졸업과제 착수 보고서

NLP 기반 Powershell 악성 스크립트 탐지 툴

지도교수	최윤호
팀 명	어쩔 보안
이름	201845108 김서영
	201924481 박지연
	201714537 최연재

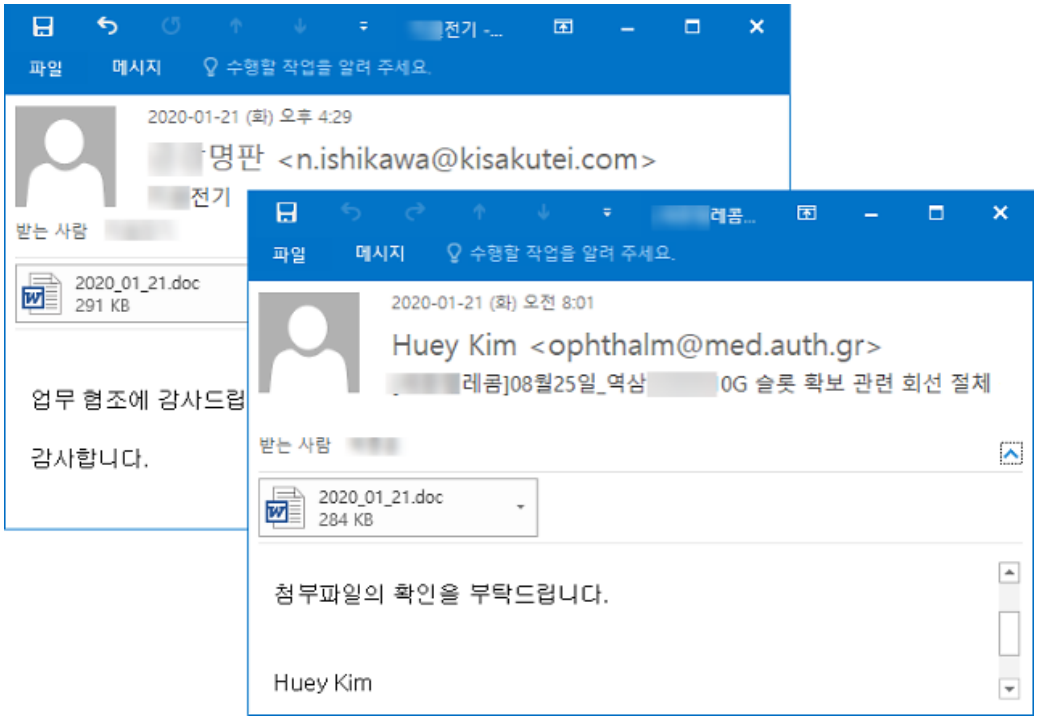
목차

1. 과제 배경 및 목표

1.1 과제 배경

PowerShell 은 시스템 관리자를 위해 특별히 설계된 Windows 명령 줄 셸로 운영체제와 프로세스를 관리하는 기능의 대부분을 제어할 수 있는 API 와 단일 함수 명령 줄 도구(cmdlet, Command-Let)를 가지고 있다. 비교적 짧은 소스코드로 효과적인 성능을 낼 수 있기 때문에 사이버 공격자에 의해 자주 사용된다.

대표적으로 2014 년부터 활동한 악성 코드인 '이모텟(Emotet)'이 피싱 메일을 통해 국내에 대량 유포된 사건이 있다. 이처럼 사이버 공격자들은 지속적으로 탐지 기술을 우회할 수 있는 방안을 찾고 있지만 기존의 시그니처 기반 탐지 방법을 사용해서는 사이버 범죄 행위를 발견하기 어렵다. 때문에 난독화된 파워쉘 기반 악성스크립트를 비난독화하고 머신러닝 모델을 활용하여 악성 스크립트 탐지 결과를 찾아내고자 한다.



악성코드 '이모텟' 예시

1.2 과제 목표

본 졸업 과제는 자연어 기반의 Powershell 악성 스크립트 탐지 툴 개발을 목표로 한다. 비난독화 도구를 통해 난독화된 powershell 스크립트를 분석한다. 비난독화한 스크립트를 RNN 과 RF 를 활용한 악성 스크립트 분류 모듈로 정상인지 악성인지 판별한다. 최종적으로 비난독화된 스크립트와 악성 스크립트 탐지 결과를 프로그램으로 나타내고자 한다.

2. 요구 조건 분석

2.1 비난독화 모듈

- 비난독화를 위해서 'PowerDecode'와 'Python'을 사용한다.
- 'PowerDecode'를 이용하여 'String concatenate/reorder/reverse/replace, Base64 encoding, ASCII encoindg, Compression deflate/GZIP' 방식으로 난독화된 스크립트를 비난독화 한다.
- 'Python'을 이용하여 'UTF-8, EUC-KR'[1]의 방식으로 난독화된 스크립트를 비난독화 한다.

2.2 데이터 전처리

- 스크립트에서 필요 없는 부분을 제외하기 위해 난독화가 제거된 악성 스크립트와 정상스크립트의 데이터 전처리 과정을 수행한다.
- 전처리 과정은 스크립트의 토큰화 과정과 word embedding 을 통한 벡터화 과정이다.
- 토큰화 과정

[1] python 의 base64 에서 지원해주는 라이브러리

- ‘a’ – ‘z’, ‘A’-‘Z’, ‘*’(숫자), ‘$’(변수참조), ‘-‘ 외의 기호를 구분기호로 사용한다.
- 파워쉘에서 단일문자 자체는 의미가 없기때문에 길이가 2 이상인 토큰만 사용한다
- 또한 파워쉘은 대소문자를 구별하지 않기 때문에 모든 토큰은 소문자로 정규화한다.
- 토큰화는 keras 에서 제공되는 Tokenizer 을 이용한다.

- 토큰화된 데이터는 keras 의 embedding()을 이용하여 임베딩 벡터로 만든다.

2.3 악성 스크립트 분류 모듈

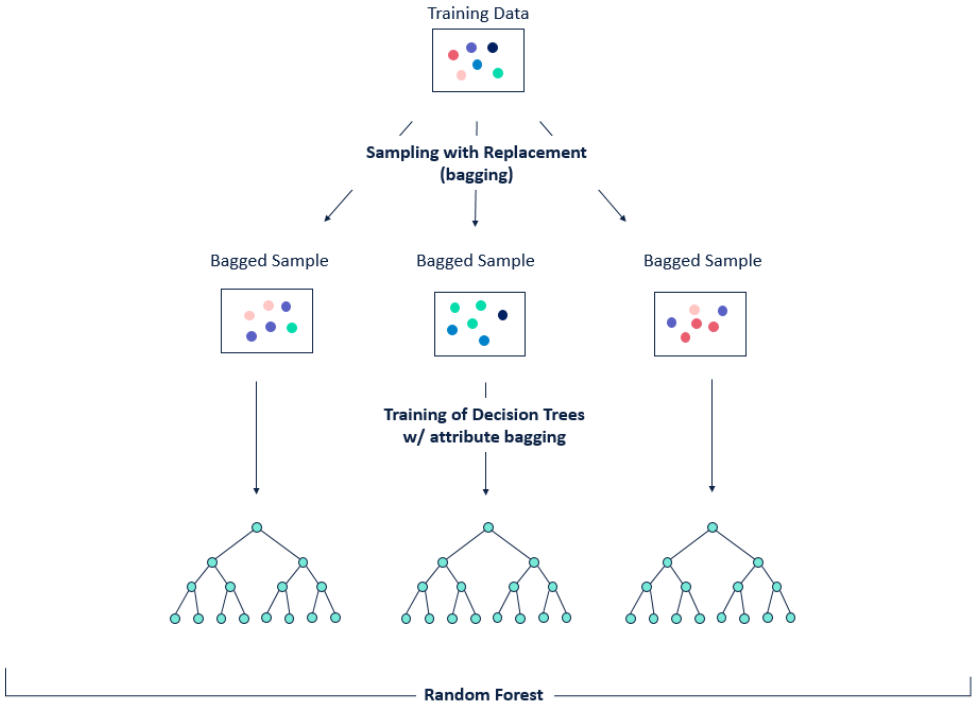
- 입력한 스크립트가 악성 스크립트인지 정상 스크립트인지 판별하는 기능
- 주어진 데이터를 라벨에 따라 분류하는 모델을 개발한다. 본 모듈은 악성 스크립트와 정상 스크립트를 분류하는 모듈이다. 따라서 이진 분류 모델을 개발해야 한다.
- 여러 개의 결정 트리를 활용하여 분류하는 랜덤 포레스트(RF)모델과 인공 신경망 중 RNN 모델을 사용하고자 한다. 두개의 모델 중 더 성능이 좋은 모델의 결과를 출력한다.
- RF 와 RNN 모델 선정이유는 RF 의 경우 알고리즘 구현이 간단하고 기존 Decision tree 의 오버피팅 문제점이 개선되었기 때문에 선정했다. 하지만 RF 만 구현했을 때 원하는 성능이 안나오는 경우를 대비하기위해 RNN 모델을 추가적으로 구현한다. RNN 은 순차 데이터에 특화된 인공신경망으로 문자열 처리에 더 적합하다 생각하여 선택하였다.

3. 연구 방향

3.1 악성 스크립트 분류 모듈

악성 스크립트 분류 모델을 개발하기 위해서는 RF 와 RNN 에 대한 이해가 필요하다.

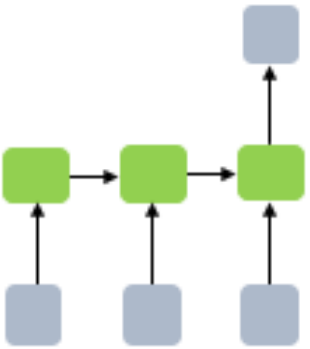
- RF(Random Forest)



- 랜덤 포레스트는 여러 개의 결정 트리를 활용한 앙상블 머신러닝 모델이다. 트리들의 분류를 집계하여 최종적으로 분류하는 모델로 임의의 숲을 구성하여 다수의 트리들로부터 분류를 집계하기 때문에 의사결정 트리의 오버피팅 한계를 극복할 수 있다.

- 트리를 만들때에는 중복을 허용하여 모든 속성(feature)들에서 임의로 일부를 선택하고 그 중 정보 획득량이 가장 높은 것을 기준으로 데이터를 분할한다. 다만 트리를 만들때 사용될 feature 을 제한함으로써 각 트리들에 다양성을 제공한다.
- Sicikt-learn 라이브러리를 통해 모델을 구현하고 데이터를 학습시킨다.

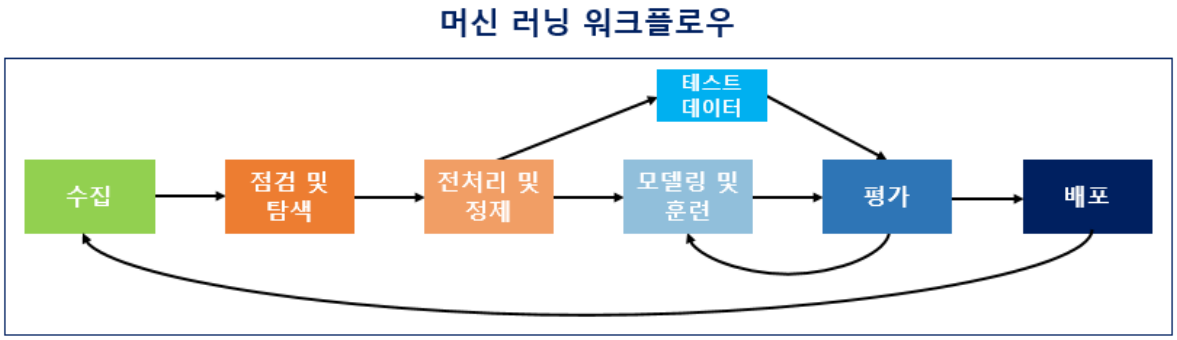
RNN(Recurrent Neural Network)



다 대 일(many-to-one)

- RNN 은 순차 데이터 처리에 특화된 인공 신경망이다. 주로 순서가 있는 데이터, 즉 문장 데이터를 입력해 문장 흐름에서 패턴을 찾아 분류하게 한다.
- 입력과 출력의 길이를 다르게 설계하여 여러 용도로 사용할 수 있다. 다 대 일(many-to-one)구조를 사용하여 이진 분류 모델을 구현하고자 한다.
- 다 대 일(many-to-one)은 입력 값으로서 데이터 시퀀스를 가지고, 단일 출력 값을 예측하는 모델로 감성분류(입력 문서가 긍정인지 부정인지 판별)에 적합하기 때문에 악성 스크립트 분류모델로 선택했다.

3.2 머신 러닝 워크플로우



머신 러닝에서 거치는 전반적인 과정에 대해 알아보고자 한다.

크게 수집, 점검 및 탐색, 전처리 및 정제, 모델링 및 훈련, 평가, 배포 단계로 나눌 수 있다.

먼저 데이터를 수집하는 과정을 거친다. 본 과제에서는 ps1 파일 형식의 데이터를 수집하고자 한다. 수집한 데이터의 구조를 파악하고 어떻게 정제할지 파악한다. 이때 그래프로 시각화하여 분석하고자 한다.

분석 결과를 바탕으로 데이터를 전처리한다. 토큰화, 정제, 정규화 등의 단계를 거친다.

모델을 구현하고 데이터를 모델에 적용하여 학습시킨 뒤 테스트 데이터로 모델의 성능을 평가한다. 성능이 좋지 않다고 판단되면 이전 단계로 돌아간다.

이러한 과정을 통해 악성 스크립트 분류 모듈을 개발하고자 한다.

4. 현실적 제약 사항 및 대책

4.1 제약 사항

1. 정상 스크립트에 비해 악성 스크립트의 수가 적어 비대칭적으로 학습될 수 있다.

2. PowerDecode 와 Python 을 이용해서는 'String concatenate/reorder/reverse/replace, Base64 encoding, ASCII encoding, Compression deflate/GZIP, UTF-8, EUC-KR'로 인코딩 된 것들만 비난독화 할 수 있다.

3. 학습데이터 수집 및 User 에서 제공되는 스크립트가 난독화 스크립트와 비난독화 스크립트가 섞여 제공된다.

4.2 대책

1.1 Virus Share, Git Hub, Hybrid Analysis, Posh Code, Git Hub, PowerShell Gallery 사이트를 이용하여, 정상 스크립트와 악성 스크립트를 수집한다.

1.2 정상 스크립트 개수를 악성 스크립트 수에 맞추어 수집하거나 악성 스크립트 데이터를 복제하여 비율을 동일하게 맞춘다. 가중치를 다르게 주는 것 또한 한 가지 방법이다. 이 중 가장 성능이 높은 방법을 채택한다.

2. 여러 개의 비난독화 툴을 사용하여 다양한 방식으로 난독화 된 스크립트를 비난독화한다.

3.1 학습데이터는 우선적으로 비난독화된 데이터를 수집한다. 난독화된 경우 비난독화 툴을 이용해 비난독화 시킨다.

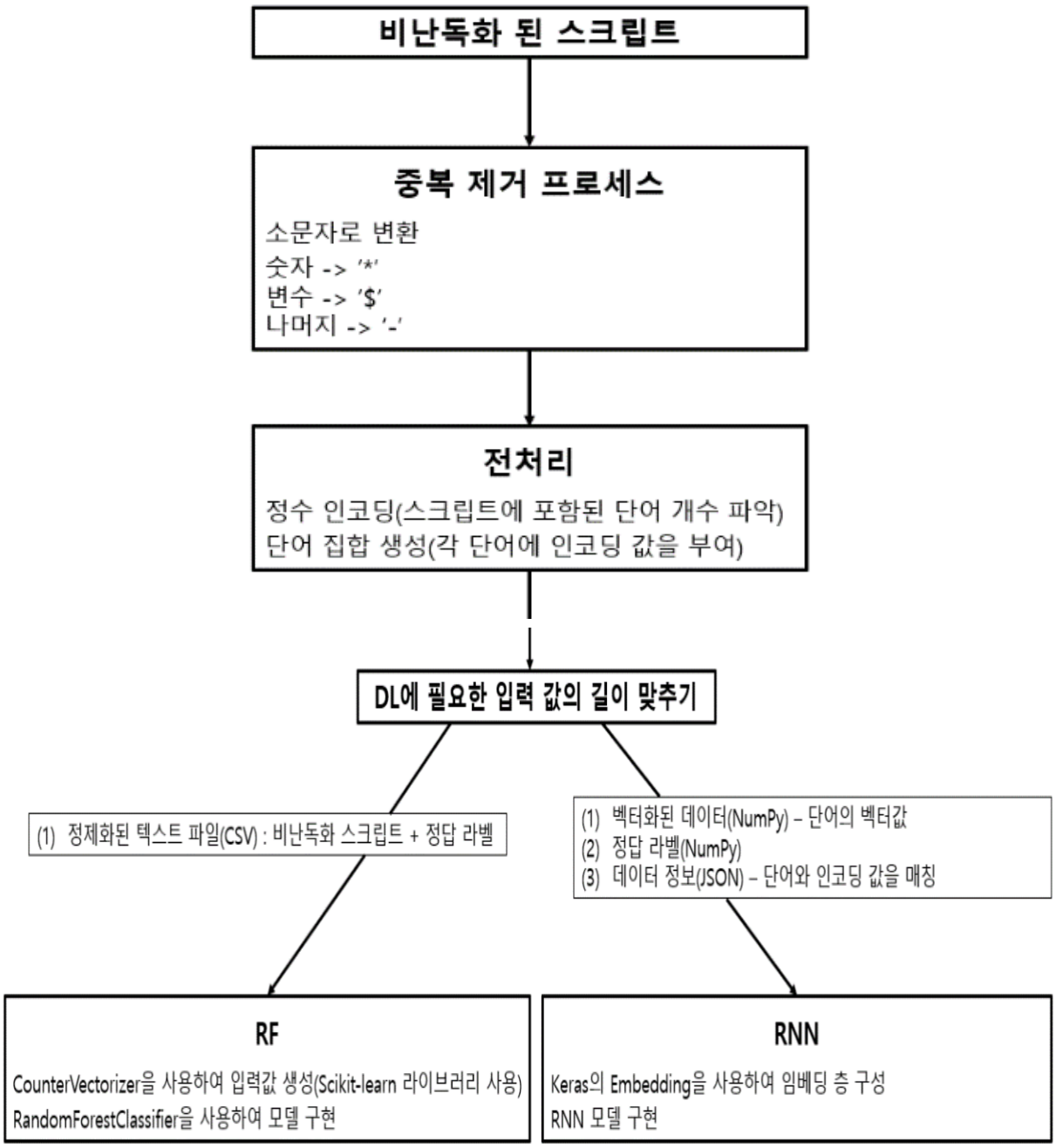
3.2 User 에서 받아오는 스크립트의 경우 예시로 비난독화 툴인 PowerDecode 에서는 비난독화 스크립트가 주어졌을 때 스크립트 그대로 출력되면서 비난독화에 실패했다는 Error message 를 출력함으로써 난독화 여부를 판단할 수 있다.

5. 설계 문서

5.1 시스템 구성도

- 실제 동작 : User 입장에서 그린 흐름도

- 모델 학습


비난독화 된 스크립트
중복 제거 프로세스
소문자로 변환
숫자 -> '*'
변수 -> '$'
나머지 -> '-'
전처리
정수 인코딩(스크립트에 포함된 단어 개수 파악)
단어 집합 생성(각 단어에 인코딩 값을 부여)
DL에 필요한 입력 값의 길이 맞추기
(1) 정제화된 텍스트 파일(CSV) : 비난독화 스크립트 + 정답 라벨
(1) 벡터화된 데이터(NumPy) – 단어의 벡터값
(2) 정답 라벨(NumPy)
(3) 데이터 정보(JSON) – 단어와 인코딩 값을 매칭
RF
CounterVectorizer을 사용하여 입력값 생성(Scikit-learn 라이브러리 사용)
RandomForestClassifier을 사용하여 모델 구현
RNN
Keras의 Embedding을 사용하여 임베딩 층 구성
RNN 모델 구현

5.2 개발 환경

개발 언어	Python3
개발 도구	Tensorflow2, Colab, PowerDecode, PowerDrive
라이브러리	Keras, Scikit-learn, Numpy, CSV
파일 형식	Txt, ps1, csv, json
실행 환경	Windows10

5.3 사용 기술

➢ **Tensorflow**

- 텐서플로우는 구글이 개발한 오픈소스 소프트웨어 라이브러리이다. 딥러닝 프로그램을 쉽게 구현할 수 있도록 다양한 기능을 제공하며 Python, java, C++ 등 다양한 언어에 API 를 제공해준다.
- 페이스북에서 개발한 파이토치와 비교해보았을 때 텐서플로우는 사용자 층이 넓고 다양한 자료와 예제가 존재한다. 따라서 본 과제에서는 텐서플로우를 사용하여 RNN 모델을 구현할 계획이다.

➢ Powershell

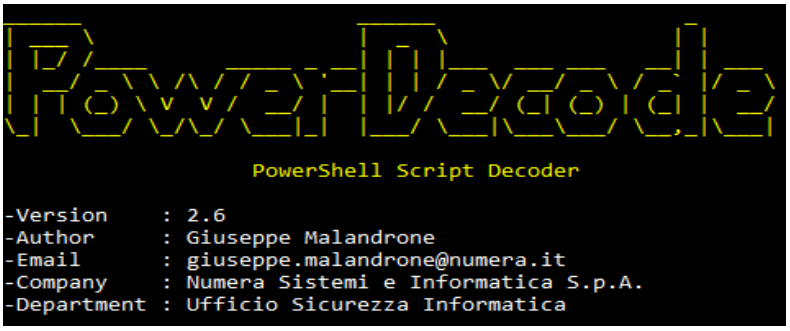
- PowerShell 은 마이크로소프트가 개발한 명령 줄 셸, 스크립팅 언어 및 구성 관리 프레임워크로 구성된 플랫폼 간 작업 자동화 솔루션으로 Windows, Linux 및 macOS 에서 실행할 수 있다.

- .NET Framework 에 기반하여 개발되었고 기존의 cmd 보다 강력한 스크립트를 지원한다. Powershell 기반 스크립트 분석에 사용한다.

➢ PowerDecode

- PowerDecode 는 다양한 난독화 형식을 통해 난독화된 파워쉘 스크립트를 비난독화하기 위한 Powershell 기반 도구이다.

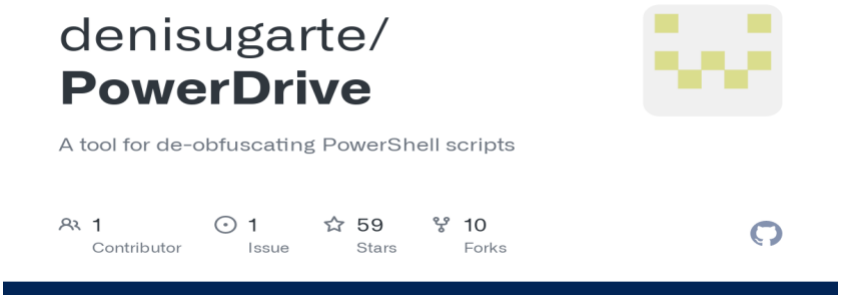
- PowerDecode 는 오픈소스이며 이번 과제에서 비난독화 툴으로 사용할 예정이다.
- 'String concatenate/reorder/reverse/replace, Base64 encoding, ASCII encoding, Compression deflate/GZIP, UTF-8, EUC-KR'로 인코딩 된 스크립트를 지원한다.



➢ **PowerDrive**

- PowerDrive 는 다양한 난독화 형식을 통해 난독화된 파워쉘 스크립트를 비난독화하기 위한 Powershell 기반 도구이다.
- 오픈소스이며 Powershell 에서 PowerDrive 모듈을 불러오는 방식으로 사용한다

- PowerDecode 의 비난독화 한계에 대한 대책방안으로서 비난독화 툴 테스트 시행 예정이다.



6. 개발 일정 및 역할 분담

6.1 개발 일정

5 월		6 월				7 월					8 월					9 월			
4 주	5 주	1 주	2 주	3 주	4 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주
스터디(비난독화, RF RNN, 자연어 처리)																			
					데이터 분석 및 전처리														
					비난독화 도구 테스트														
								중간보고서 작성											

<table>
<tr><th colspan="2">5 월</th><th colspan="4">6 월</th><th colspan="5">7 월</th><th colspan="5">8 월</th><th colspan="4">9 월</th></tr>
<tr><th>4 주</th><th>5 주</th><th>1 주</th><th>2 주</th><th>3 주</th><th>4 주</th><th>1 주</th><th>2 주</th><th>3 주</th><th>4 주</th><th>5 주</th><th>1 주</th><th>2 주</th><th>3 주</th><th>4 주</th><th>5 주</th><th>1 주</th><th>2 주</th><th>3 주</th><th>4 주</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">분류 모델 스터디 및 개발</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">모델 평가</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">GUI 만들기</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4">테스트 및 디버깅</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">최종발표/
보고서 준비</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">포스터
제작</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>최종
발표</td></tr>
</table>

6.2 역할 분담

이름	역할 분담
김서영	수집된 학습 데이터 분석 및 전처리, 딥러닝 모듈 개발
최연재	비난독화 툴 성능 분석, 학습데이터 전처리, Python GUI
박지연	딥러닝 모듈 개발, 모델 학습 및 검증
공통	학습데이터 수집 테스팅 및 디버깅 보고서 작성 및 발표