SHIN&KIM
법무법인(유) 세종

AI 리버스 엔지니어링(REA)과 ArtCraft 사례로 살펴보는 저작권법적 쟁점

2026.11.04

1. 들어가며

최근 AI 코딩 에이전트에 리버스 엔지니어링 도구를 연결해 주는 오픈소스 'REA(Reverse Engineer Anything)'와, 어도비·마이크로소프트의 제품군을 '클린룸' 방식[1]으로 재구현했다고 주장하며 등장한 'ArtCraft'가 업계의 주목을 끌고 있습니다. 이번 뉴스레터에서는 위 각 사례에서 우리 저작권법적으로 문제될 수 있는 지점들을 짚어보겠습니다.

2. 현황

REA는 사용자가 '이 프로그램의 어떤 기능이 어떻게 구현되어 있는지 알려 달라'고 말로 지시하면, AI가 전문 분석 도구를 대신 조작하여 실행 파일을 살펴보고 결과를 설명해 주는 오픈소스 프로그램입니다. 종래 전문가만 다루던 리버스 엔지니어링 도구(Ghidra, IDA 등)[2]의 문턱을 일반 개발자 수준으로 낮추었다고 평가되고 있으며, 2026년 10월 한 주 만에 깃허브(GitHub) 추천 수만 10만 개를 넘겼을 정도로 큰 화제가 되고 있습니다.[3] 개발자도 공식 설명에서 원본 소스코드를 복원하거나 앱을 자동으로 복제하는 도구가 아니라고 밝히고 있으며, 연구·호환 목적과 자신이 권한을 가진 소프트웨어에만 사용하라고 안내하고 있습니다. 그러나 실제로는 REA를 활용한 콘솔 독점 게임의 PC 이식판이 잇달아 등장하는 등 문제가 되고 있습니다.

한편, ArtCraft는 미국 개발자가 Claude로 만든 포토샵 등 어도비 7개 앱과 MS 오피스 3종의 무료 대체 소프트웨어입니다. 해당 개발자는 원본 코드를 보지 않고 공개 문서와 관찰된 동작만으로 작성한 '클린룸 재구현'이라고 설명하면서, REA나 Ghidra는 사용하지 않았다고 밝혔습니다.

이와 같은 프로그램들은 기본적으로 원본 프로그램들의 기능을 동일, 유사하게 재현하는 데에 사용되고 있는데, 프로그램 동작 과정에서 구체적으로 어떤 작업들이 일어나는지에 따라 저작권법상 다양한 이슈가 제기될 수 있습니다. 이하에서 여러 상황을 가정하여 살펴보겠습니다.

3. 제기될 수 있는 저작권 이슈

(1) ArtCraft와 같은 클린룸 방식의 경우

① 원본 코드를 보지 않고 공개 문서와 동작 관찰만으로 재구현하는 경우: 컴퓨터프로그램은 저작물로 보호되지만(저작권법 제4조 제1항 제9호), 프로그램 언어·규약·해법에는 저작권법이 적용되지 않습니다(제101조의2). 따라서 프로그램의 기능·알고리즘 그 자체는 보호의 대상이 아니므로 프로그램저작권 침해는 성립하기 어렵습니다. 다만 '클린룸'은 개발 과정에 관한 스스로의 주장일 뿐이므로, 원본 코드나 역분석 결과에 접근하지 않았다는 사실을 개발 기록으로 뒷받침해 둘 필요가 있습니다.

② 화면 구성까지 동일하게 만든 경우: UI는 코드와 별개로 평가됩니다. 아이콘·그래픽·화면 디자인에 창작성이 있으면 그 부분에 관한 저작권 침해가 성립할 수 있습니다. 다만 메뉴 구조·도구 배열처럼 기능상 꼭 필요한 부분의 경우에는 보호의 범위가 좁아질 수 있으므로, 요소별로 창작적 표현인지 여부를 가려내야 합니다. 명칭·로고의 상표권, 성과물 무단사용(부정경쟁방지법 제2조 제1호 (파)목) 여부는 별도로 검토할 필요가 있습니다.

(2) REA를 활용하는 경우

① 정당하게 구입한 프로그램을 REA로 분석하여 원리만 파악하는 경우: 복제에는 일시적 저장도 포함되므로(제2조 제22호), 프로그램 실행 파일을 역분석 도구로 불러들여 사람이 읽을 수 있는 형태의 코드로 되돌리는 과정에서, 원 프로그램이 메모리에 저장되면 복제에 해당할 수 있고, 다른 형태의 코드로 변환되면 2차적저작물작성에 해당할 수 있습니다(제5조, 제22조).[4] 다만 저작권법은 위와 같은 프로그램 역분석을 허용하는 예외 규정을 두고 있습니다. 먼저, 정당한 권한에 따라 프로그램을 이용하는 자는 그 프로그램의 기초를 이루는 아이디어 및 원리를 확인하기 위하여 프로그램의 기능을 조사·연구·시험할 목적으로 복제할 수 있습니다(제101조의3 제1항 제6호). 다음으로 정당한 권한에 의하여 프로그램을 이용하는 자 또는 그의 허락을 받은 자는 호환에 필요한 정보를 쉽게 얻을 수 없고 그 획득이 불가피한 경우에는, 해당 프로그램의 호환에 필요한 부분에 한정하여 프로그램의 저작재산권자의 허락을 받지 아니하고 프로그램코드 역분석을 할 수 있습니다(제2조 제34호, 제101조의4 제1항). 따라서 이처럼 정당하게 구입한 프로그램의 작동 원리를 이해할 목적으로 분석하는 경우라면, 첫 번째 예외에 해당할 가능성이 높습니다. 다만 저작권자의 이익을 부당하게 해치는 경우에는 위 예외가 적용되지 않고(같은 항 단서), 라이선스 계약에 역분석 금지 조항이 있으면 저작권 침해와는 별개로 계약 위반이 문제될 수 있습니다.

② 분석 결과를 토대로 대체 프로그램을 만들어 배포하는 경우: 이 경우는 호환 목적을 벗어나게 되므로 앞서 본 제101조의4의 적용을 받기 어렵고, 분석 단계의 복제도 권리자 이익을 해치는 것으로 평가될 가능성이 있습니다. 분석으로 얻은 결과를 보면서 작성한 코드에는 원 프로그램의 표현(코드 문언, 구조·배열)이 그대로 담겨있을 가능성도 높아 복제권·2차적저작물작성권 및 전송권 침해가 성립할 수 있습니다.

③ 콘솔 게임을 분석하여 PC 이식판을 배포하는 경우: 원 프로그램의 표현을 유지한 채 변환하는 형태이므로 복제권·2차적저작물작성권 침해에 해당할 가능성이 높고, 이를 깃허브 등에 올려 사용자들이 다운받을 수 있게 한다면 전송권 침해도 문제됩니다.

4. 시사점

AI기술의 비약적 발전과 함께, 과거에는 전문가 팀이 수개월을 들이던 소스코드 분석과 재구현을 이제는 개발자 한 명이 AI와 함께 며칠 만에 해낼 수 있게 되었습니다. 그러나 타사 프로그램을 분석하는 경우에는 AI 에이전트가 어떤 프로그램을 어디까지 분석하게 할 것인지 등을 내부 기준으로 정하고, 분석 전에 라이선스의 역분석 금지 조항 등을 확인할 필요가 있습니다. AI로 대체 소프트웨어를 개발하는 경우에

는 AI에게 어떤 자료를 주고 무엇을 금지했는지 등 작업과정을 프롬프트와 작업 기록으로 남겨, 원본 코드에 접근하지 않았다는 사정을 사후에 입증할 수 있어야 할 것입니다.

[1] 클린룸은 개발자가 원본 프로그램의 소스코드를 보지 않고, 공개된 기술문서와 겉으로 드러난 기능만을 토대로 처음부터 독자 코드를 작성하는 개발방식을 말합니다.
[2] Ghidra는 미국 국가안보국이 2019년 오픈소스로 공개한 무료 리버스 엔지니어링 도구이고, IDA는 벨기에 Hex-Rays사의 상용 도구로 보안 업계에서 표준으로 쓰이고 있습니다. 두 도구 모두 실행 파일의 기계어를 사람이 읽을 수 있는 코드로 되돌리는 디컴파일 기능을 수행합니다.
[3] 다만 REA는 디컴파일을 직접 수행하지는 않고 Ghidra와 같은 기존 분석 도구를 AI가 쓰기 쉽게 연결해 주는 역할을 한다고 알려져 있습니다.
[4] 원 프로그램의 창작적 표현을 이용해 새 프로그램을 만들면, 언어를 바꾸거나 코드를 다시 썼더라도 2차적저작물작성권 침해가 성립할 수 있습니다.

[English version] AI-Assisted Reverse Engineering and Clean-Room Development: Copyright Risks Under Korean Law

관련구성원

김우균
변호사

02-316-4083
wgkim@shinkim.com

김흥식
전문위원

02-316-1985
hsikim@shinkim.com

송종훈
변호사

02-316-2590
johsong@shinkim.com

박성호
변호사

02-316-1980
shopark@shinkim.com

Copyright SHIN & KIM LLC. All rights reserved.