2018 정보보호 R&D 데이터 챌린지

- AI기반 악성코드 탐지 트랙 [대학(원)생] -

Dept. of Computer Science, Kookmin University
Information Security Lab.

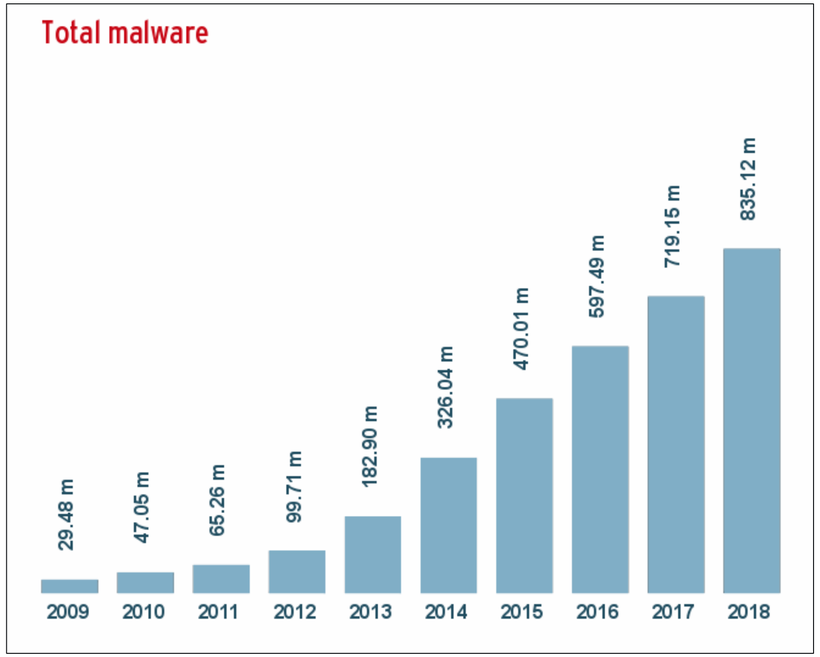
TEAM : KMU INFOSEC (발표자: 정성민)

목차

- 서론
- 악성코드 탐지 시스템
  - 딥러닝 기반 악성코드 탐지 모델
  - 엘라스틱서치 기반 최대 유사 악성코드 검색 모델
  - 머신러닝 기반 앙상블 모델
- 특징 정보
  - 정적 특징
  - 동적 특징
- 데이터셋
- 결과

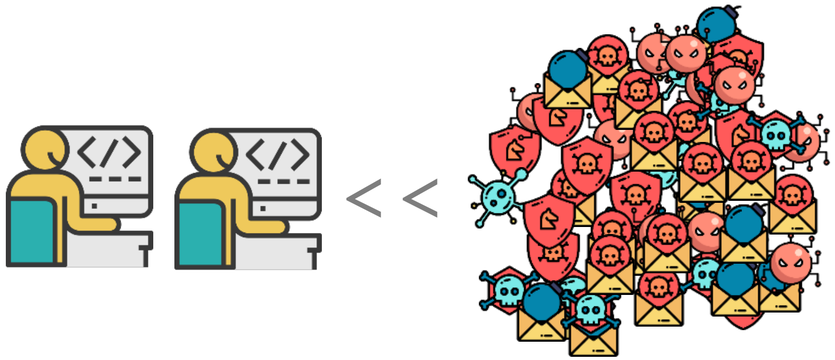
서론

- 악성코드의 기하급수적인 증가, 변종/신종 악성코드 출현
  - More than 430 Million malware were discovered in 2015 (increasing 36% in 2014)
    - Internet Security Threat Report (Symantec, 2016)
  - AV-TEST Statistics



https://www.av-test.org/en/statistics/malware/

- 그에 비해 현저히 적은 악성코드 분석가 수



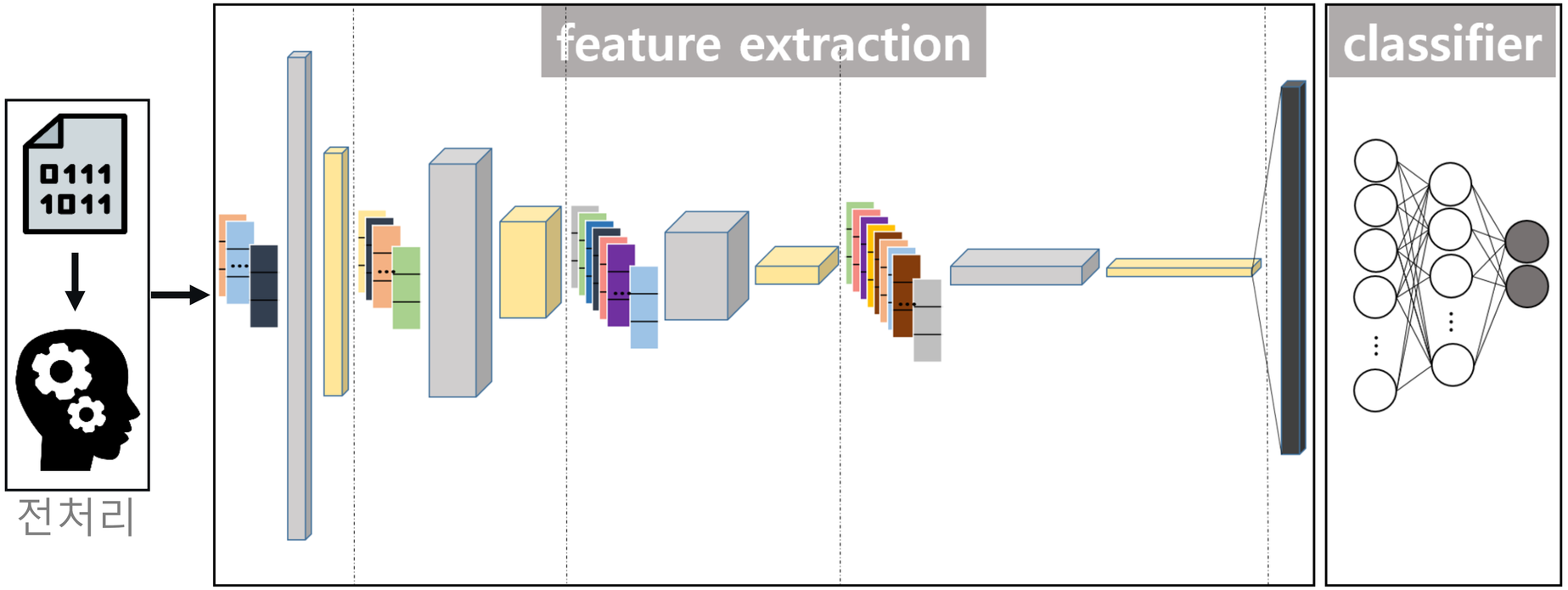
서론

- 전통적인 악성 코드의 분석 방법
  - 정적 분석: w/o running (코드 정보, 엔트로피, 헤더 정보, 파일 크기 등)
    - 특징: fast, simple
  - 동적 분석: with running (실행 정보, API call, 네트워크 통신 정보 등)
    - 특징: slow, compact(critical), robust to obfuscation

- Hand-crafted feature → Machine Learning-based approach 로의 변화
  - 자동화
  - 양질의 특징 추출

악성코드 탐지 시스템

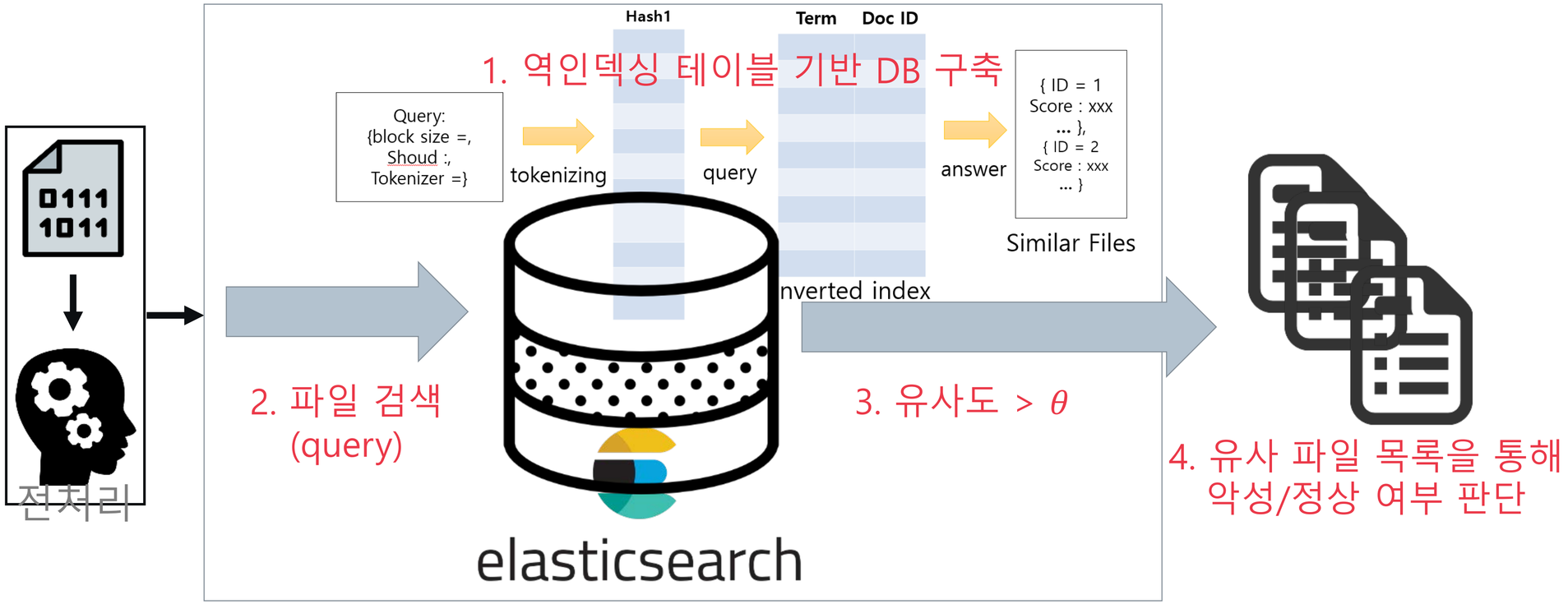
- 딥러닝 기반 탐지 모델
  - 전처리 (preprocessing)
  - CNN 기반의 딥러닝 학습 모델링



[HYPER PARAMETER]
Objective function: cross entropy
Optimizer: Adam
Activation Function: Leaky ReLU
Conv filter → Batch Normalization
No dropout
Learning rate = 0.0001

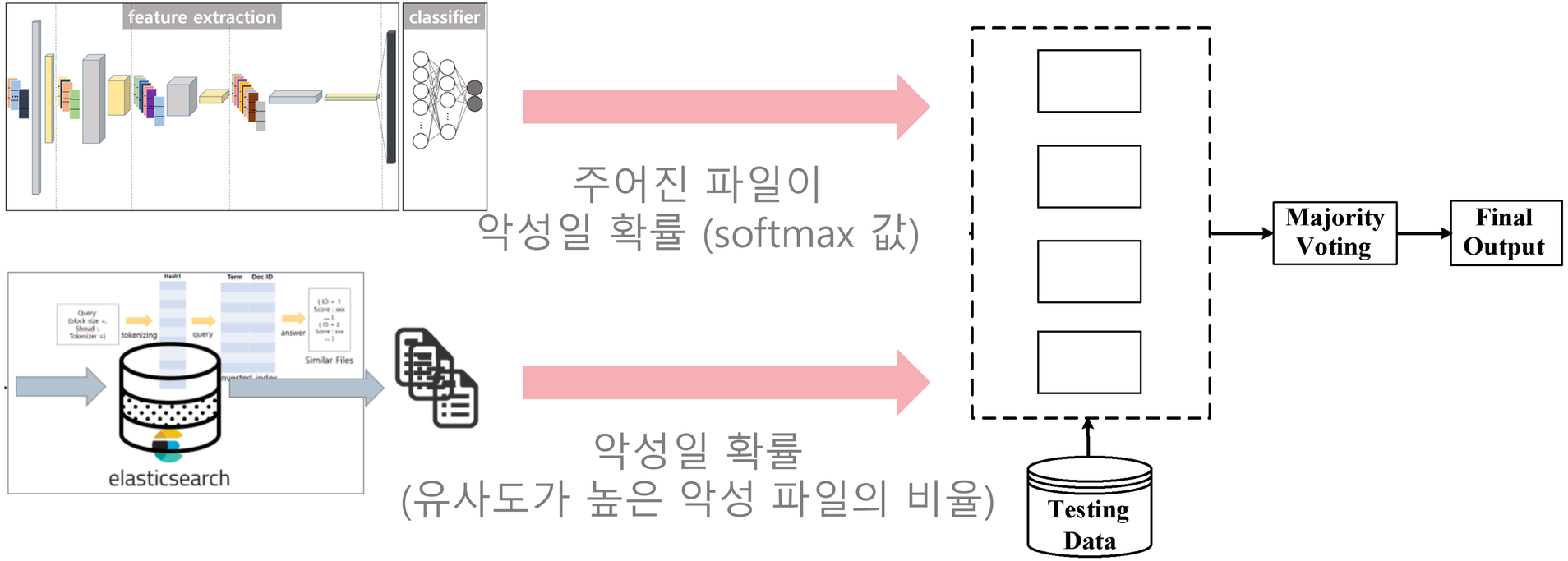
악성코드 탐지 시스템

- 엘라스틱서치 기반 최대 유사 악성코드 검색 모델
  - 전처리 (preprocessing) → 특징 추출
  - 특징 정보를 많이 가질 수록 유사 점수(score) 상승 → 점수에 따라 악성/정상 여부 판단



악성코드 탐지 시스템

- 머신 러닝 기반 앙상블 모델
  - 다양한 특징 정보에 대한 두 모델(딥러닝, 엘라스틱서치)의 결과를 상호보완함
  - 사용 앙상블 모델: Random Forest



특징 정보 (feature)

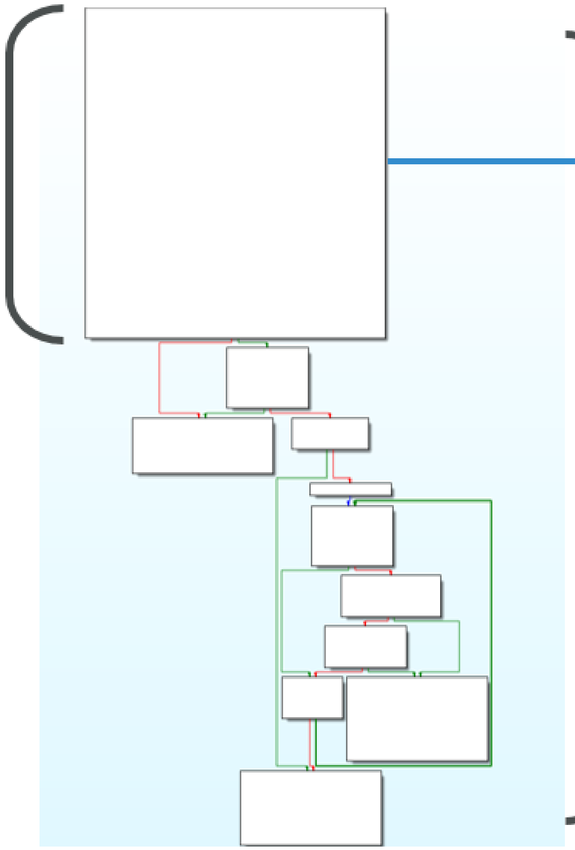
- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - disassemble information
  - strings
  - ssdeep
  - Import information

- 동적 정보 [Cuckoo Sandbox]
  - API call sequence

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - **structured opcode sequence**
  - disassemble information
  - strings
  - ssdeep
  - Import information

기본 블록
(basic block)



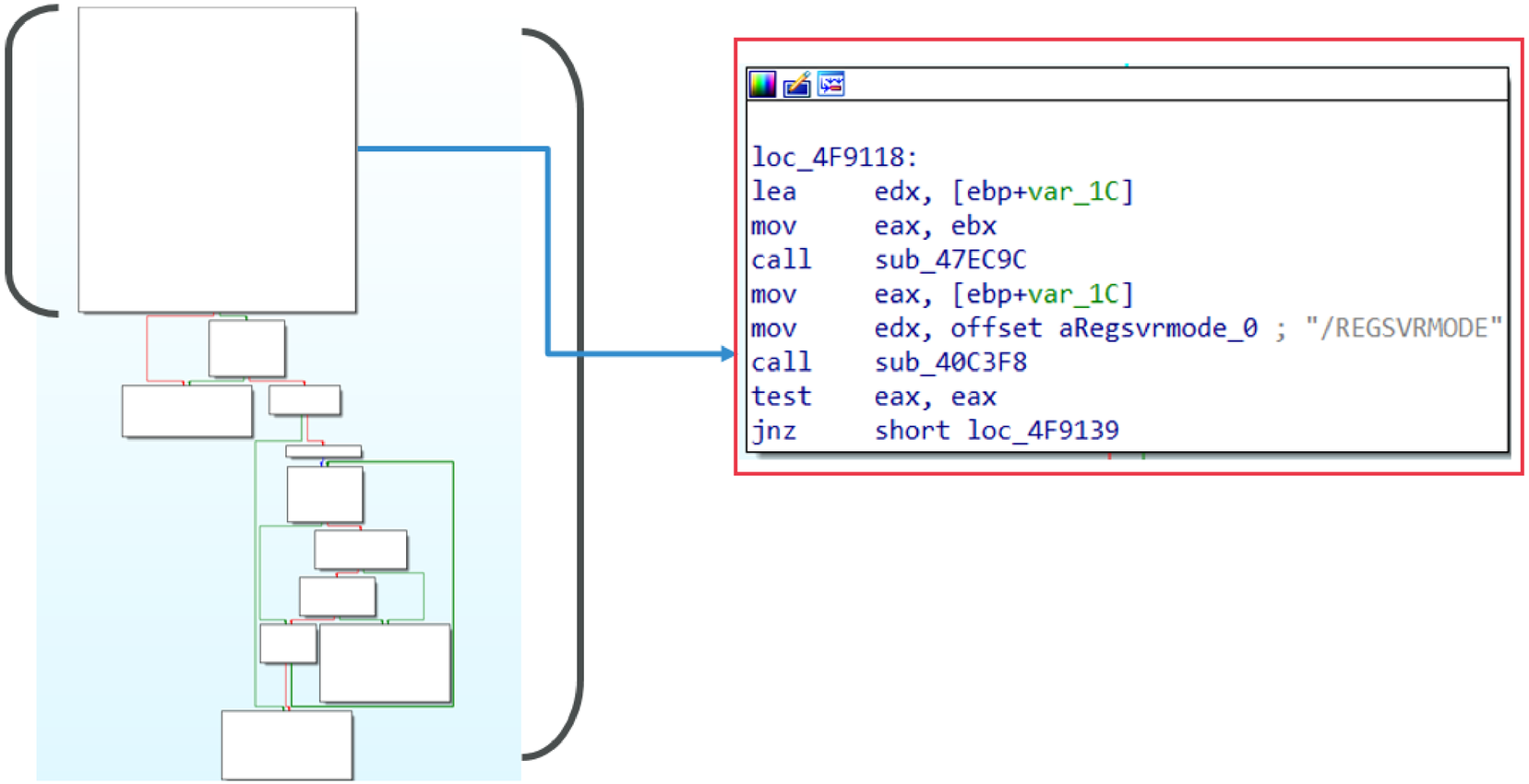
```
loc_4F9118:
lea     edx, [ebp+var_1C]
mov     eax, ebx
call    sub_47EC9C
mov     eax, [ebp+var_1C]
mov     edx, offset aRegsvrmode_0 ; "/REGSVRMODE"
call    sub_40C3F8
test    eax, eax
jnz     short loc_4F9139
```

함수
(function)

- 동적 정보 [Cuckoo Sandbox]
  - API call sequence

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - **disassemble information**
  - strings
  - ssdeep
  - Import information
- 동적 정보 [Cuckoo Sandbox]
  - API call sequence



특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - disassemble information
  - **strings**
  - ssdeep
  - Import information
- 동적 정보 [Cuckoo Sandbox]
  - API call sequence

Binary file 에 대해
정규식을 이용하여 추출

['₩x1f !"#$%&₩'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[₩₩]^_`ABCDEFGHIJKLMNOPQRSTUVWXYZ{|}~',
'₩x1f !"#$%&₩'()*+,-./0123456789:;<=>?@abcdefghijklmnopqrstuvwxyz[₩₩]^_`abcdefghijklmnopqrstuvwxyz{|}~',
'₩x1fgm]zs', '₩x1ftuVS;%', ' ', ' (((((H', " <requestedExecutionLevel
level='asInvoker' uiAccess='false' />", ' (', ' </requestedPrivileges>', '
<requestedPrivileges>', ' </security>', ' <security>', ' </trustInfo>', ' <trustInfo xmlns="urn:schemas-
microsoft-com:asm.v3">', ' 4P/qD', " Base Class Array'", ' Base Class Descriptor at (', " Class Hierarchy Descriptor'",
" Complete Object Locator'", " Type Descriptor'", ' delete', ' delete[]', ' new[]', '!25jK94', '!:O/.E', '!E>^~A', '"B <1=',
'"hLFGp', '"~):+nN', '#49sOD', '#9₩x1fRM6', '#IsMPQ', "$'cdAV", '$5PiU&n', '$f[2@D', '$h[#XC', '%;s%^,',
'%S#[k₩x1f=', "'! 2XVa", "'+T^=^$", "'/)f3.", "'<v(TW", "'HtkeQ", '(/+uxrT', '(`Y-b0', '(null)', ')@Ag^#|', ')lk₩₩p)',
')}|)}|', "*'3IU32LR", '*HgW&66v', "*VP[yC:'", '+0₩₩H{Oh', '+6@_m,', ",0{x8'vZ", "-'tfmH", '.00cfg', '.3MY^^',
'.?AVbad_alloc@std@@', '.?AVbad_array_new_length@std@@', '.?AVbad_exception@std@@',
'.?AVexception@std@@', '.?AVlength_error@std@@', '.?AVlogic_error@std@@', '.?AVout_of_range@std@@',
'.?AVtype_info@@', '.CRT$XCA', '.CRT$XCAA', '.CRT$XCZ', '.CRT$XIA', '.CRT$XIAA', '.CRT$XIAC', '.CRT$XIC',
'.CRT$XIZ', '.CRT$XLA', '.CRT$XLZ', '.CRT$XPA', '.CRT$XPX', '.CRT$XPXA', '.CRT$XPZ', '.CRT$XTA', '.CRT$XTZ', '.PF bi',
'.data$r', '.gfids', '.gfids$x', '.gfids$y', '.idata$2', '.idata$3', '.idata$4', '.idata$5', '.idata$6', '.rdata', '.rdata$T',
'.rdata$r', '.rdata$sxdata', '.rdata$zzzdbg', '.rsrc$01', '.rsrc$02', '.rtc$IAA', '.rtc$IZZ', '.rtc$TAA', '.rtc$TZZ', '.text$mn',
'.text$x', '.tls$ZZZ', '.xdata$x', '/#8_o8_o8_o', "/f1X1'|p5", '0₩x1f020j0r0', '0 0(00080@0H0P0X0`0h0p0x0', '0#0-0=0',
'0$0,040<0D0L0T0₩₩0d0l0t0|0', '0$1C1`1{1', "0'090K0]0o0", '0*0/0@0F0Q0Y0d0j0u0{0', '0,161<1B1', '0.LyJo',
'0=xaK/', '

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - disassemble information
  - strings
  - **ssdeep**
  - Import information
- 동적 정보 [Cuckoo Sandbox]
  - API call sequence

- ssdeep
  - Fuzzy 해시 기반의 파일 유사도 측정 프로그램
  - Fuzzy 해시: 일정 크기 단위로 구분, 각 블록의 해시 값을 생성하는 방식


```
{ -
  "_index": "kisa_only_our_data_ssdeep",
  "_type": "type_ssdeep",
  "_id": "b4263c6c73b32ff1a5b2b462f2c9192f",
  "_version": 1,
  "found": true,
  "_source": { -
    "chunk_size": "768",
    "chunk": "zXjvSeUvK9U1AhqCf5lyySKFsn6g8fDrrDOmjPJOMw80xGSTwGKn23+zjGk",
    "double_chunk": "zuHQJf3jFC6gBYJOMwLxFTX2Gk"
  }
}
```

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - disassemble information
  - strings
  - ssdeep
  - **Import information: IAT 에서 라이브러리 정보 추출**
- 동적 정보 [Cuckoo Sandbox]
  - API call sequence


```
import
  KERNEL32
  ADVAPI32
    0 : "RegOpenKeyA"
    1 : "GetUserNameA"
    2 : "FreeSid"
    3 : "AllocateAndInitializeSid"
    4 : "RegDeleteKeyA"
    5 : "EqualSid"
    6 : "RegSetValueExA"
    7 : "GetTokenInformation"
    8 : "OpenProcessToken"
    9 : "RegCloseKey"
  SHELL32
    0 : "SHChangeNotify"
    1 : "ShellExecuteA"
    2 : "IsUserAnAdmin"
  WININET
    0 : "HttpOpenRequestA"
    1 : "InternetOpenUrlA"
    2 : "HttpSendRequestA"
    3 : "InternetOpenA"
    4 : "InternetCloseHandle"
    5 : "InternetConnectA"
    6 : "InternetReadFile"
```

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - IDAPython 기반 정적 정보 추출 자동화

```
JSON
  string
  export
  opcode
    FILE_SIZE : 602768
    CRC32 : 3265744573
  import
    SHA256 : "670b9e2719f3767f5af3adb0493264a1e612ac29e8fd61b9c1354f2e3ecce99b"
  asm
    MD5 : "00bb510faed27f32f6877adf81b0ee66"
```

특징 정보 (feature)

- 정적 정보 [IDA Pro]
  - structured opcode sequence
  - disassemble information
  - strings
  - ssdeep
  - Import information
- 동적 정보 [Cuckoo Sandbox]
  - **API call sequence**

Cuckoo Sandbox 를 통해
프로세스 별 API 호출 시퀀스 추출

```
▼ 3440 {81}
    RegCreateKeyExW : 7
    NtDuplicateObject : 7
    NtOpenSection : 4
    NtFreeVirtualMemory : 2
    RegCloseKey : 90
    NtQueryKey : 2
    NtReadFile : 4
    LdrUnloadDll : 12
    HttpOpenRequestA : 11
    GetSystemInfo : 1
    RegQueryValueExA : 64
    getaddrinfo : 2
    InternetOpenA : 11
    OpenServiceW : 1
    CopyFileA : 1
    NtTerminateProcess : 3
    NtClose : 118
    RegCreateKeyExA : 5
    GetFileSize : 5
    InternetCloseHandle : 33
    SetFileTime : 3
    NtDelayExecution : 13
    InternetConnectA : 11
    InternetQueryOptionA : 1
    NtDeviceIoControlFile : 24
    NtAllocateVirtualMemory : 22
    ReadProcessMemory : 5
    RegOpenKeyExA : 23
    DeleteFileW : 1
    NtWriteFile : 2
    LdrGetDllHandle : 11
    OpenServiceA : 2
    NtWriteVirtualMemory : 1
    SetFilePointer : 4
    NtQueryValueKey : 1
    NtResumeThread : 2
    CryptAcquireContextA : 1
```

데이터셋

- 목적
  - 빅데이터 기반의 신뢰도 높은 모델 생성
  - 다양한 백신사의 라벨 활용

- 데이터셋
  - 악성: 자체 수집 약 500,000개
  - 정상: 자체 수집 약 100,000개

  * 금년 KISA 학습/예선 데이터 적극 활용

결과

- 본선 대회 대비 정확도
  - KISA 예선 대회 데이터를 통한 예상 정확도 : 97~99%

- 본선 실제 대회 정확도

AI기반 악성코드 탐지_대학(원)부문

순위	팀명	탐지율									
		1차 1회	1차 2회	1차 3회	1차 4회	1차 5회	2차 1회	2차 2회	2차 3회	2차 4회	2차 5회
1위	KMU InfoSec	14:21:40 94.14%	17:31:16 94.64%	17:43:43 95.91%	18:25:36 94.72%	18:26:29 95.91%	12:55:10 93.87%	14:09:12 95.43%	14:47:07 95.63%	15:22:53 96.28%	

Thank you

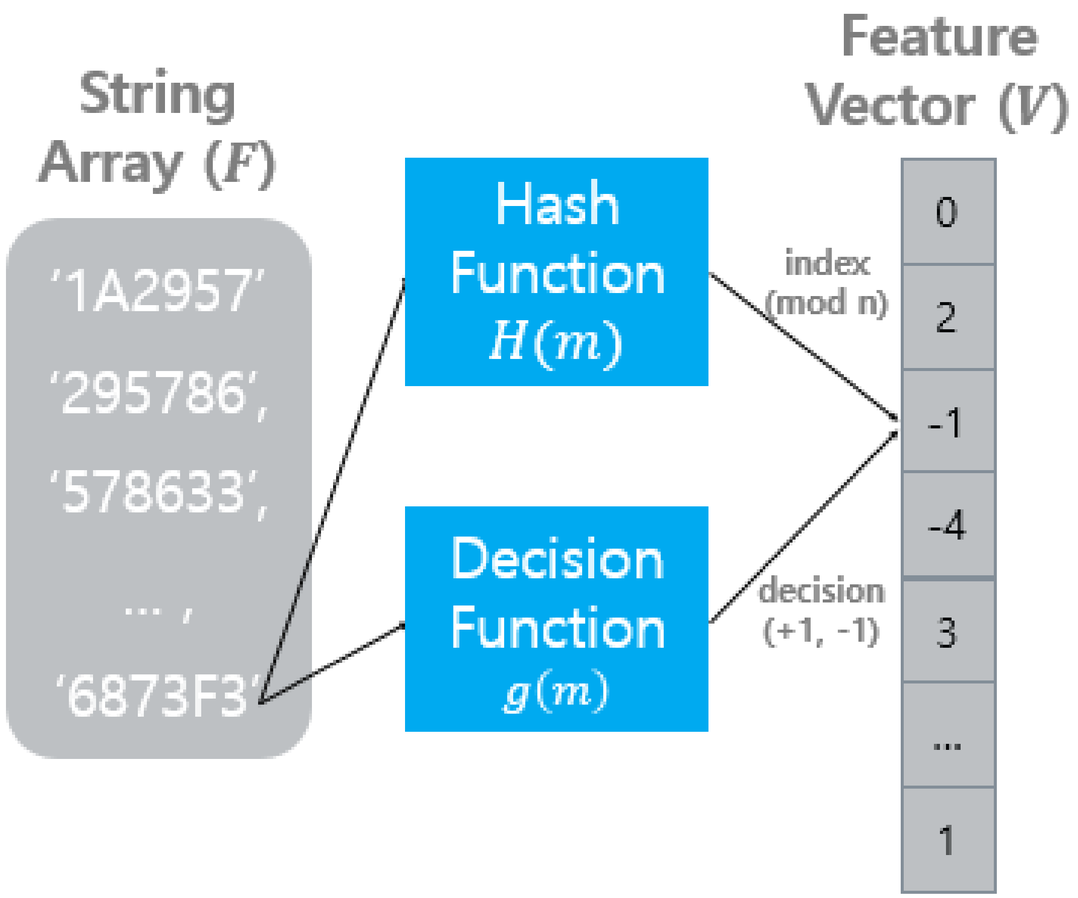
&

Q n A

API call sequence (for Deep Learning)

- N-gram chunking (N=4)
  - Feature Hashing
    - **Frequency**-based feature vector

```
def hashing_vectorizer(features: array of string, N: integer):
    x := new vector[N]
    for f in features:
        idx := hash(f) mod N
        if g(f) == 1:
            x[idx] += 1
        else:
            x[idx] -= 1
    return x
```



APPENDIX

structured opcode sequence (for Deep Learning)

- N-gram ? V-gram !

함수 A

```
① loc_8F8E3E:
60 pusha
66+bt        sp, 0Dh
E9+jmp       loc_A6B9BC
73+; END OF FUNCTION
```

```
② loc_A6B9BC:
01+add       eax, edi
60 pusha
F5 cmc
E9+jmp       loc_A6CA33
6E+; END OF FUNCTION
```

```
③ loc_A6CA33:
53 push      ebx
39+cmp       eax, edx
9C pushf
9C pushf
E9+jmp       loc_A4E049
0C+; END OF FUNCTION
```

"함수 A" OPCODE SEQUENCE (3-D List)

OPCODES = [...,
① [['66', '66', 'E9', '73'], *b.b*
② ['01', '60', 'F5', 'E9', '6E'], *b.b*
③ ['53', '39', '9C', '9C', 'E9', '0C']], *b.b*
...]

함수 A

"함수 A"에 대한 특징 집합 추출

- trigram (size : 13)

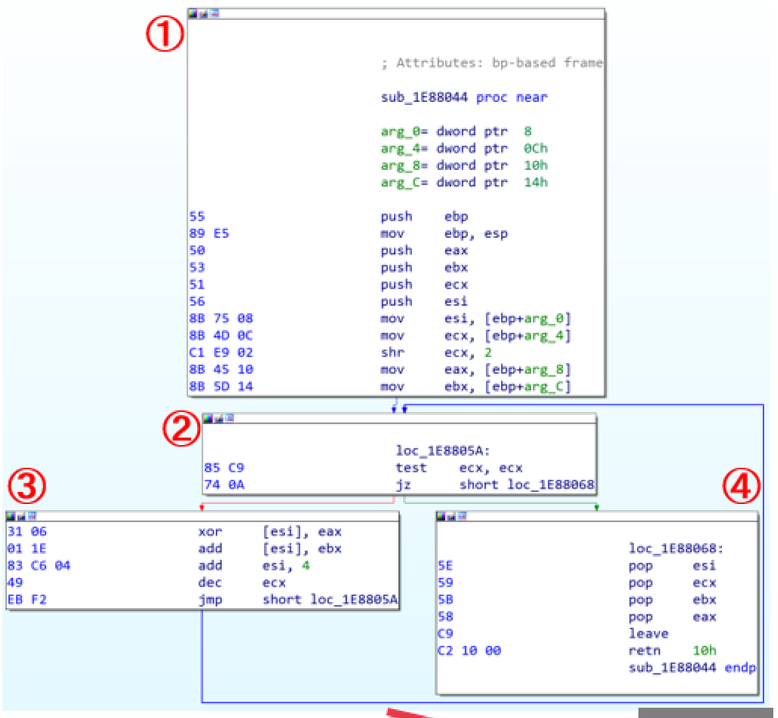
= {'6666E9', '66E973', 'E97301', '730160', '0160F5', '60F5E9', 'F5E96E', 'E96E53', '6E5339', '53399C', '399C9C', '9C9CE9', '9CE90C'}

- v-gram (size : 3)

= {'6666E973', '0160F5E96E', '53399C9CE90C'}

structured opcode sequence (for Deep Learning)

- N-gram ? V-gram !



- Basic block

md5 hash

① ['55', '89', '50', '53', '51', '56', '8B', '8B', 'C1', '8B', '8B'] → C6A9FBE75628012CC69FD1FC415B20F1

② ['85', '74'] → E345FAC6BC5C868F0222430C733FA26E

③ ['31', '01', '83', '49', 'EB'] → AC2D461042F560151E439BE4F0F2A6B4

④ ['5E', '59', '5B', '58', 'C9', 'C2'] → C6A9FBE75628012CC69FD1FC415B20F1

- Function: XOR operation result of basic block representative value

→ C6A9FBE75628012CC69FD1FC415B20F1 XOR E345FAC6BC5C868F0222430C733FA26E XOR AC2D461042F560151E439BE4F0F2A6B4 XOR C6A9FBE75628012CC69FD1FC415B20F1

= **4F68BCD6FEA9E69A1C61D8E883CD04DA**

3-D LIST

b.b b.b b.b b.b

FILE = [..., [['55', '89', ..., '8B', '8B'], ['85', '74'], ['31', '01', ..., 'EB'], ['5E', '59', ..., 'C2']], ...]

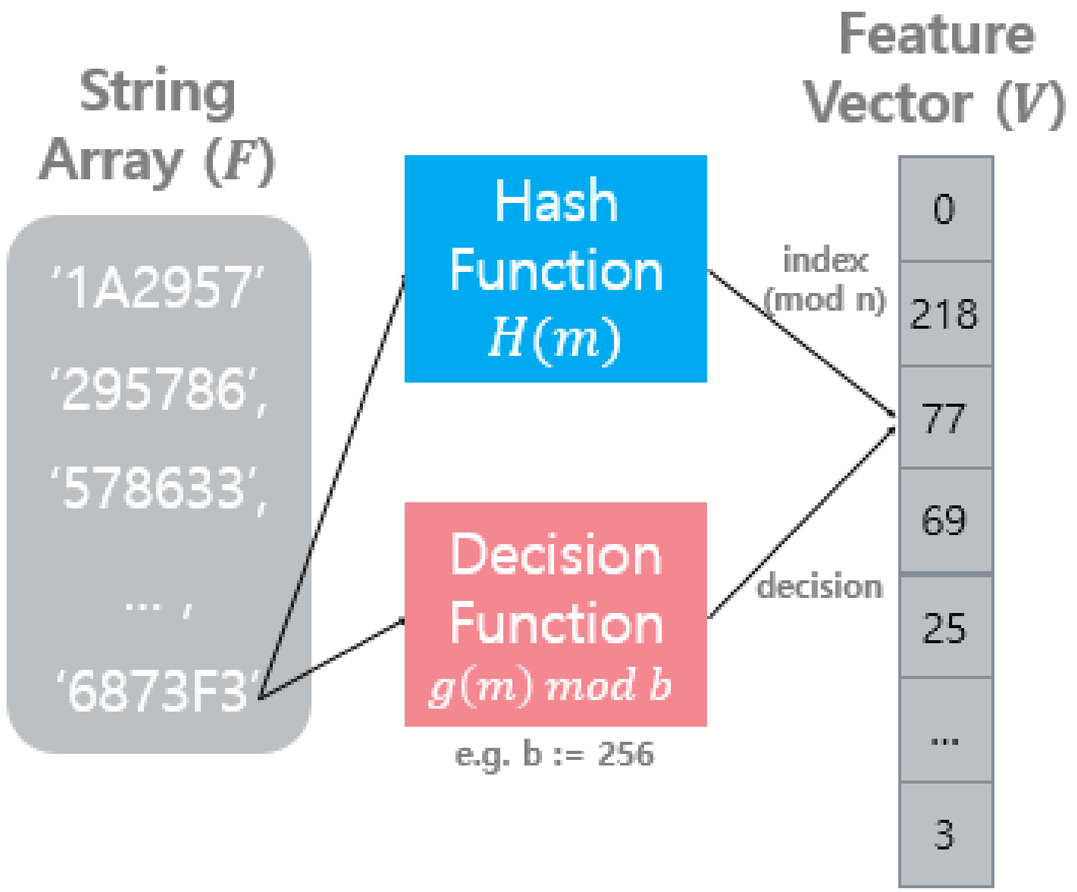
function

structured opcode sequence (for Deep Learning)

- V-gram chunking
  - Feature Hashing
    - **Content**-based feature vector
      - overwrite max value

```
def hashing_vectorizer(features: array of string, N: integer,
                                                    B: integer):
    x := new vector[N]
    for f in features:
        idx := hash(f) mod N
        value := G(f) mod B
        x[idx] = value > x[idx] ? value : x[idx]

    return x
```



엘라스틱서치를 위한 특징 정보 추출

- disassemble, ssdeep, strings, import
  - 엘라스틱서치의 value 로 가공하지 않고 사용