

DEOBFUSCATOR

# Project Brainstorming Result

2025.03.26

14조

안상준, 손예진, 박혜연

지도교수 : 조은선

# Contents

01	아이디어 발산
02	아이디어 수렴
03	시각화

## 01 아이디어 발산

## 회의모습



## 01 아이디어 발산

난독화  
역난독화  
llm  
바이너리  
loop  
switch  
가상화  
if else  
pin tool  
jump  
table  
fine  
tuning  
prompting  
chat gpt  
claude

tigress  
ollvm  
vmprotect  
github  
c  
gcc  
clang  
optimize  
동적 분석  
정적 분석  
IDA  
OllyDbg  
ai  
feature  
selection  
vectorize

## 02 아이디어 수렴

핵심 개념	정의 및 하위 개념 목록
AI	<p>1. <u>LLM</u> : claude, chat gpt↓ claude version : 3.7, 3.5 chat gpt : 4.5, 4, o1, o3-mini, o3-mini-high <u>LLaMA</u> : 3.1, 3.2, 3.3 Gemini, PaLM, LaMDA</p> <p>2. <u>Training</u> : fine tuning, prompting, 새로운 모델 구축↓ fine tuning : binary에서 feature를 추출하여 학습, max length를 조절하여 최적의 length탐색 <u>prompting</u> : ChatGpt, Claude를 이용하여, zero shot이나 in context learning을 사용 <u>feature</u> : Pin Tool을 이용하여 trace를 뽑아 feature로 사용 새로운 모델 <u>구축</u> : transformer등과 같은 모델 구조를 사용해서 학습</p> <p>3. <u>Preprocessing</u> : feature selection, vectorize <u>feature selection</u> : instruction sequence 에서 ai 학습에 불필요한 정보 제거로 모델의 효율성을 높임 <u>vectorize</u> : 기존 자연어 처리 방식을 사용하지 않고 바이너리 코드에 맞는 vectorize방식 사용</p>
도구	<p>1. <u>분석</u> : IDA, OllyDbg, PinTool, objdump, ghidra, PEView, QEMU 2. <u>난독화</u> : Ollvm, Tigress, VMProtect, Themida, CodeVirtualizer 3. <u>컴파일러</u> : gcc, clang, optimize, javac, icc, -o2, -oz, -ofast</p>
난독화	<p>1. <u>기법</u> : flattening, opaque predicate, control flow, renaming <u>flattening</u> : 함수의 기본 블록을 하나의 큰 switch 문 안에 배치하여 원래의 제어 흐름을 숨김 <u>control flow</u> : 제어 흐름 난독화는 프로그램의 실행 흐름을 복잡하게 만들어 코드 분석을 어렵게 <u>renaming</u> : 변수, 메서드, 클래스의 이름을 의미 없는 문자열로 변경하여 코드의 가독성을 낮춤 <u>opaque predicate</u> : 항상 참 또는 거짓으로 평가되는 조건문을 삽입하여 불필요한 분기를 만들</p>
제어구조	<p>1. loop, switch, if-else</p>

## 02 아이디어 수렴

바이너리에서  
구조를 식별하는  
LLM 모델 개발

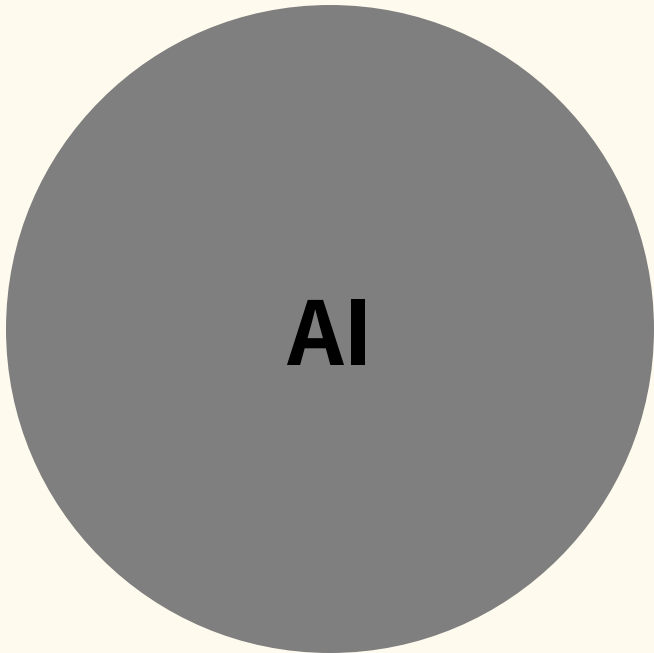
AI

도구

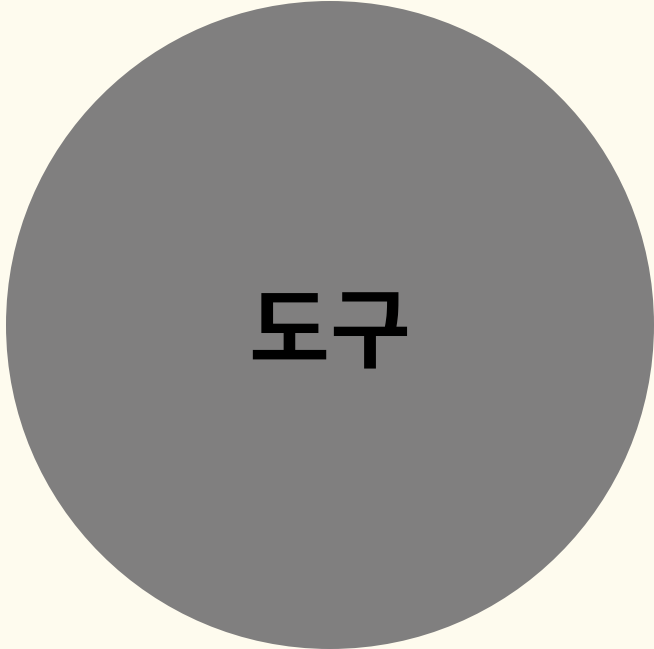
난독화

제어구조

02 아이디어 수렴



LLM	Claude : 3.7, 3.5 Chat GPT : 4.5, 4, o1, o3-mini, o3-mini-high LLaMA : 3.1, 3.2, 3.3 Gemini, PaLM, LaMDA
Training	<b>fine tuning</b> : binary에서 feature를 추출하여 학습, max length를 조절하여 최적의 length를 탐색 <b>prompting</b> : Chat GPT, Claude를 이용하여, zero shot이나 in context learning을 사용 <b>feature</b> : Pin Tool을 이용하여 trace를 뽑아 feature로 사용 <b>새로운 모델 구축</b> : transformer 등과 같은 모델 구조를 사용해서 학습
Preprocessing	<b>feature selection</b> : instruction sequence 에서 AI 학습에 불필요한 정보 제거로 모델의 효율성을 높임 <b>vectorize</b> : 기존 자연어 처리 방식을 사용하지 않고 바이너리 코드에 맞는 vectorize 방식 사용



분석	IDA, OllyDbg, PinTool, objdump, Ghidra, PEView, QEMU
난독화	Ollvm, Tigress, VMProtect, Themida, CodeVirtualizer
컴파일러	gcc, clang, javac, icc, optimize, -O2, -Oz, -Ofast



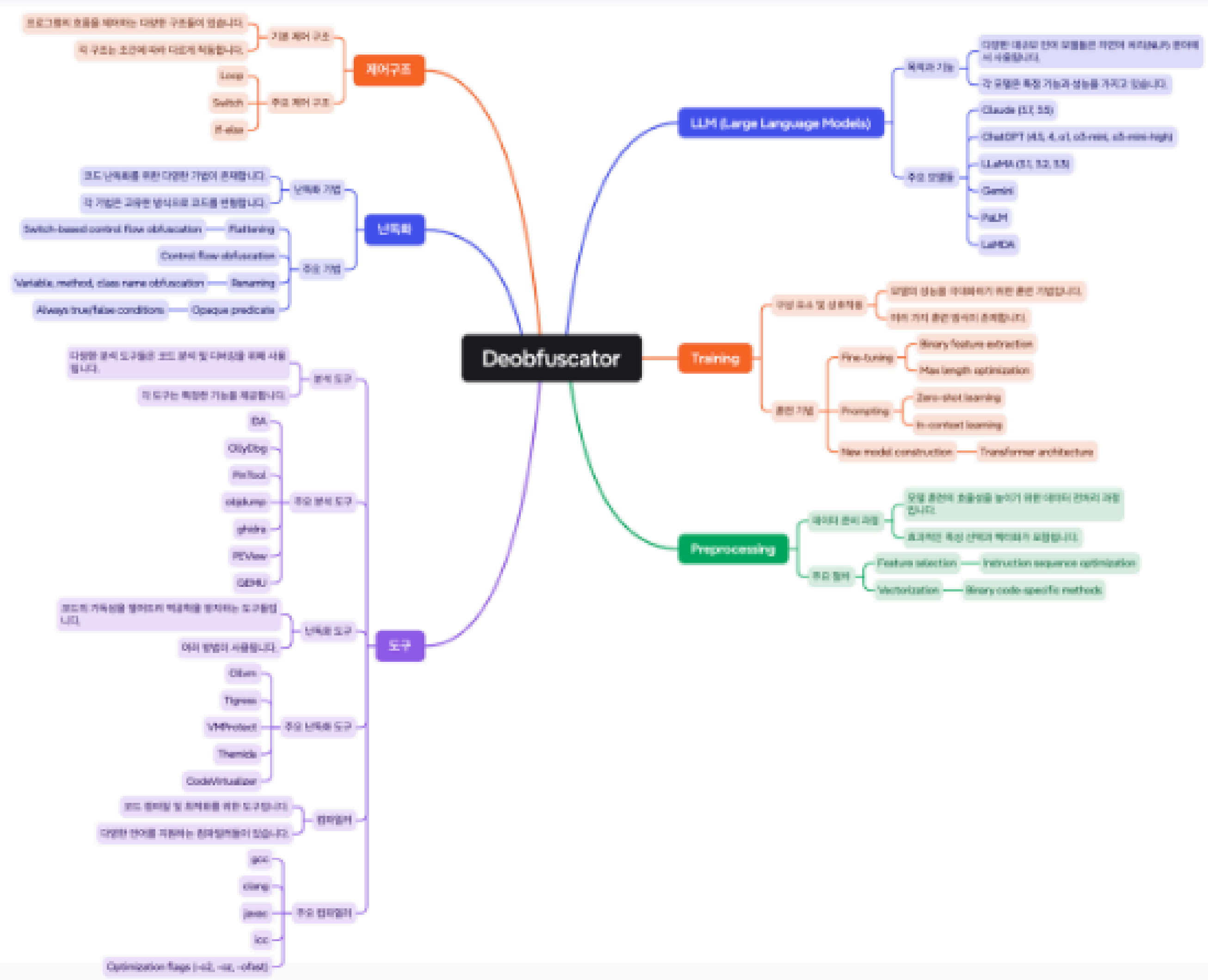


<b>flattening</b>	함수의 기본 블록을 하나의 큰 switch 문 안에 배치하여 원래의 제어 흐름을 숨김
<b>control flow</b>	프로그램의 실행 흐름을 복잡하게 만들어 코드 분석을 어렵게 함
<b>renaming</b>	변수, 메서드, 클래스의 이름을 의미 없는 문자열로 변경하여 코드의 가독성을 낮춤
<b>opaque predicate</b>	항상 참 또는 거짓으로 평가되는 조건문을 삽입하여 불필요한 분기를 만듦



if-else	cmp, test je, jne, jg, jl
switch	jmp eax
loop	loop: dec ecx jnz loop
call	call function ret

03 시각화



DEOBFUSCATOR

**감사합니다**

---

14조

안상준, 손예진, 박혜연

지도교수 : 조은선