NLP기반 익스플로잇 코드 생성 자동화 시스템 설계

진규정* · 임정우** · 장한나*** · 이효찬**** · 배송현**** · 최민석***** · 최준원******

*강원대학교 · **한신대학교 · ***성신여자대학교 · ****고려대학교 · *****대구가톨릭대학교 · ******경희사이버대학교

Automatic Exploit Generation and Enhancement using NLP-based Techniques

Gyu-Jeong Jin* · Jung-Woo Im** · Han-Na Jang***

· Hyo-Chan Lee**** · Song-Hyun Bae**** · Min-Seok Choi***** · Jun-Won Choi******

*Kangwon National University

**Hanshin University

***Sungshin Women's University

****Korea University

*****Deagu Catholic University

******KyungHee Cyber University

요 약

최근 제로데이 취약점의 발견과 악용으로, 소프트웨어 보안 패치가 배포되기 전에 공격자들에 의해 악용되고 있다. 또한, 해커들은 AI 및 머신러닝 같은 첨단기술을 사용함으로써 공격을 더욱 정교하게 수행하고 있다. 이에 따라, 고도화된 공격에 대응하기 위한 기존의 퍼저(fuzzer)와 사후 대응 방식에서 한계점이 드러나고 있다. 머신러닝으로 정교화 및 자동화된 기술은 보안 취약점을 빠르게 악용하고 확산시킬 수 있다. 본 논문에서는 제로데이 공격을 사전에 탐지하는 방법으로, 자연어 처리 기반의 익스플로잇 코드 자동 생성 모델을 사용하여 웹 환경에서의 취약점을 사전에 점검하는 기술을 제안한다. 그 예시로 Cross Site Script 코드와 SQL Injeciton 구문을 토크나이저와 LLama 모델의 아키텍처를 이용하여 학습시킨 NLP기반의 익스플로잇 코드 자동 생성 모델을 제시한다.

I. 서론

현대의 정보 보안에서는 제로데이 취약점의 발견과 악용이 심각한 문제로 인식되고 있다. 이러한 취약점은 공격자들에 의해 발견되고 악용되어 소프트웨어의 보안 패치를 배포하기 전에 피해를 입힐 위험이 있다. 또한 공격자들은 인공지능 및 머신러닝 기술을 활용하여 공격을 더욱 정교하게 수행하고 있다. 이에 따라 기존의 퍼저(fuzzer)와 사후 대응 방법은 이러한 고도화된 공격에 대응하기에 한계가 있다. 본 논문에서는 이러한 문제에 대응하기 위해 자연어 처리(NLP) 기술을 활용한 NLP 기반 익스플로잇 코드 자동 생성 모델을 사용하여, 웹 환경에서 취약점을 사전에 점검할 수 있는 시뮬레이션 기술을 제시한다. 이 모델은 토크나이저와 Large Language Model Meta AI의 한 종류인 LLama 모델을 사용하여 학습된다. 제안 기술은 보안 전문가가 취약점을 미리 파악하고 대응할 수 있는 능동적이고 효율적인 방법을 제공할 것으로 기대된다. 또한, 본 연구와 시뮬레이션을 통해 다양한 웹 취약점을 점검하고, 보안 패치를 통한 사전 대응 기술을 제안함으로써 자동 취약점 탐지 분야에 기여할 것이다.

II. 배경 이론

2.1 SQL Injection

SQL Injection은 웹 애플리케이션의 주요 보안 취약점 중 하나로, 공격자가 악의적인 SQL 쿼리를 웹 응용 프로그램에 주입하여 데이터베

이스에 접근하거나 조작하는 공격기법이다. 이는 주로 사용자로부터의 입력값을 검증 없이 직접 SQL 쿼리에 삽입할 때 발생한다. 공격자는 이를 통해 데이터베이스의 내용을 노출하거나 수정할 수 있다. 공격 쿼리는 Fig. 1과 같다.

```
" or 1=0 –
" or 1=1 –
" or 1=1 or ""=1
' or (EXISTS)
' UNION ALL SELECT
' UNION SELECT
```

Fig. 1. SQL Injection Query

2.2 Cross Site Script

Cross Site Scripting(XSS)은 웹 애플리케이션 보안 취약점 중 하나로, 공격자가 웹 페이지에 악의적인 스크립트를 삽입하여 다른 사용자의 브라우저에서 실행시켜 공격을 수행하는 기법이다. 이를 통해 공격자는 세션이나 쿠키를 탈취하거나, 웹 사용자의 브라우저를 제어할 수 있다. XSS는 FIg. 2와 같이 저장되는 XSS와 반사형 XSS로 나뉘며, 웹 애플리케이션에서 사용자 입력을 필터링하거나 검증 없이 그대로 출력할 때 발생할 수 있다.

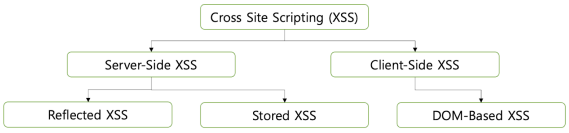

Fig. 2. Cross-Site-Script

2.3 Natural Language Processing (NLP)

Natural Language Processing(NLP)는 기계가 인간의 언어를 이해하고 처리하도록 하는 인공지능의 한 분야로, 텍스트 분석, 기계 번역, 질의응답 시스템 등 다양한 응용 프로그램에서 사용된다. 주요 기술로는 구문과 형태소 분석, 토큰화 등이 있으며, 이러한 기술은 자연어 데이터의 이해 및 처리를 위해 적용된다.

2.4 Tokenizer

Tokenizer는 자연어 처리에서 텍스트를 토큰으로 분리하는 도구로, 문장, 단어, 형태소 등의 단위로 텍스트를 분해하여 모델이 처리하기 쉬운 형태로 만든다. Tokenizer는 주로 문장 분리, 단어 분리, 형태소 분석 등의 작업에 사용되며, NLP 모델의 입력 데이터를 준비하는 데에 필수적인 도구다.

2.5 LLama

LLama는 NLP 모델 중 하나로, 텍스트의 의미를 이해하고 처리하는 데 사용된다. 또한, 대규모 데이터셋을 기반으로 학습되어 자연어 이해, 생성 및 번역과 같은 작업을 수행한다. 이러한 모델은 토큰화된 텍스트를 입력으로 받아, 각 토큰의 의미를 분석하여 이를 바탕으로 원하는 작업을 수행한다. LLama의 이러한 특성은 다양한 자연어 처리 작업에 적용되어 언어의 깊이 있는 이해를 가능하게 한다.

III. 데이터셋 구축

데이터셋을 구축할 때, One-Line Code를 생성을 위해서는 이스케이프 문자를 처리 가능한 데이터 형식으로 변경하거나, 토크나이저의 수정이 요구된다. 따라서 Fig. 3과 같은 코드가 필요하다. 위와 같은 코드 수정을 통해 익스플로잇 코드는 한 줄의 코드로 표현이 되고, 정형화된다.

```
lines = [“a sa”, “\n”, “\t”]
tok = tokenizer()
print(tok.encode(lines))
print(tok.decode(tok.encode(lines)))
```

Fig. 3. One-Line-Code

3.1 Tokenizer를 통한 SQL Injection 구문 학습

SQL Injection이란 코드 인젝션 기법의 일환으로, 공격자가 클라이언트의 입력값을 조작하여 데이터베이스에 악의적인 SQL쿼리를 주입하여 공격한다. 본 논문에서는 NLP 기반의 익스플로잇 자동화에 대하여 정형화된 SQL Injeciton 구문을 기반으로 Tokenizer를 훈련시키고, 다양한 단어 크기의 어휘 크기를 갖는 모델을 구축한다. 사용된 SQL Injection 구문은 SQL Injection cheat sheet의 다양한 구문을 정

형화한 것이다. 이후 수집된 데이터를 전처리하여 HTML 태그, JavaScript 코드 등의 특수 문자를 제거하고 대체하여 SQL Injection 페이로드를 정제된 텍스트로 생성한다.

또한, 중복된 페이로드를 제거하고 데이터의 균형을 유지하는 등의 작업을 수행한다. 마지막으로, 전처리된 SQL 페이로드를 Tokenizer를 사용하여 토큰으로 분할한다. 이 과정에서는 페이로드를 단어, 구두점 그리고 다른 의미 단위로 나누어 각 토큰을 생성한다. 모델은 데이터셋의 크기와 메모리 제약을 감안하여 8000, 10000, 12000, 14000로 어휘 크기를 각각 설정하였다. 이러한 과정을 통해 생성된 Tokenizer를 사용하여 다양한 SQL Injection 구문을 학습함으로써 NLP 기반의 익스플로잇 자동화에 필요한 모델을 구축할 수 있다.

3.2 Tokenizer를 통한 XSS구문 학습

Cross-Site Scripting(XSS)는 웹 응용 프로그램의 보안 취약점으로, 공격자가 웹 페이지에 악성 스크립트를 삽입함으로써 클라이언트 측에 스크립트를 실행시키는 공격기법이다. 본 논문에서는 NLP 기반의 익스플로잇 자동화를 위해 Tokenizer를 사용하여 다양한 XSS 구문을 학습한다. Tokenizer를 통한 XSS 구문 학습 과정은 다음과 같다. 사용된 XSS 구문은 github, OWASP cheat sheet, portswigger cheat sheet의 여러 구문을 기반으로 정형화되었다. 수집된 데이터는 전처리 과정을 통해 HTML 태그와 JavaScript 코드와 같은 특수 문자를 제거하고 대체함으로써 XSS 페이로드를 정제된 텍스트로 생성한다. 또한, 중복된 페이로드를 제거하고 데이터의 균형을 유지하기 위한 작업을 수행한다. 모델은 SQL과 같이 데이터셋 크기와 메모리 제약을 고려하여 8000, 10000, 12000, 14000의 어휘 크기를 가진 Tokenizer를 사용하여 학습된다. 마지막으로, 전처리된 XSS 페이로드는 Tokenizer를 사용하여 토큰으로 분할된다. 이 과정에서는 페이로드를 단어, 구두점, 다른 의미 단위로 나누어 각 토큰을 생성한다. 모델은 각각 다른 어휘 크기를 가진 Tokenizer를 통해 학습된다. 이와 같은 과정을 통해 Tokenizer를 사용하여 다양한 XSS 구문을 학습함으로써 NLP 기반의 익스플로잇 자동화에 필요한 모델을 구축할 수 있다.

IV. 모델 구현

악성코드를 사전에 훈련시키기 위해, LLama와 유사한 구조를 가진 모델을 활용하여 언어 모델을 구성하였다.

4.1 토크나이저

악성코드의 토크나이징을 위해 구조적 특성이 영어 문장과 유사하다는 점을 고려하여 bpe 알고리즘[1]을 지원하는 sentencepiece를 사용하였다.

4.2 모델 아키텍처

4.2.1 Attention 메커니즘

LLama 아키텍처에서 큰 크기의 언어 모델에 Self Attention 메커니즘과 Grouped Query Attention(GQA)를 적용하였고, 작은 크기의 모델에서는 Multi-Head Attention을 사용하였다. 이를 바탕으로 본 논문에서는 사전 학습(pretraining)하고자 하는 언어 모델에 Multi-Head Attention을 적용하였다.

4.2.2 포지셔널 임베딩

[2]는 포지셔널 임베딩에 Rotary Embedding 방식을 사용하였다. 반면 [3]에서는 기존의 Relative Position Encoding(RPE) 방식이 주로 덧셈을 기반으로 하는 것과 달리, 복소수로의 변환과 회전을 통해 포지셔널 임베딩을 효과적으로 주입할 수 있는 접근법을 사용하였다. 이 방법은 장기적인 의존성을 유지하며 Efficient Transformer 모델을 구현할 수 있다는 장점을 제공한다. 따라서 해당 포지셔널 임베딩 기법을 채택하여 사용하였다.

4.2.3 활성화 함수

LLama 모델에서는 기존 PaLM에서 사용된 4d 대신 2/3 * 4d의 파라미터를 적용한 변형된 SWIGLU를 활용하여 활성화 함수를 적용하였다[4]. 따라서 LLama 아키텍처에서 활성화 함수는 SWIGLU를 활용하였다.

4.3 옵티마이저

LLama 아키텍처에서는 AdamW 옵티마이저가 사용되었다. 하지만 본 연구에서는 LLama에서 적용된 AdamW와 더불어 Adam 옵티마이저를 추가로 사용하여 두 가지 방식으로 모델 트레이닝을 진행하였다.

V. 익스플로잇 코드 생성 및 결과

5.1 익스플로잇 코드 생성

익스플로잇 코드 시뮬레이션을 위해, 학습된 모델을 generate함수를 통해 익스플로잇 코드를 생성하였다. 이때 생성된 SQL injection 및 XSS 구문은 실제 취약한 웹 애플리케이션에서 적용될 수 있다.

5.2 SQL Injection 코드 생성

LLama 모델을 활용하여 생성된 SQL Injection 코드는 Fig. 4와 같다. 이 코드는 주어진 입력에 대해 공격자가 데이터베이스에 액세스하거나 조작할 수 있는 쿼리 구문을 포함한다. 생성된 코드는 일반적인 SQL Injeciton 시나리오에서 사용될 수 있으며, 이를 통해 생성된 익스플로잇 코드는 SQL Injection 취약점을 자동으로 점검하는 데 사용된다.

```
SELECT id FROM sysusers where id='admin' or ' '
SELECT id FROM sysusers where id=' ' or '7=7'
SELECT id FROM sysusers where id='admin or '0'
SELECT id FROM sysusers where id=' ' or 7=7#
SELECT id FROM sysusers where id='admin' UNION SELECT'
```

Fig. 4. SQL Injection Query

5.3 Cross Site Script (XSS) 코드 생성

XSS 코드 또한 LLama 모델을 활용하여 생성되었으며, 생성된 XSS 코드는 Fig. 5와 같다. 이 코드는 주어진 입력에 대해 악의적인 스크립트를 웹 페이지에 삽입하여 클라이언트 측 스크립트를 실행시키는 공격기법을 포함하고 있다. 생성된 코드는 일반적인 XSS 시나리오에서 사용될 수 있으며, 웹 애플리케이션에서의 보안 취약점을 자동으로 점검하는 데 활용된다.

```
generate(llama)

['<style>@keyframes x{}</style><sub style="animation-name:x" onanimationstart="alert(1)"></title></STYLE>',
 '<style>@keyframes x{}</style><tbody style="animation-name:x"oW>!>x>=confirm()>',
 '<style>@keyframes x{}</style><marquee style="animation-name:x" onanimationstart="alert(1)"></marquee>=</div>',
 '<style>@keyframes x{}</style><slot style="animation-name:x" onanimationend="alert(1)"></slot>>>>',
 '<style>@import"http://ha.ckers.org/xss.css";</style>!#x/W')}..(1)">=alert']
```

Fig. 5. XSS code

VI. 결론

본 논문에서는 NLP(Natural Language Processing) 기술을 활용하여 자동으로 익스플로잇 코드를 생성하고 개선하는 방법을 제시한다. 최근 제로데이 취약점의 발견과 악용으로 인해 보안 패치 이전에 악용되는 문제를 해결하기 위해, 토크나이저와 LLama 모델을 활용하여 SQL Injection과 Cross Site Scripting(XSS) 공격에 대한 익스플로잇 코드를 자동으로 생성하고 개선하는 방법을 제안한다. 실제로, 제안한 모델은 사전에 취약점을 점검하고 공격을 탐지하는 데 매우 유용한 도구로 기능한다. SQL Injection 및 XSS 공격의 취약점을 자동으로 식별하고 이에 대한 익스플로잇 코드를 생성함으로써, 더욱 효율적으로 보안 취약점에 대응할 수 있게 된다. 따라서, 본 연구는 자동 취약점 탐지 분야에 새롭게 기여하며, 보다 안전하고 신뢰할 수 있는 웹 애플리케이션 개발을 위한 중요한 발전을 이끌어 낼 것으로 기대된다.

[참고문헌]

[1] Rico Sennrich, Barry Haddow and Alexandra Birch, Neural Machine Translation of Rare Words with Subword Units, arxiv, arXiv:1508.07909, Aug, 2015

[2] Jianlin Su, Yu Lu and Shengfeng Pan, RoFormer: Enhanced Transformer with Rotary Position Embedding, arxiv, arXiv:2104.09864, Nov, 2023

[3] Hugo Touvron, Thibaut Lavril and Gautier Izacard, LLaMA: Open and Efficient Foundation Language Models, arxiv, arXiv:2302.13971, Feb, 2023

[4] Noam Shazeer, GLU Variants Improve Transformer, arxiv, arXiv:2002.05202, Feb, 2020