-정성균 개인팀-
정보보호 R&D 데이터 챌린지
악성코드 탐지 트랙

31th Master Course **Sungkyun Jung**

Digital Forensic Research Center

Center for Information Security Technologies

Korea University

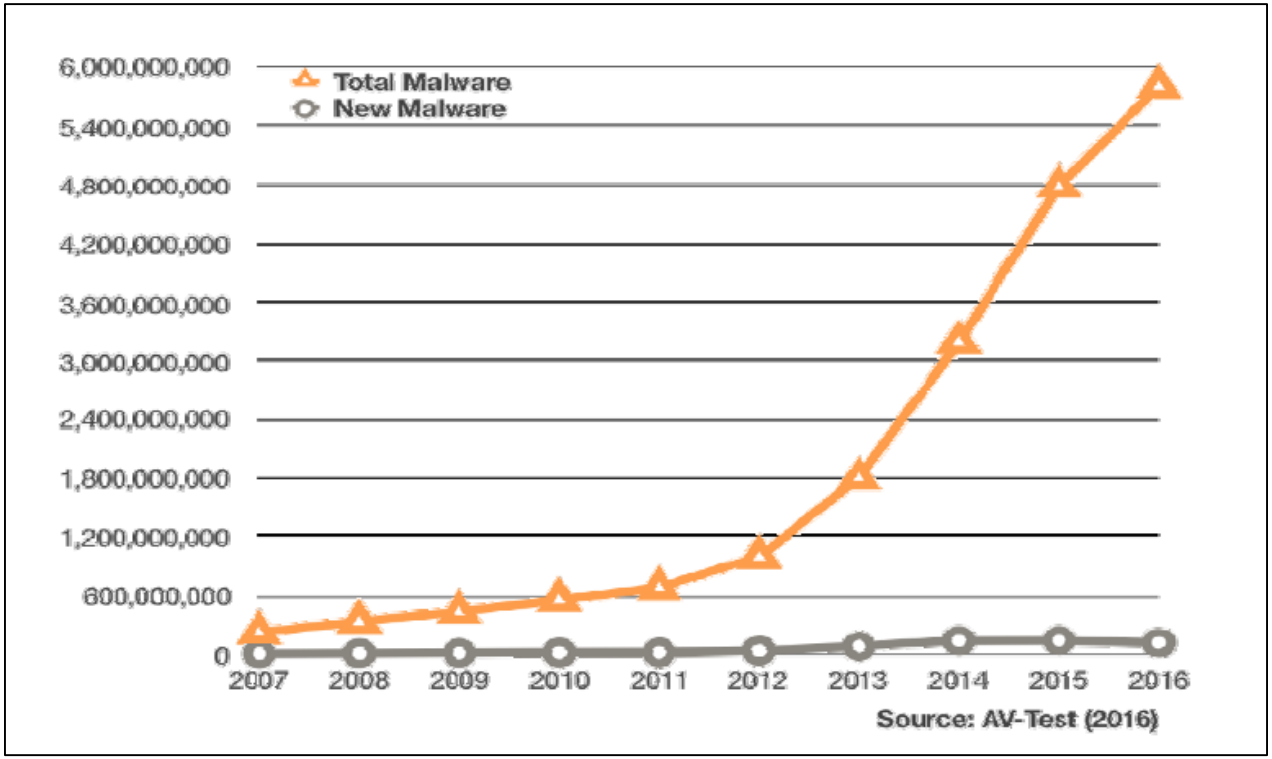
Digital Forensic Research Center
Center for Information Security Technologies
in Korea University

목차

- Overview
- 방법론
- 도구 개발 및 평가
- 데이터 분류 결과

Overview

- **악성코드의 출현**
  - AV-TEST 2017, 악성코드는 연간 기준 6억 개 이상 출현
  - 악성 코드의 출현 개수는 매년 증가
    - 10년 전과 비교해 300배 증가



Overview

- **Challenges**
  - 전문가가 막대한 양의 바이너리를 일일이 분석하는 것은 물리적으로 불가능
    - 악성 코드 탐지의 자동화
  - Malware Obfuscation
    - 새로운 malware의 80%는 packing 적용
    - 새로운 malware 중 50%는 과거의 malware를 repacking 한 형태
    - Polymorphic malware (실행 코드 암호화)
    - Metamorphic malware (고유 기능만 유지하고 모든 부분을 변경)
  - 단일 분석기법으로 대응하는데 한계 존재
    - Machine learning을 이용한 분석 동시 반영

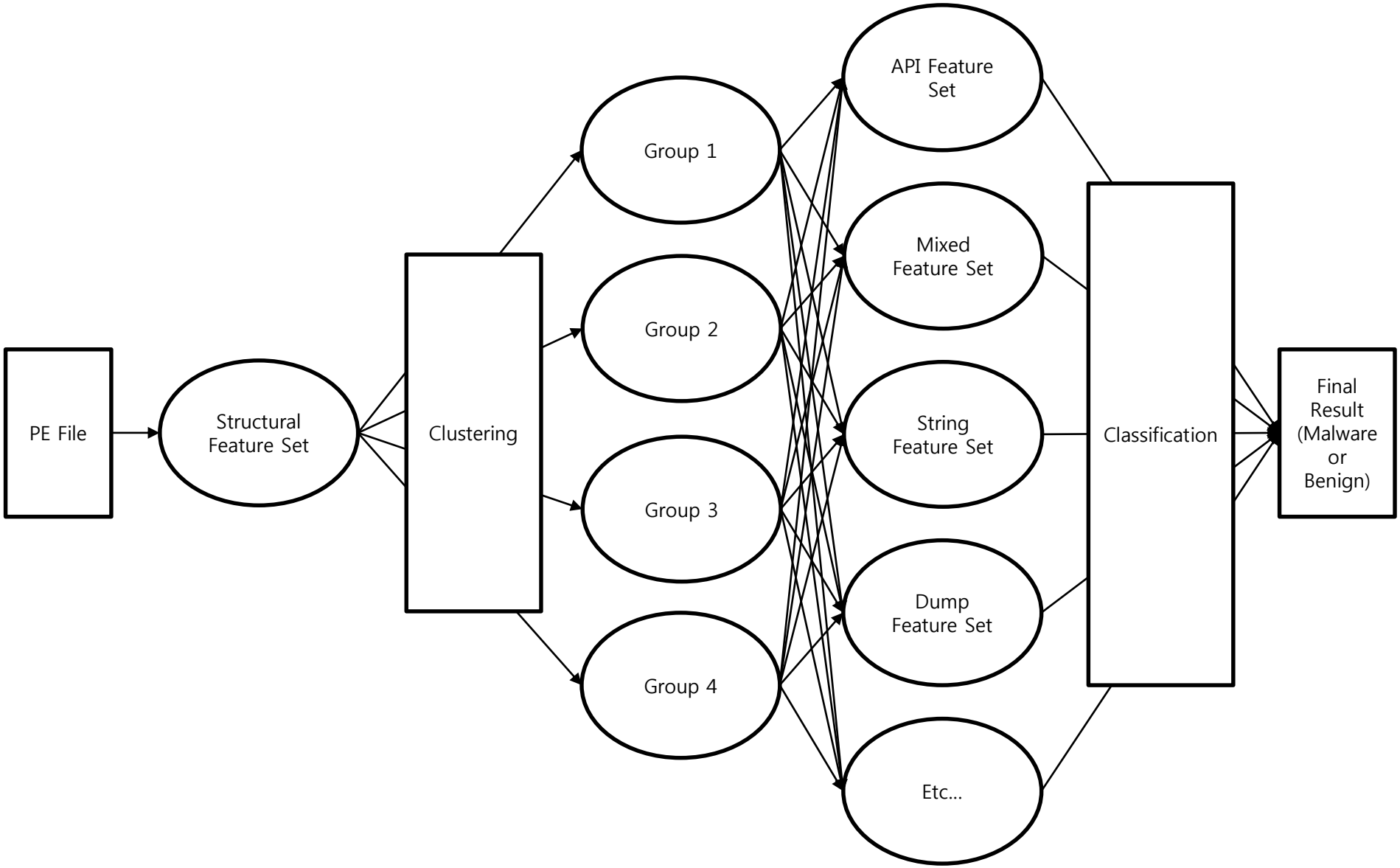
Overview

- **Malware 탐지에 이용되는 Features**
  - File size, File 확장자, 생성시간 – 기본 정보
  - Packer – Packing 여부 및 Packer 정보
  - Imported DLL, API list – DLL 및 API 호출 정보
  - PE Header – Entry Point, Compilation Time, ImageBase, etc..
  - Strings – 유의미한 ASCII 문자열
  - Operation code – frequency, sequence 기반
  - Byte sequence (hex view) – sequence 기반
  - Entropy – randomness 측정 (packing)
  - Image Representation – Byte 단위로 rgb 기반 pixel 변환

방법론

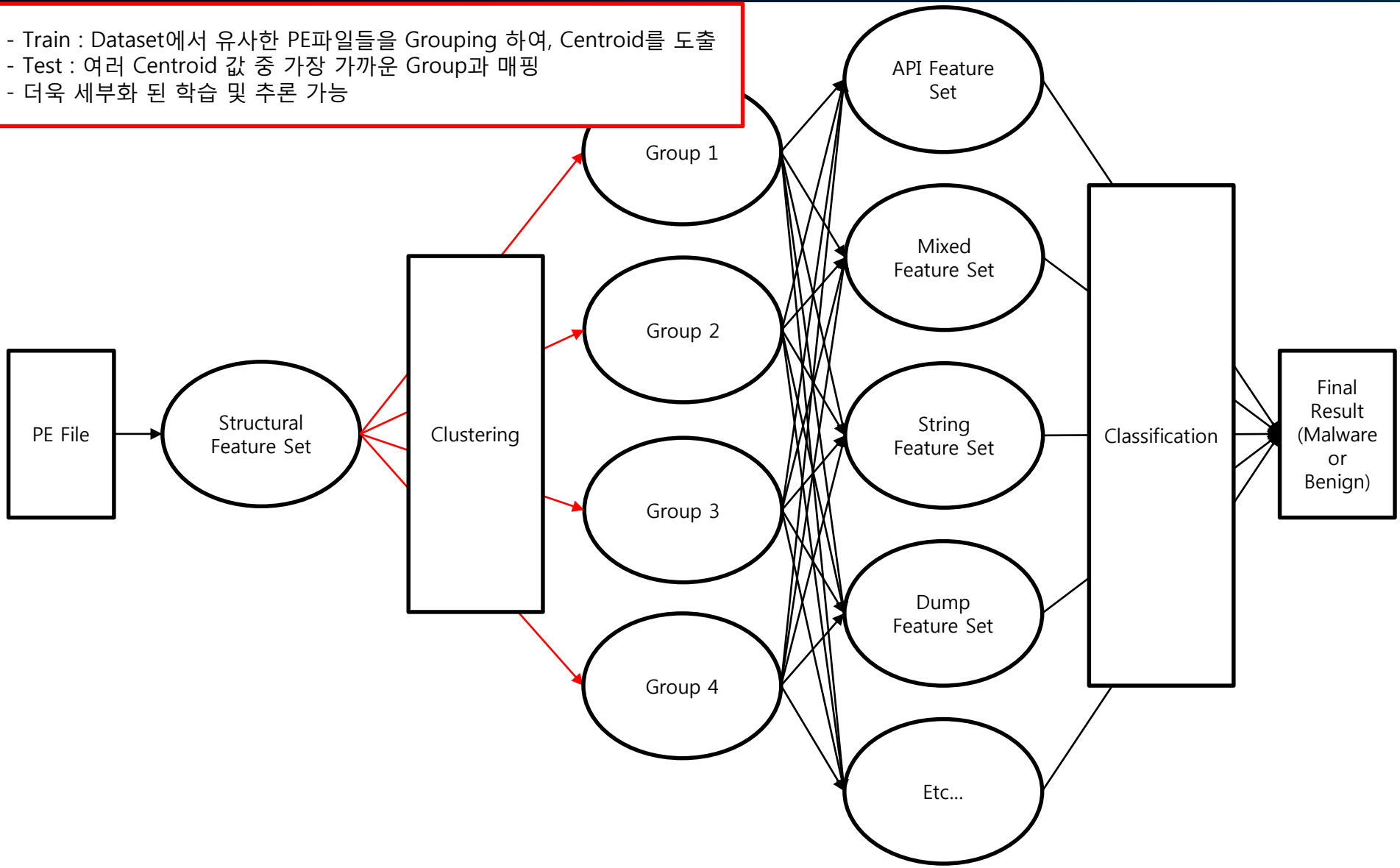
- **Multiple Feature sets 기반의 기계 학습 모델**
  - 기존 악성코드 탐지 모델의 경우, 단일 Feature set만을 사용하고 Unpacking을 가정
    - API list만으로 feature set을 구성할 경우, 악성코드 제작자는 IAT 영역을 정상 파일처럼 조작하는 것만으로 탐지 모델을 우회 가능
    - 제공된 데이터셋을 분석한 결과, 실제로 어떠한 API도 보이지 않는 PE 파일이 존재
  - 총 4개의 Feature set 사용
    - API, Mixed, String, Dump, ASM
    - 최종 추론에는 모든 Feature set의 결과를 종합
      - 정확도 및 신뢰도 향상

방법론


PE File
Structural Feature Set
Clustering
Group 1
Group 2
Group 3
Group 4
API Feature Set
Mixed Feature Set
String Feature Set
Dump Feature Set
Etc...
Classification
Final Result (Malware or Benign)

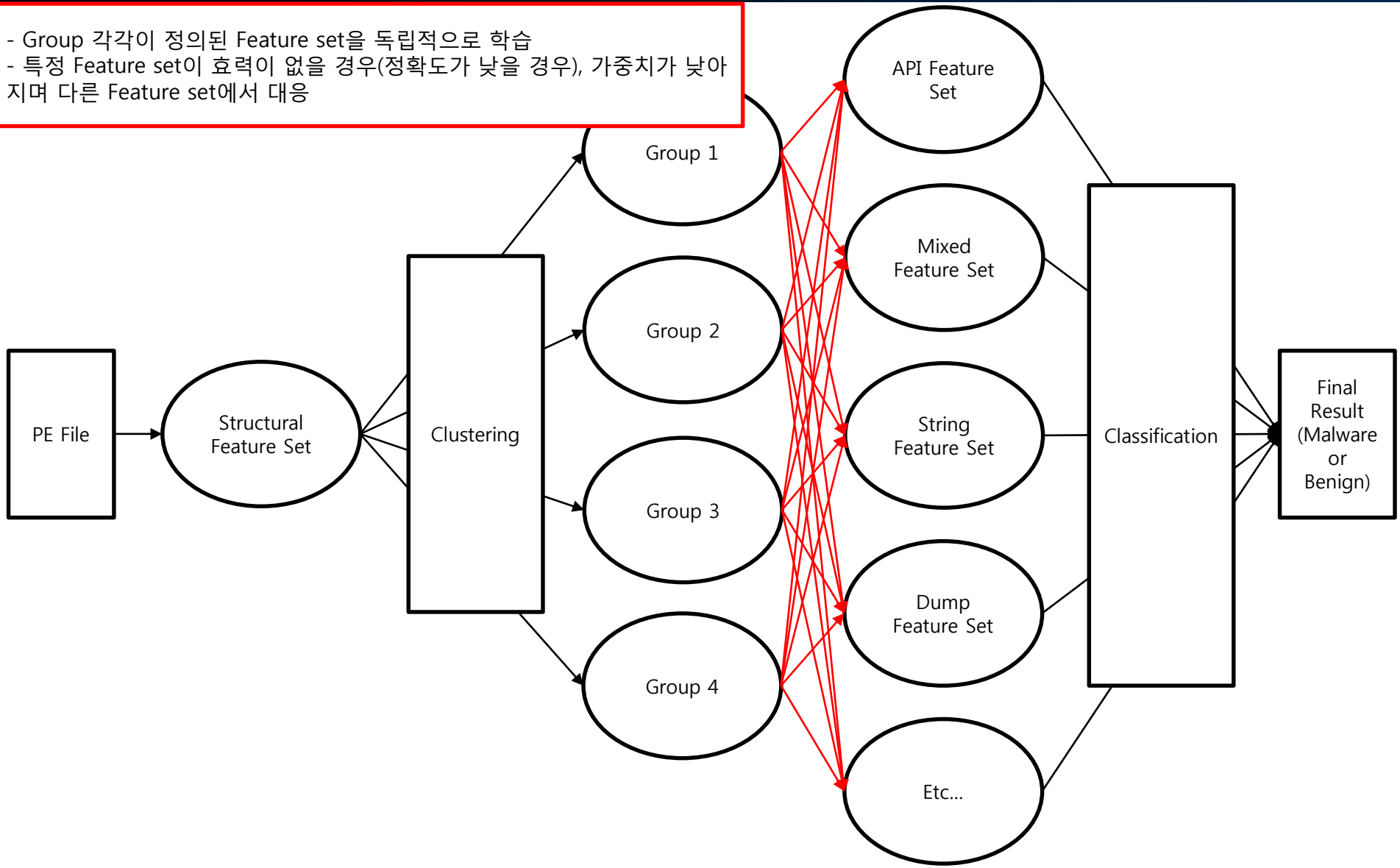
방법론

- Train : Dataset에서 유사한 PE파일들을 Grouping 하여, Centroid를 도출
- Test : 여러 Centroid 값 중 가장 가까운 Group과 매핑
- 더욱 세부화 된 학습 및 추론 가능



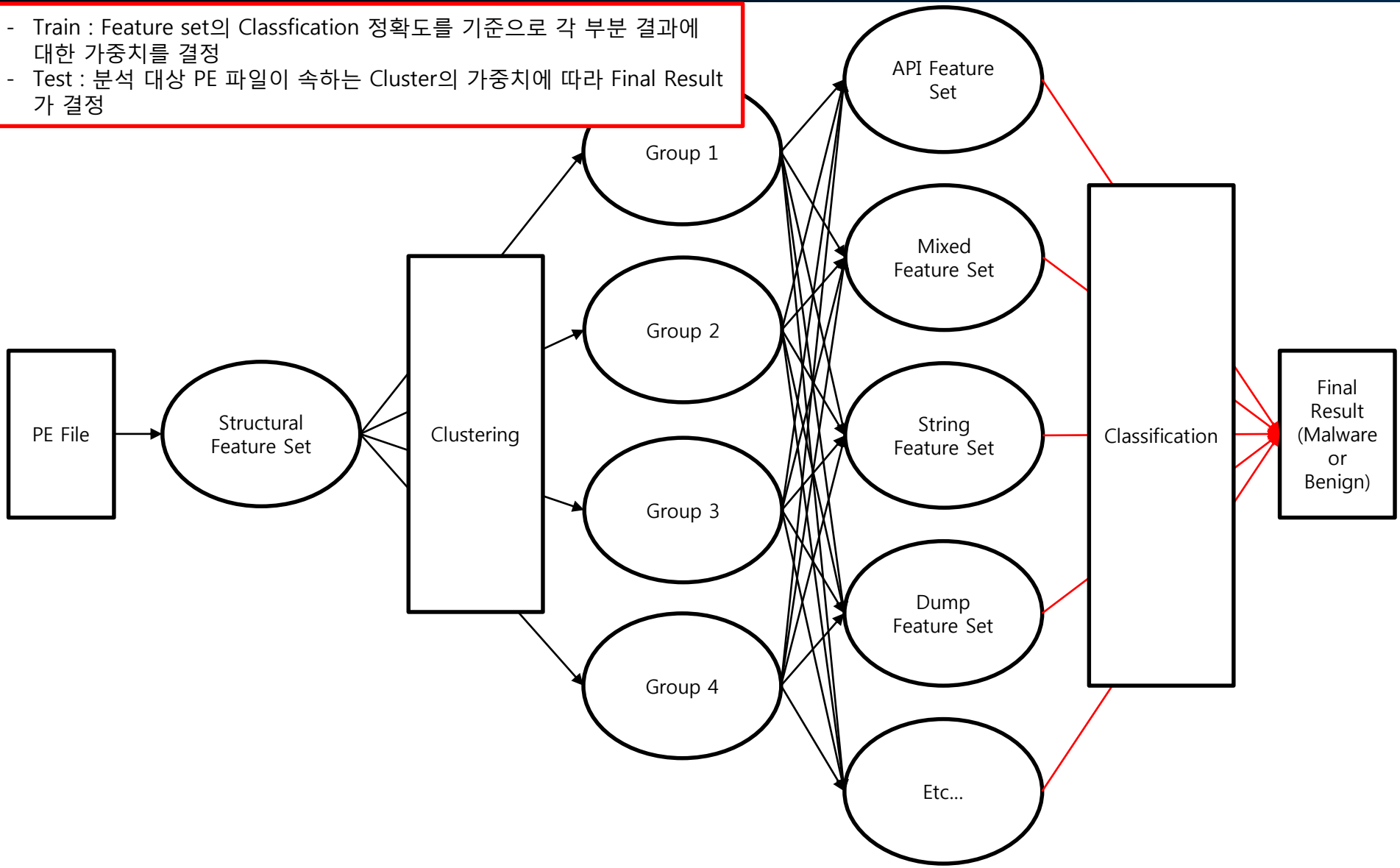
방법론

- Group 각각이 정의된 Feature set을 독립적으로 학습
- 특정 Feature set이 효력이 없을 경우(정확도가 낮을 경우), 가중치가 낮아지며 다른 Feature set에서 대응



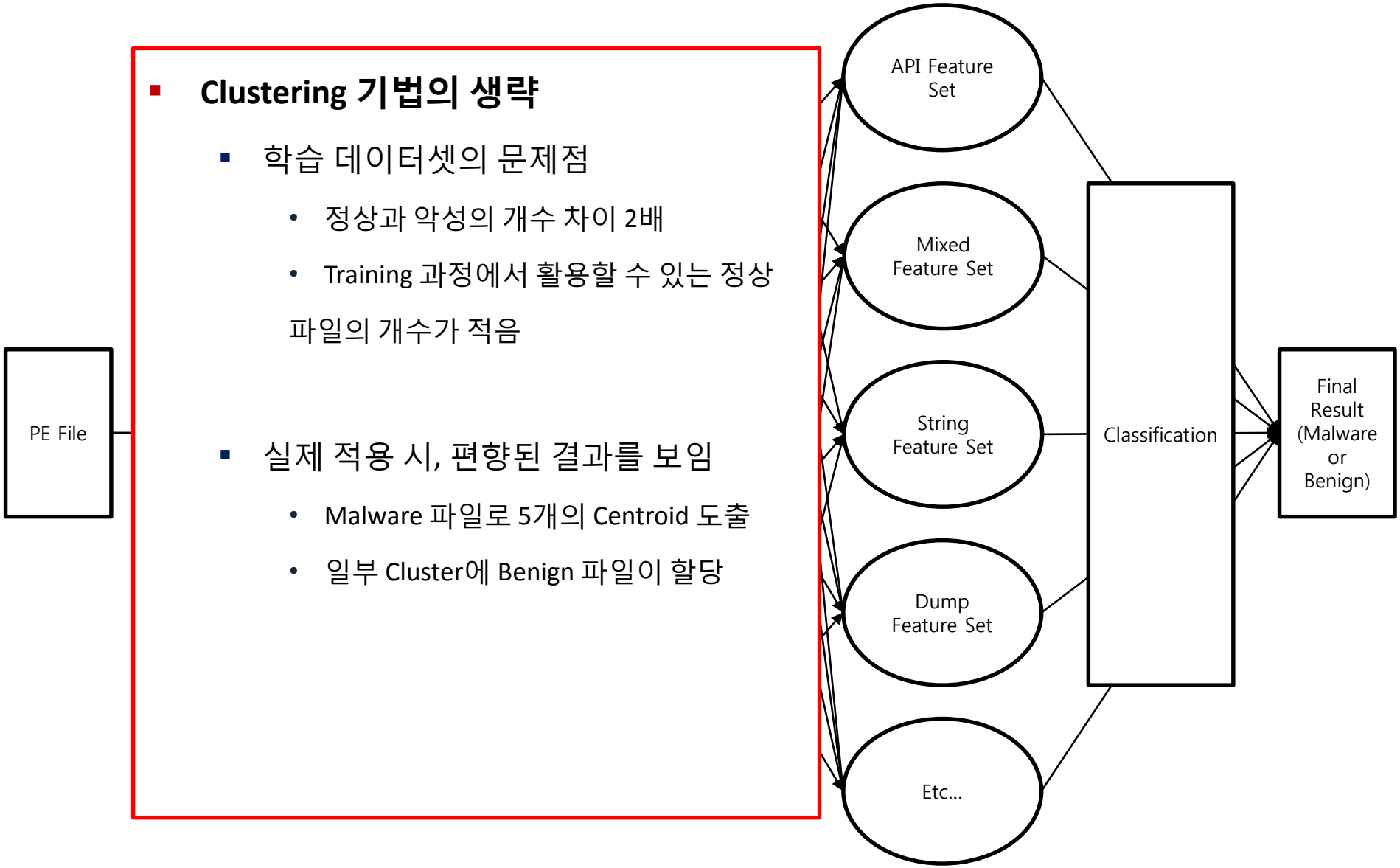
방법론

- Train : Feature set의 Classfication 정확도를 기준으로 각 부분 결과에 대한 가중치를 결정
- Test : 분석 대상 PE 파일이 속하는 Cluster의 가중치에 따라 Final Result 가 결정



방법론

- **Clustering 기법의 생략**
  - 학습 데이터셋의 문제점
    - 정상과 악성의 개수 차이 2배
    - Training 과정에서 활용할 수 있는 정상 파일의 개수가 적음
  - 실제 적용 시, 편향된 결과를 보임
    - Malware 파일로 5개의 Centroid 도출
    - 일부 Cluster에 Benign 파일이 할당



방법론

- **Feature set 정의**
  - API
    - Malware 데이터셋의 모든 PE 파일에서 IAT 조회
    - 가장 많이 호출되는 상위 API 약 1000개를 Feature set으로 정의
    - 각 PE 파일에서 Feature set에 정의된 API를 호출 시 1(True) 아니면 0(False)

	A	B	C	D	E	F	G	H	I	J	K	L
1	ExitProcess	GetProcAddress	GetLastError	CloseHandle	GetModuleHandleA	Sleep	WriteFile	RegCloseKey	GetModuleFileNameA	LoadLibraryA	ReadFile	SetFilePointer
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	1	1	0	1	1	1	0	0	1	1

  - Mixed
    - PE Header 기반의 Feature
    - EXE 또는 DLL 여부, 표준 Section의 사용 여부 등 부수적인 요소 추가 (53개 항목)
    - 일부 데이터가 pefile 라이브러리를 통해 호출되지 않아, feature 추출 실패
      - 다른 feature set에서 보완

방법론

- **Feature set 정의**
  - String
    - 각 PE 파일을 스캔하여 Printable한 문자열을 모두 추출

04ab3c25d24d8781c17264ae4f7a2f82.string

```
!This program cannot be run in DOS mode.

$ t,Fj0M(90M(90M(90M(9= M(9= M(9= M(9v 97M(995 92M(995 9%M(90M)9 M(9M4 93M(9= 91M(90M 91M(9M4 91M(9Rich0M(9 .text `.rdata @.data .rsrc
@.reloc % fiA/ EgMg LBkD JhIj PngvP I2B5 FnIn CwAAf u
hD fyQv ITsH LRQS CyMl ZBAVP IAcO FhMk dTMT OXw6 NjUR Ygsx cwA+ OxQT NDY3 YQgg I2Z3 AAAAf LAPj u
hD Jig5 LCMi P2IX KjYW Yz87P Ix0I cAcA AAANPj u
hD fzs+ NCEf fGE1 JAAA AAAHP u
hD hH0L fiA/ EgMg LBkD JhIj PngvP I2B5 FnIn CwAAf u
hD hx0L MiZi CB4C CQU8 KD06 HwNpP DmA7 JH0V CTUA AAAVPj u
hD Ag4R EAAX HRQM Dzwu JDk/P OAgM BRYZ HA0+ CGlh E2d9 ZnN6 AG4Q CXxk exZ3 eghv CmRi fXwT G2R0 C2Nm EhN9 LAYN cRgB fBVm HD8H EAst czAa
GCgi a3wc BBB/ AT0z CQAAf u
hD fyQv ITsH LRQS CyMl ZBAVP IAcO FhMk dTMT OXw6 NjUR Ygsx cwA+ OxQT NDY3 YQgg I2Z3 AAAAf LAPj u
hD MxM6 Hg0G CmQm AAAA u
hD h h81L NSIB FRtm CmI4 BQc+ HAEbP GiUE fhdn bmkM ATQ7 Oz0q YXIg HDs3 N2Bj LgIJ AB0N GAAAf u
hD hh1L MiIg AGFh Lhdp BwAA u
hD ODI9 LC4+ BTwT CGA/ AAw/P eQAk DDEx Oilz GWAi CB9o IAQE AAAAf HwPj u
hD JSIZ KDkZ BGcp ezIj Gh87P E20A AAAPPj u
hD NTEj EzYC CjIl fyUJ biUgP GAc+ PQY9 EykU JARj JB11 JycN AAAAf HgPj u
hD h(2L ISU6 CDVg AgEA AAAG u
hD hX2L JXEd dD8y KDcF OCoI AXoYP MR8C Cwgg Ezh8 KTw8 DSYJ HBM+ KS8A NgMh BxEc NnwF AAAAf LAPj u
hD AhUD Lg0J CjY+ BCB4 GRAJP GwcD ODcV NwYS eBgh bhk2 FjI2 KQg+ CTxg Kmgp AAAAf JwPj u
hD AhUH MR4v FiEg OB0D OzICP CRsD Pi06 AAAAf EgPj u
hD BA8e BgRk BTg+ NiEr JT8iP Dzp8 By5i PWN0 CycA AAAXPj u
hD GRED GwUD AQ4S CRYA BQ9sP Igcb cR9h NAVw GDU7 Hjkp JnIk chQ/ IDpk fgAAf u
```

방법론

- **Feature set 정의**
  - Dump
    - Peframe 파이썬 라이브러리의 get_dump 모듈 사용 및 추출
    - PE 파일에서의 각 주소 값에 대한 의미 해석 결과

febaeffa94702b5e5e6a6b2ea25617c4.dump

```
----------Version Information----------

[VS_VERSIONINFO]
0xA3938    0x0    Length:                    0x364
0xA393A    0x2    ValueLength:               0x34
0xA393C    0x4    Type:                      0x0

[VS_FIXEDFILEINFO]
0xA3960    0x0    Signature:                 0xFEEF04BD
0xA3964    0x4    StrucVersion:              0x10000
0xA3968    0x8    FileVersionMS:             0x80001
0xA396C    0xC    FileVersionLS:             0x17853B5
0xA3970    0x10   ProductVersionMS:          0x80001
0xA3974    0x14   ProductVersionLS:          0x17853B5
0xA3978    0x18   FileFlagsMask:             0x3F
0xA397C    0x1C   FileFlags:                 0x0
0xA3980    0x20   FileOS:                    0x40004
0xA3984    0x24   FileType:                  0x2
0xA3988    0x28   FileSubtype:               0x0
0xA398C    0x2C   FileDateMS:               0x0
0xA3990    0x30   FileDateLS:               0x0

[StringFileInfo]
0xA3994    0x0    Length:                    0x2C2
0xA3996    0x2    ValueLength:               0x0
0xA3998    0x4    Type:                      0x1
```

방법론

- **문자열 Feature(String, Dump)의 벡터화**
  - Sklearn 라이브러리의 TfidfVectorizer 모듈 사용
    - TfidfVectorizer는 문서 집합으로부터 단어의 수를 세고 TF-IDF 방식으로 단어의 가중치를 조정한 벡터를 만드는 방법
      - TF-IDF(Term Frequency – Inverse Document Frequency) 인코딩은 단어를 갯수 그대로 카운트하지 않고 모든 문서에 공통적으로 들어있는 단어의 경우 문서 구별 능력이 떨어진다고 보아 가중치를 축소하는 방법
    - String 및 Dump 각각 7,184,501와 3,617,350의 Feature 개수가 도출
  - Sklearn 라이브러리의 SelectKBest 모듈 사용
    - 임의의 k 개만큼 score가 높은 Feature를 선정하는 방법
      - String 및 Dump의 Feature 개수를 각각 5,000개로 제한

방법론

- 라벨 예측을 위한 계산 식

$$weight_{featureset}(0 \leq weight \leq 1) = Accuracy\ of\ each\ feature\ set\ in\ Training\ Phase$$

$$YPred_{featureset} = Prediction\ Value(if\ Malware\ 1\ else\ -1)\ of\ each\ feature\ set's\ Classifier$$

$$YPred_{normal} = YPred_{Api} * weight_{Api} + YPred_{String} * weight_{String} + YPred_{Dump} * weight_{Dump} + YPred_{Mixed} * weight_{Mixed}$$

$$Label = Malware\ if\ (YPred_{normal\ or\ excluded}) \geq 0\ else\ Benign$$

도구 개발 및 평가

- **평가 결과 (30% Random Test)**
  - API Feature Set = 0.909252669039
  - Mixed Feature Set = 0.956183274021
  - String Feature Set = 0.950177935943
  - Dump Feature Set= 0.948398576512
  - Merged Accuracy = 0.966859430605 // api + mixed + string + dump
  - Merged Accuracy = 0.966859430605 // mixed + string + dump
  - Merged Accuracy = 0.971975088968 // api + string + dump
  - Merged Accuracy = 0.973754448399 // api + mixed + dump
  - Merged Accuracy = 0.975756227758 // api + mixed + string

- 각 Feature set을 독립적으로 사용한 경우보다 병합한 결과가 항상 더 좋은 결과를 보임
- 일부 PE 파일에 대해 추출되지 않는 mixed feature를 제외하고, feature extraction 단계에 소모되는 시간을 절감

데이터 분류 결과

- **1차 데이터 분류 결과**
  - Malware : 6,212개
  - Benign : 1,288 개
  - 정확도 : 88.133%
- **2차 데이터 분류 결과**
  - Malware : 6,252개
  - Benign : 1,248개
  - 정확도 : 87.707%

Thank you for listening