 라이선스 구매 유도 메시지에 포함되는 문자열인 “Evaluation period has expired !”와 “Session limit exceeded!”가 노출된 것을 확인하였다. 즉, 해당 함수는 평가 기

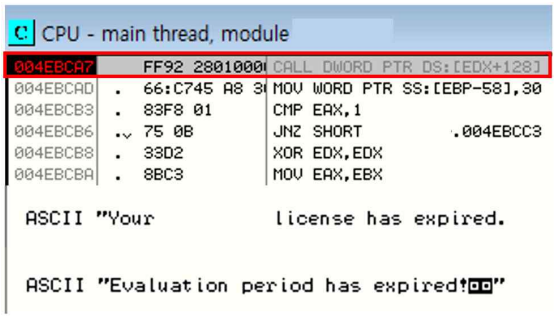

그림 5. 분석한 라이선스 구매 유도 메시지 출력 함수

간이 만료되었을 때, 라이선스 구매 유도를 위한 함수인 것으로 가정하였다.

따라서 본 논문에서는 해당 함수가 라이선스 구매 유도 메시지를 출력하는 함수이며, 기능을 제한하는 코드를 실행하는 함수로 가정하여, 해당 함수가 호출되지 않도록 수정하였다. 그 결과, 그림 6과 같이 최대 입력 가능한 명령 횟수를 초과하더라도, 라이선스 구매 유도 메시지가 출력되지 않고 정상적으로 사용하는 결과를 실증하였다.

그림 6. 기능 제한 우회 결과

본 논문에서 분석한 A 소프트웨어의 소프트웨어 저작권 보호 기술 취약점을 통하여, 소프트웨어 내부에 하드코딩된 라이선스 정보가 존재한다는 것뿐만 아니라, 특정 코드를 수정함으로써 기능 제한을 우회하는 취약점을 도출하였다. 도출된 취약점을 통하여 소프트웨어 불법복제와 같은 불법적인 사용이 가능한 것으로 판단된다.

IV. 결론

본 논문에서는 소프트웨어 저작권 보호 기술의 안전성 향상을 위하여, 소프트웨어 저작권 보호 기술 중 라이선스 인증이 적용된 A 소프

트웨어를 기반으로, 라이선스 인증 방식 및 과정을 분석하고 불법복제 취약점을 도출하였다. 도출한 취약점으로는 라이선스 인증 취약점과 기능 제한 취약점이 있으며, 실질적인 취약점인 기능 제한 취약점을 실증함으로써 신규 취약점을 발굴하였다. 라이선스 인증 취약점은 라이선스 키가 하드코딩되어 코드상에 그대로 노출되는 취약점이지만, 실질적으로 라이선스 인증이 우회되지는 않고 기능이 제한되었다. 이에 특정 코드를 수정함으로써 기능 제한을 우회하여 소프트웨어를 구매한 사용자와 동일하게 제한되었던 기능의 사용이 가능한 취약점을 검증하였다.

향후 연구로는, 본 논문에서 발굴한 취약점의 근본적인 문제점 해결을 위하여, 하드코딩된 라이선스 키를 노출하지 않는 방안 및 강인한 라이선스 인증 방안을 연구할 계획이다.

감사의 글

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

[참고문헌]

[1] 한국정보통신진흥협회, "2019 ICT실태조사 보고서," https://www.kait.or.kr/notice/stat_board_view.jsp, 2020년 9월 10일 등록, 2020년 2월 2일 접속.

[2] 이진천, "소프트웨어 라이센스 종류," 대한설비공학회 설비저널, 제43권, 제6호, pp. 102-103, 2014년 6월.

[3] 소프트웨어정책연구소, "국가별 SW 불법복제율," https://stat.spri.kr/posts/view/22306?code=stat_sw_illegal_copy, 2019년 6월 28일 등록, 2020년 2월 2일 접속.

[4] 조성제, 김동진, 박민규, "소프트웨어 저작권 보호 기술 동향," 한국정보기술학회 학회지, 제11권, 제2호, pp. 23-32, 2013년 12월.

[5] 이상렬, "인터넷을 통한 소프트웨어 불법사용 방지시스템 설계," 한국컴퓨터정보학회 논문지, 제6권, 제4호, pp. 110-118, 2001년 12월.

[6] Y. Wang, D. Gong, B. Lu, F. Xiang, and F. Liu, "Exception Handling-Based Dynamic Software Watermarking," IEEE ACCESS, vol. 6, pp. 8882-8889, Feb. 2018.

[7] J. Katz and Y. Lindell, "Introduction to Modern Cryptography," 3rd Ed., CRC press, Dec. 2020.