##### Project Document

Research Proposal

|  |  |
| --- | --- |
| Project Name | 바이너리 프로그램에서 제어 구조를 식별하는 도구 개발 |

15 조

202002514 안상준

202202602 손예진

202202487 박혜연

지도교수: 조은선 교수님 (서명)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev# | Date | Affected Section | Author |
| 1 | 2023/03/11 | 초안 작성 | 박혜연 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

목차

[1. 연구 주제 이름 5](#_Toc192519163)

[2. 연구 배경 및 관련 연구 5](#_Toc192519164)

[3. 프로젝트 수행자의 의도 5](#_Toc192519165)

[4. 탐구 내용 및 기대 결과 5](#_Toc192519166)

[5. 프로젝트 관련 학습 계획 6](#_Toc192519167)

[6. 연구 일정 계획 6](#_Toc192519168)

List of Figure

그림 목차 항목을 찾을 수 없습니다.

# 연구 주제 이름

참고] 주제에 근사한 이름으로 작명

‣ 연구의 핵심 키워드를 포함하여 논문 스타일로 작성

‣ 특정 문제를 해결하는 연구인지, 기존 연구의 확장인지 명확히 구분

# 연구 배경 및 관련 연구

난독화는 다양한 방법을 적용하여 코드의 가독성을 낮추는 것이다. 난독화 된 코드를 분석해보면 반복되는 패턴을 확인할 수 있다. 이 패턴을 기반으로 LLM 학습을 진행한다면 충분히 역난독화를 수행할 수 있을 것이라고 생각한다. 따라서 가상화 난독화를 역난독화 할 수 있는 모델을 만드는 것이 목적이다.

악성코드가 발견되었을 때, 이를 빠르게 분석하는 것이 매우 중요하다. 그러나, 가상화 기반 난독화 된 악성코드가 증가하면서 분석이 점점 어려워지고 있다. 따라서 이를 빠르게 분석할 수 있는 도구가 필요하다. 최근 LLM이 다양한 분야에서 사용되고 있지만, 난독화/역난독화 등 보안 분야에서는 아직 흔히 사용되지 않는 것으로 보인다. 이는 데이터셋의 부족과 LLM의 신뢰성 문제가 주된 원인이다. 그렇기에 이 프로젝트를 통해 LLM 학습을 위한 역난독화 데이터셋을 생성하고 LLM이 수행한 역난독화를 신뢰성 있게 사용할 수 있다는 것을 입증할 필요가 있다.

참고] 기존 연구를 바탕으로 연구 배경을 정리 (아래 표 참조)

‣ 관련 논문을 (3편 이상) 바탕으로 정리

‣ 기존 연구의 한계를 분석하여 연구 필요성을 강조

# 프로젝트 수행자의 의도

LLM을 이용하여 가상화 난독화 된 코드를 역난독화 하는 것이 가장 큰 목적이다. 가상화 난독화 된 코드를 분석하여 원본 코드의 구조를 확인할 수 있다. 이는 위/변조 된 프로그램이나 난독화 된 악성코드를 탐지하여 이로 인한 피해를 막을 수 있다. 이 과정에 LLM을 사용한다면, 더 빠른 분석이 가능할 것이라고 생각한다.

# 탐구 내용 및 기대 결과

가상화 역난독화에 특성화 된 모델을 개발할 것이다. 역난독화 분야에서는 아직 LLM이 많이 사용되지 않는다. 이는 LLM에 대한 신뢰성의 문제일 것이다. 이번 프로젝트를 통해 보안 분야에서 LLM이 사용될 수 있음을 입증하고, 실제 사용하여 코드 분석에 필요한 시간이 줄어들 수 있기를 바란다. 또한, 데이터 역시 공개되어있는 데이터셋을 사용하기 어려울 것이라고 생각하는데, 직접 데이터셋을 구축하여 LLM의 학습에 활용하고, 추후 다른 연구에도 사용될 수 있을 것이라고 생각한다.

# 프로젝트 관련 학습 계획

|  |  |  |
| --- | --- | --- |
| 학습할 내용 | 기간 | 역할 분담 |
| 난독화, 컴파일러, 바이너리코드 등 프로젝트에 필요한 기본 지식 | 03.01 ~ 03.31 |  |
| IDA 사용법 | 03.01 ~ 03.31 |  |
| 바이너리 코드에서의 switch, while 구조 | 03.01 ~ 03.31 |  |
|  |  |  |

# 연구 일정 계획

|  |  |  |
| --- | --- | --- |
| 조사할 내용 | 기간 | 역할 분담 |
| Switch문 탐지를 위한 특징적인 코드 패턴 | 03.01 ~ 03.31 |  |
| LLM 학습을 위한 데이터 생성 및 전처리 방안 | 04.01 ~ |  |
|  |  |  |
|  |  |  |

**Related Work Summary Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 연구 제목(저자) | 저널/컨퍼런스(연도) | 주요 내용 요약 | 주요 인사이트 |
| 1 | Can LLMsObfuscate Code? A Systematic Analysis of Large Language Models into Assembly Code Obfuscation  (Seyedreza Mohseni, Seyedali Mohammadi, Deepa Tilwani, Yash Saxena, Gerald Ketu Ndawula, Sriram Vema, Edward Raff, Manas Gaur) | AAAI(2025) | 기존의 난독화 엔진을 LLM으로 대체하여 코드 난독화를 진행하였다. 난독화 기법으로는 Dead Code Insertion, Register Substitution, Control Flow Change 를 사용해서 어셈블리 코드 데이터셋 MAD를 생성했다. LLM을 사용해 prompting과 in-context-learning을 통해 이 데이터셋의 학습가능성과 신뢰성을 입증했다. 여러 LLM 중에서 GPT가 뛰어난 성능을 보였다. | 어셈블리 코드는 학습에 필요하지않은 부분도 많이 포함되어있기 때문에 LLM을 역난독화에 사용하기 위해서는 데이터의 전처리가 중요할 것으로 보인다.  현재까지는 난독화에 대한 LLM의 평가 기준이 모호하기 때문에 모델의 성능을 평가할 기준에 대해서도 고민이 필요할 것 같다. |
| 2 | Deobfuscation of Virtualization-Obfuscated Software  (Kevin Coogan, Gen Lu, Saumya Debray) | CCS(2011) | 가상화 기반 난독화 된 악성코드가 증가하면서, 이를 빠르게 분석하는 것이 점점 어려워지고 있다. 기존에는 리버스 엔지니어링 후, 이를 기반으로 바이트 코드에 내재된 논리를 복원하는 방식으로 이루어진다. 이 논문에서는 system call의 흐름을 분석하여 난독화 된 프로그램의 원래 논리를 복원하는 접근 방식을 제안한다. trace 추출, system call 및 인자 추출, 명령어 분석, 관련 명령어 subtrace 생성의 단계로 이루어진다. 이는 악성코드 분석 및 리버스 엔지니어링에 효과적으로 사용될 수 있다. | 프로그램의 구조를 확인하기 위해 다양한 방법을 시도해 볼 수 있을 것 같다. 또한, 데이터 수집 및 전처리 과정에 대해서도 고민이 필요할 것이다. 코드 분석 방식에 따라 필요한 정보가 다르기에 데이터 전처리 방식도 달라져야 할 것이라고 생각한다. |
| 3 |  |  |  |  |