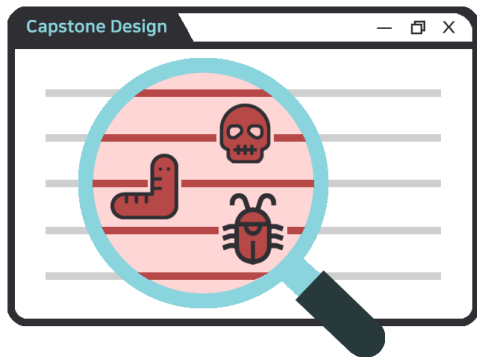


MASK

Malware
Analysis
System
in Kookmin



16조 - 한채연 김영재 명준우 이유정 허준녕

NCNP

“기하급수적으로 증가하는 악성코드”

“

현격히 부족한
악성코드 전문가의 수

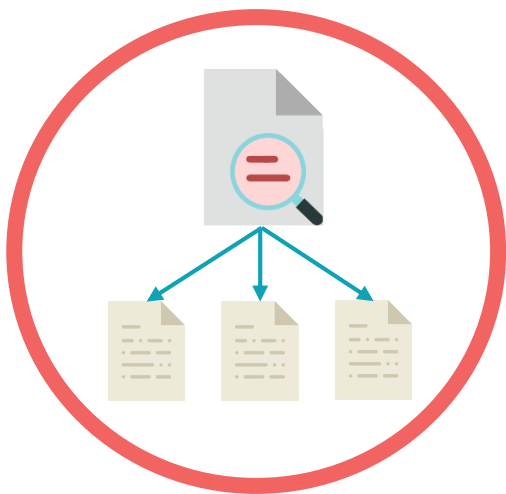
”



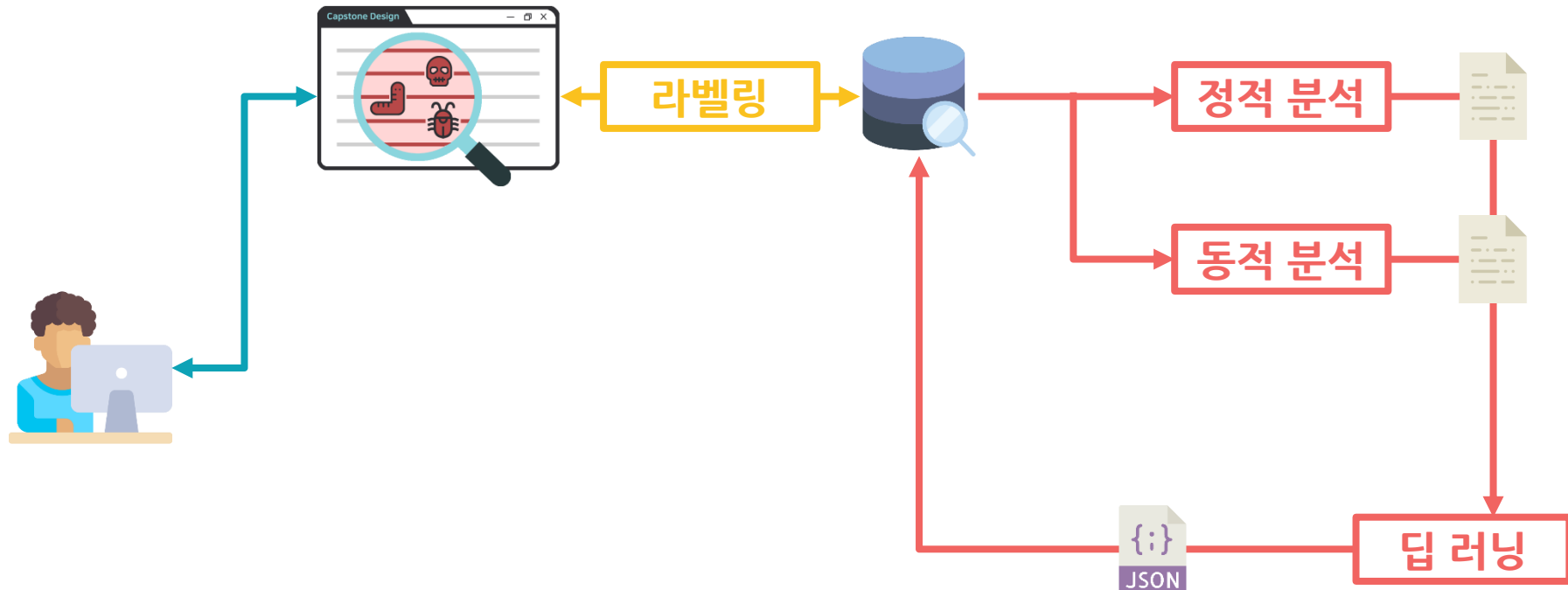
Last update: 03-01-2018 11:58

Copyright © AV-TEST GmbH, www.av-test.org

01. 프로젝트 목표



01. 프로젝트 목표



02. 수행 내용

정적 분석



동적 분석



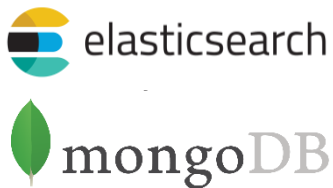
라벨링



딥 러닝

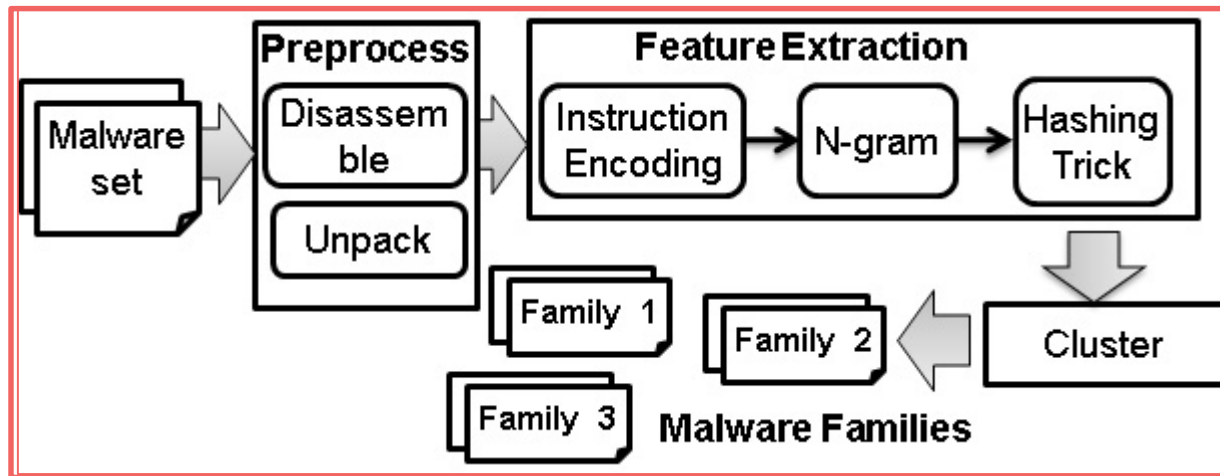


데이터 처리



웹





A system overview of MutantX-S (출처 : MutantX-S:
Scalable Malware Clustering Based on Static Features)



Disassemble
by IDA Pro



Extract
Opcode Sequence



N-gram



Feature
Hashing

```
4004ed: 55      push    rbp
4004ee: 48 89 e5  mov     rbp, rsp
4004f1: c7 45 ec 00 00 00 00  mov     DWORD PTR [rbp-0x14], 0x0
4004f8: c7 45 f0 01 00 00 00  mov     DWORD PTR [rbp-0x10], 0x1
4004ff: c7 45 f4 02 00 00 00  mov     DWORD PTR [rbp-0xc], 0x2
400506: c7 45 f8 03 00 00 00  mov     DWORD PTR [rbp-0x8], 0x3
40050d: c7 45 fc 04 00 00 00  mov     DWORD PTR [rbp-0x4], 0x4
400514: c7 45 ec 00 00 00 00  mov     DWORD PTR [rbp-0x14], 0x0
40051b: eb 13     jmp     400530 <main+0x43>
40051d: 8b 05 15 0b 20 00     mov     eax, DWORD PTR [rip+0x200b15] # 601038 <globalA>
400523: 83 e8 01     sub     eax, 0x1
400526: 89 05 0c 0b 20 00     mov     DWORD PTR [rip+0x200b0c], eax # 601038 <globalA>
40052c: 83 45 ec 01     add     DWORD PTR [rbp-0x14], 0x1 # 601038 <globalA>
400530: 8b 05 02 0b 20 00     mov     eax, DWORD PTR [rip+0x200b02] # 601038 <globalA>
400536: 39 45 ec     cmp     DWORD PTR [rbp-0x14], eax
400539: 7c e2     jl      40051d <main+0x30>
40053b: 5d     pop     rbp
40053c: c3     ret
40053d: 0f 1f 00     nop     DWORD PTR [rax]
```

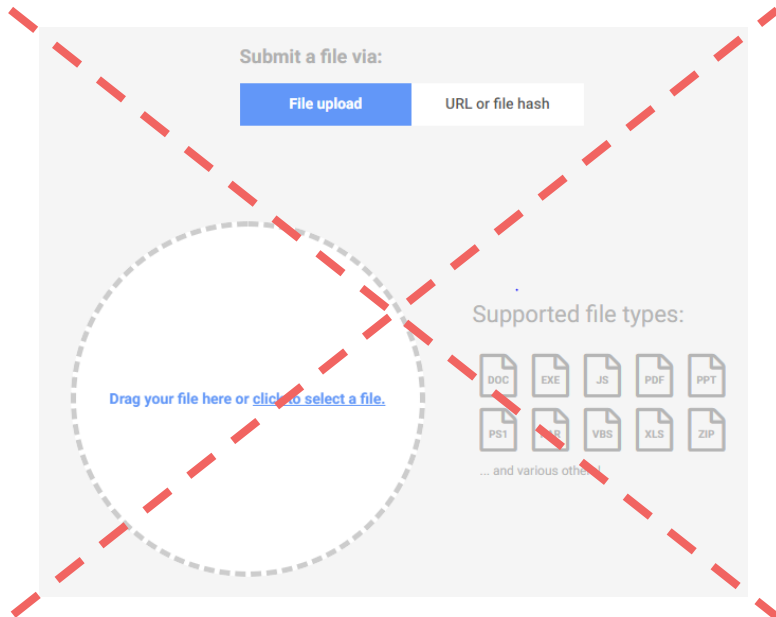
opcode sequence

비정상적인 접근을 탐지하기 위해 의도적으로 설치해 둔 시스템

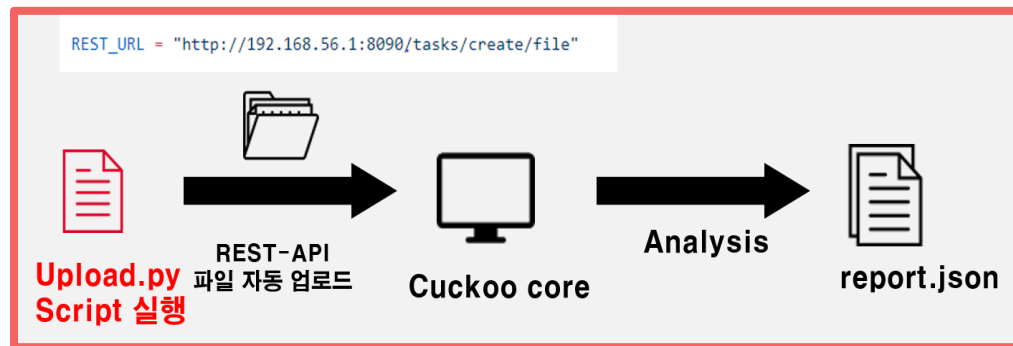


02. 수행 내용

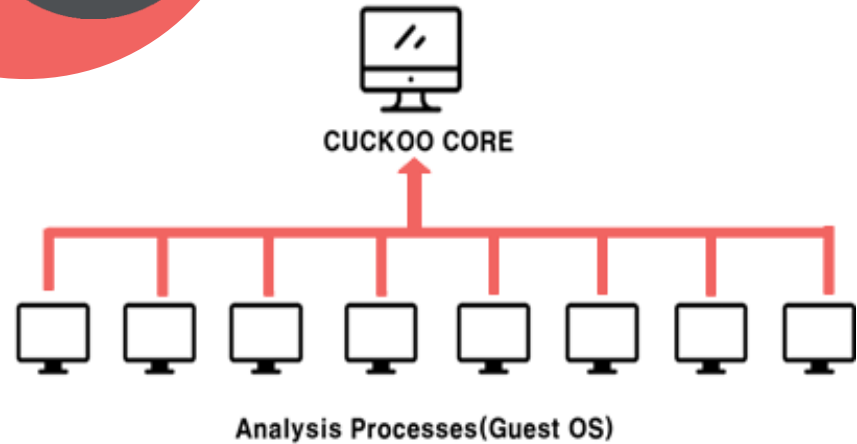
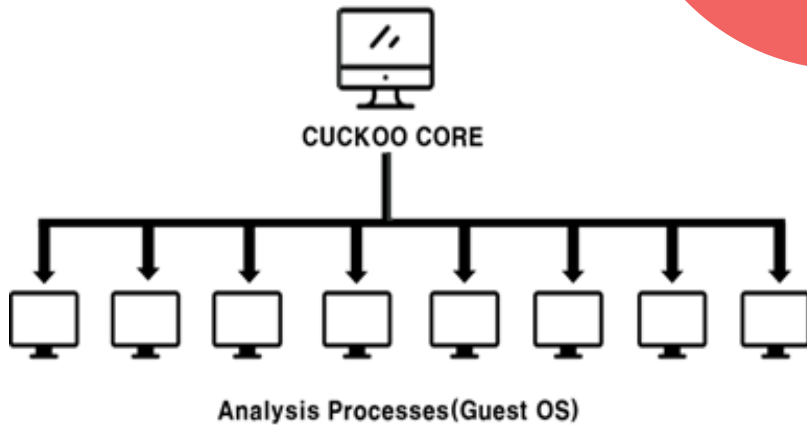
동적 분석



웹 인터페이스를 이용하여 파일을 업로드하는 방법



웹 인터페이스를 이용하여 파일을 업로드하는 방법

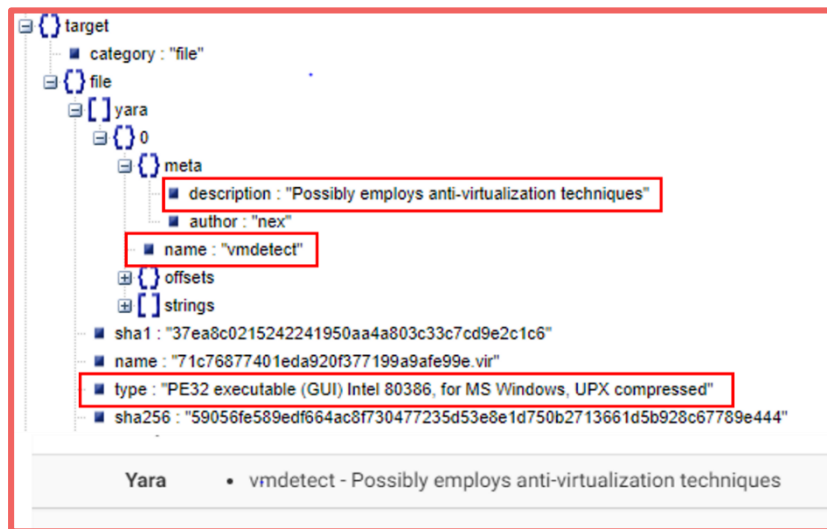


01 Process memory

프로세스에 대한 메모리 덤프 분석

02 Target

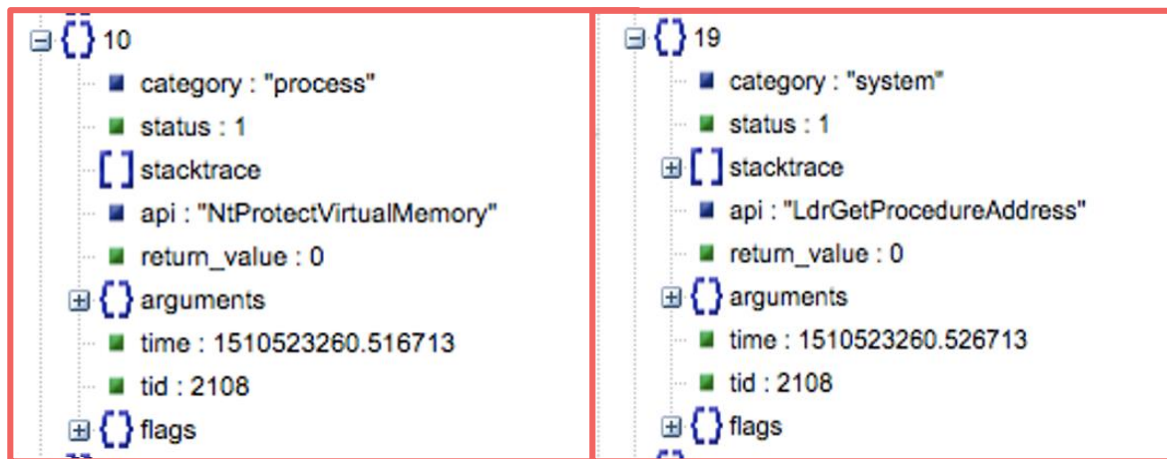
yara rule에 의해 탐지되었을 경우 나타나는 정보이다.



03 네트워크 프로토콜, 악성코드를 실행한 host 정보

04 정적 분석 결과(Strings..)

05 Behavior(API 통계, API call sequence..)



The screenshot displays two side-by-side panels of API call logs from Cuckoo Sandbox. The left panel, labeled '10', shows the API call sequence for a process with category 'process'. The right panel, labeled '19', shows the API call sequence for a process with category 'system'. Both panels list the same set of fields: category, status, stacktrace, api, return_value, arguments, time, tid, and flags. The 'api' field in the left panel is 'NtProtectVirtualMemory', while in the right panel it is 'LdrGetProcedureAddress'. The 'time' and 'tid' values are identical in both panels.

Field	Process 10 (category: "process")	Process 19 (category: "system")
category	"process"	"system"
status	1	1
stacktrace	[stacktrace]	[stacktrace]
api	"NtProtectVirtualMemory"	"LdrGetProcedureAddress"
return_value	0	0
arguments	[arguments]	[arguments]
time	1510523260.516713	1510523260.526713
tid	2108	2108
flags	[flags]	[flags]

한 프로세스의 10번째, 19번째 API호출 기록(리포트)

05

Behavior(API 통계, API call sequence..)

쿠쿠샌드박스는 아래와 같이 323개 함수의 API 호출을 기록하고, 자체적으로 17개의 카테고리 분류한다.

class	description	example	# of APIs
A	file/directory	CopyFile, CreateDirectory, GetFileType, ...	47
B	registry	RegCreateKeyEx, NtCreateKey, RegDeleteValue, ...	38
C	internet explorer	CDocument_write, CScriptElement_put_src, ...	7
D	user interface	DrawText, FindWindow, LoadString, ...	11
E	net API	NetGetJoinInformation, NetShareEnum, ...	6
F	network	DnsQuery_A, GetAdaptersInfo, HttpOpenRequestA, ...	62
G	OLE	CoCreateInstance, CoInitialize, ...	3
H	process	CreateProcess, CreateThread, Module32First, ...	41
I	synchronization	GetLocalTime, GetSystemTime, ...	8
J	resource	FindResource, LoadResource, ...	6
K	services	ControlService, CreateService, ...	12
L	system	GetNativeSystemInfo, LdrLoadDll, NtClose, ...	26
M	certificate	CertControlStore, CertOpenStore, ...	5
N	encryption	CryptCreateHash, CryptGenKey, ...	19
O	exception	SetUnhandledExceptionFilter, RtlDispatchException, ...	6
P	misc	GetUserName, GetDiskFreeSpace, WriteConsole, ...	20
Q	notification	_anomaly_, _exception_, ...	4

API Table (출처: 고동우, 김휘강(2017) "API콜 시퀀스와 Locality Sensitive Hashing을 이용한 악성코드 클러스터링 기법에 관한 연구", 정보보호학회논문지)

06

Signatures

위의 악성코드 정보들을 바탕으로 나타난 악성코드 특징(description)


Signatures
① Queries for the computername (1 event)
① Checks amount of memory in system, this can be used to detect virtual machines that have a low amount of memory available (1 event)
① A process attempted to delay the analysis task. (1 event)
① Drops a binary and executes it (1 event)
① Checks adapter addresses which can be used to detect virtual network interfaces (1 event)
① Potentially malicious URLs were found in the process memory dump (50 out of 124 events)
② Attempts to identify installed AV products by installation directory (3 events)
② Deletes its original binary from disk (1 event)
② A process performed obfuscation on information about the computer or sent it to a remote location indicative of CnC Traffic/Preperations. (4 events)

웹 인터페이스에서 확인 가능한 파일에 대한 signatures


07

Score


signatures로 식별한 패턴을 통해 의심스러운 평균 수준을 수치화한 정도

 Score

This file shows some signs of potential malicious behavior.
The score of this file is **1.2 out of 10**.

 Score

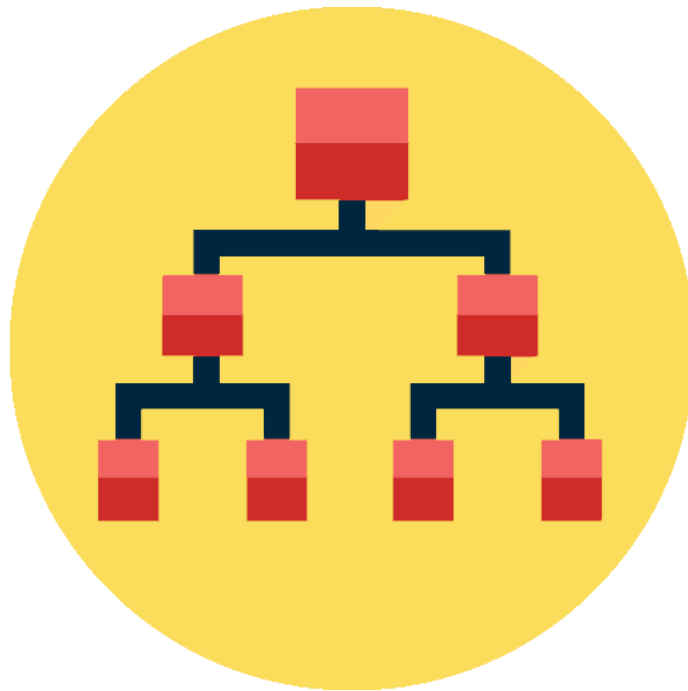
This file shows numerous signs of malicious behavior.
The score of this file is **4.2 out of 10**.

 Score

This file is **very suspicious**, with a score of **5.4 out of 10!**

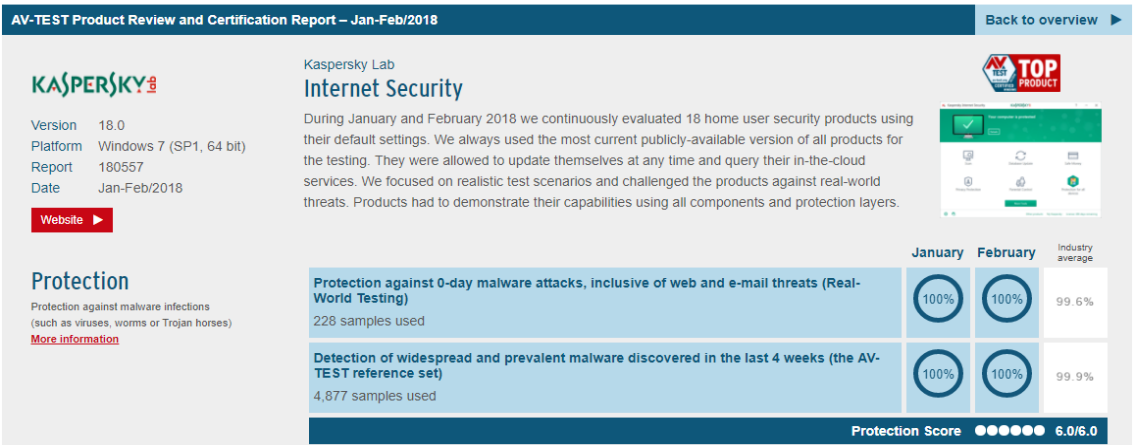
웹 인터페이스에서 확인 가능한 파일에 대한 score

악성코드를 어떻게 분류할 것인가





The best antivirus software for Windows Home User



VB100 results from 2018-02 (latest) on Windows 7 Professional, Windows 10 Professional

Read the full review, or download it.

SELECT INDICATORS		EXPORT TO CSV				
Tested product	Result	RAP Overview	WildList (%)	WildList (%)	False positives	False positives
Kaspersky Lab K Kaspersky Endpoint Security 10 for Windows	Passed virus 100		100.00	100.00	0	0

“악성코드는 일반적으로 7가지로 분류할 수 있다.”

Virus

Worm

Trojan


Downloader

Rootkit

Ransomware

Backdoor

01 카스퍼스키 라벨 가져오기



48 engines detected this file

SHA-256 c155eef293c5bc6606ccdcf4e671a7ef136fed1c764dcc0f8e53a40104c6096

File name 00cafeb38cd971f14b5159e813ee005f.virus

File size 327.86 KB

Last analysis 2016-09-08 21:32:01 UTC

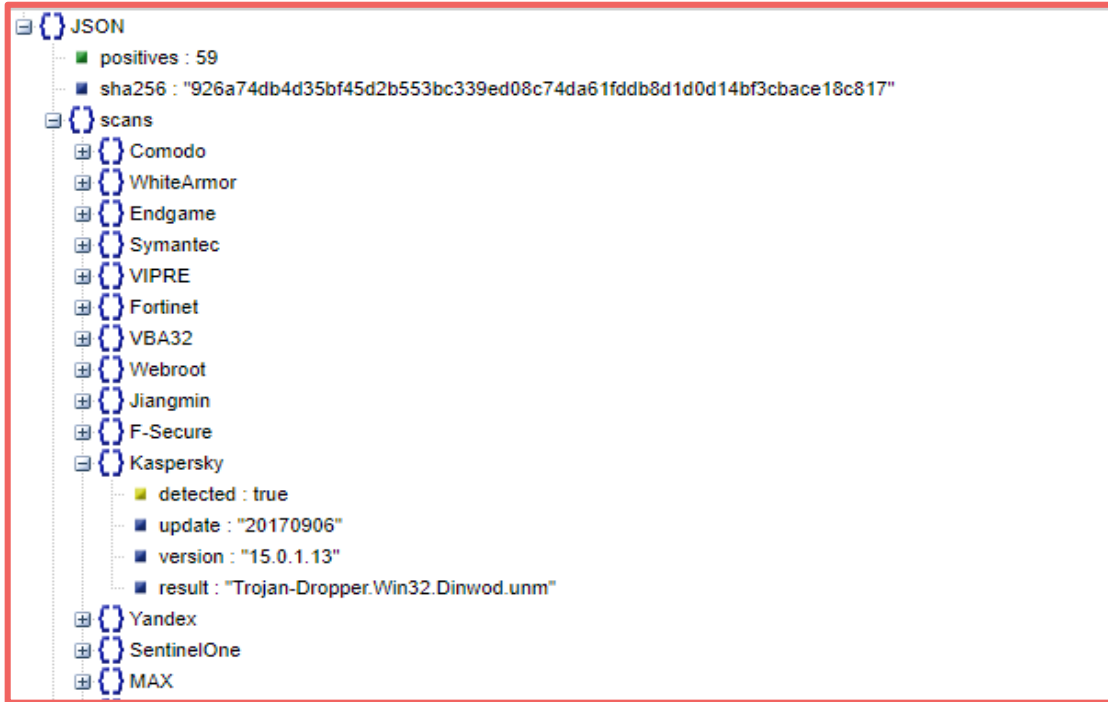
48 / 57

Detection Details Relations Behavior Community

Ad-Aware	! Trojan.Generic.12238246	AegisLab	! Troj.W32.AntiFW!Wsb
AhnLab-V3	! PUP/Win32.TSULoader.R104284	ALYac	! Trojan.Generic.12238246
Antiy-AVL	! Trojan/Win32.AntiFW.b	Arcabit	! Trojan.Generic.DBABDA6
Avast	! Win32:InstallRex-BH [PUP]	AVG	! InstallRex.256
Avira	! TR/Rogue.11512086.1	AVware	! Trojan.Win32.Generic!BT
Baidu	! Win32.Trojan-Downloader.Agent.bj	BitDefender	! Trojan.Generic.12238246
Bkav	! W32.HfsAdware.4191	CAT-QuickHeal	! PUA.Sergeypetr.Gen
ClamAV	! Win.Trojan.Antifw-171	Comodo	! Application.Win32.InstalleRex.DP

바이러스토탈에서 약 60여개의 안티바이러스의 분석 결과

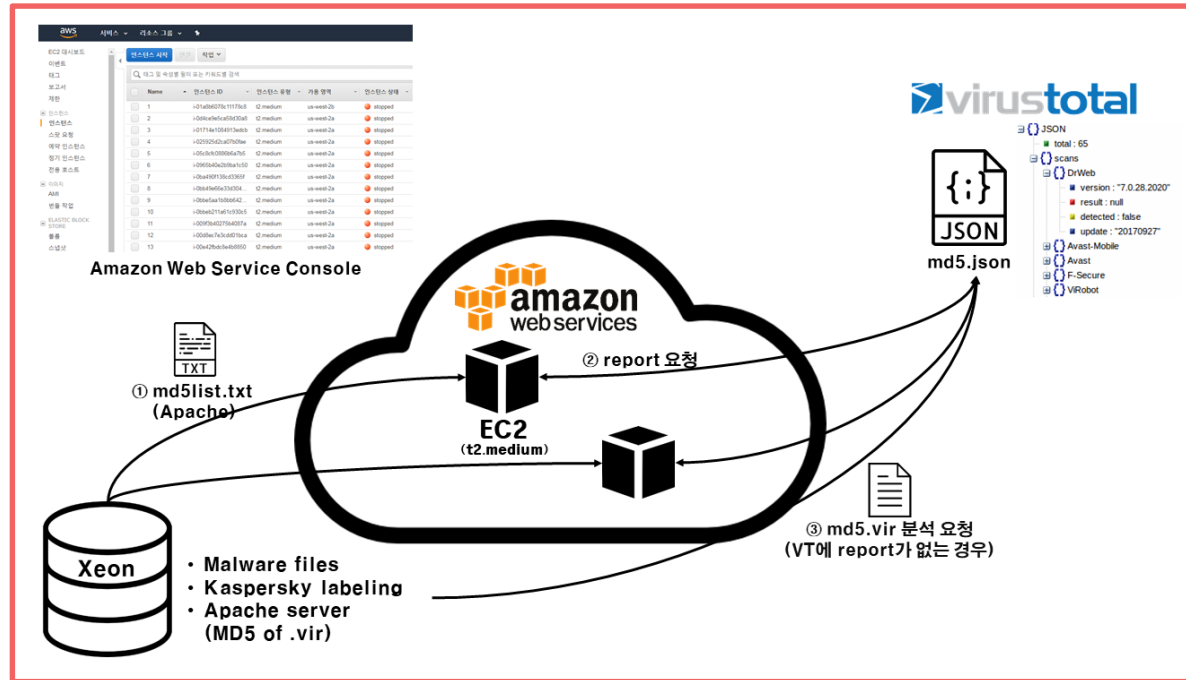
01 카스퍼스키 라벨 가져오기



바이러스토탈에서 제공되는 API를 통하여 받을 수 있는 json 형식의 분석 결과

02

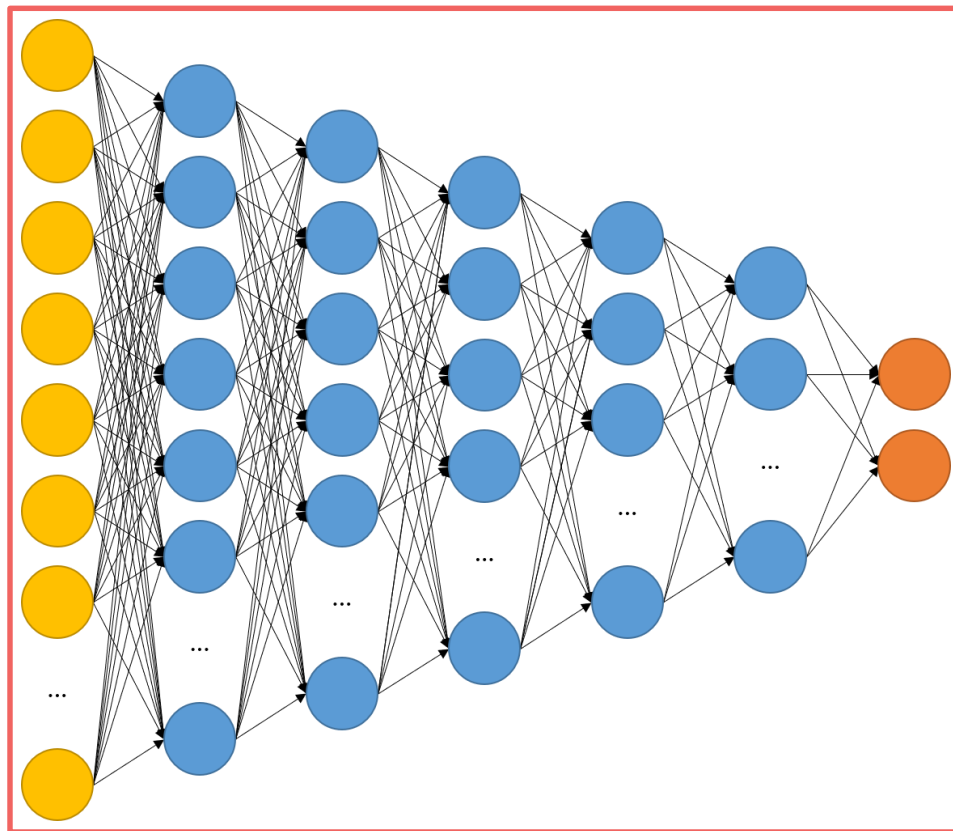
대량의 악성코드의 바이러스토탈 리포트를 가져오기



AWS EC2 인스턴스를 활용한 분산 분석



정적 분석



심층신경망 모델 구조

대량의 악성코드 샘플 데이터를 어떻게 운용하여
데이터 사용에 대한 시간 비용을 최소화 할 것 인가?

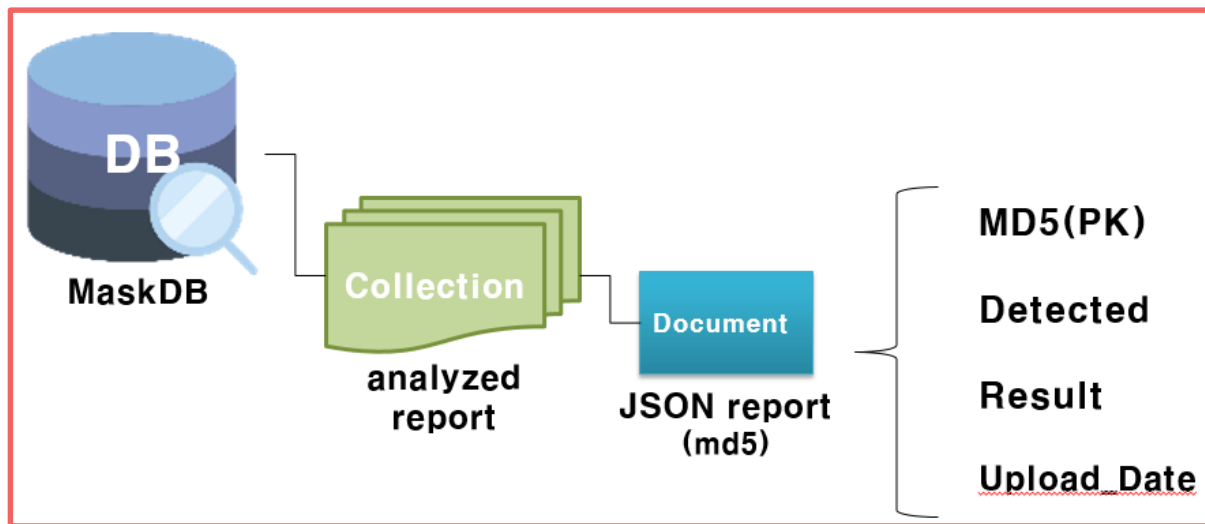


01 DB의 유연성

실험적인 Data 사용 및 저장을 위한 DB

02 Read 연산 위주의 DB

서비스 중 DB 접근은 주로 Data 검색을 위해 사용



NOSQL DBMS인 MongoDB를 구축

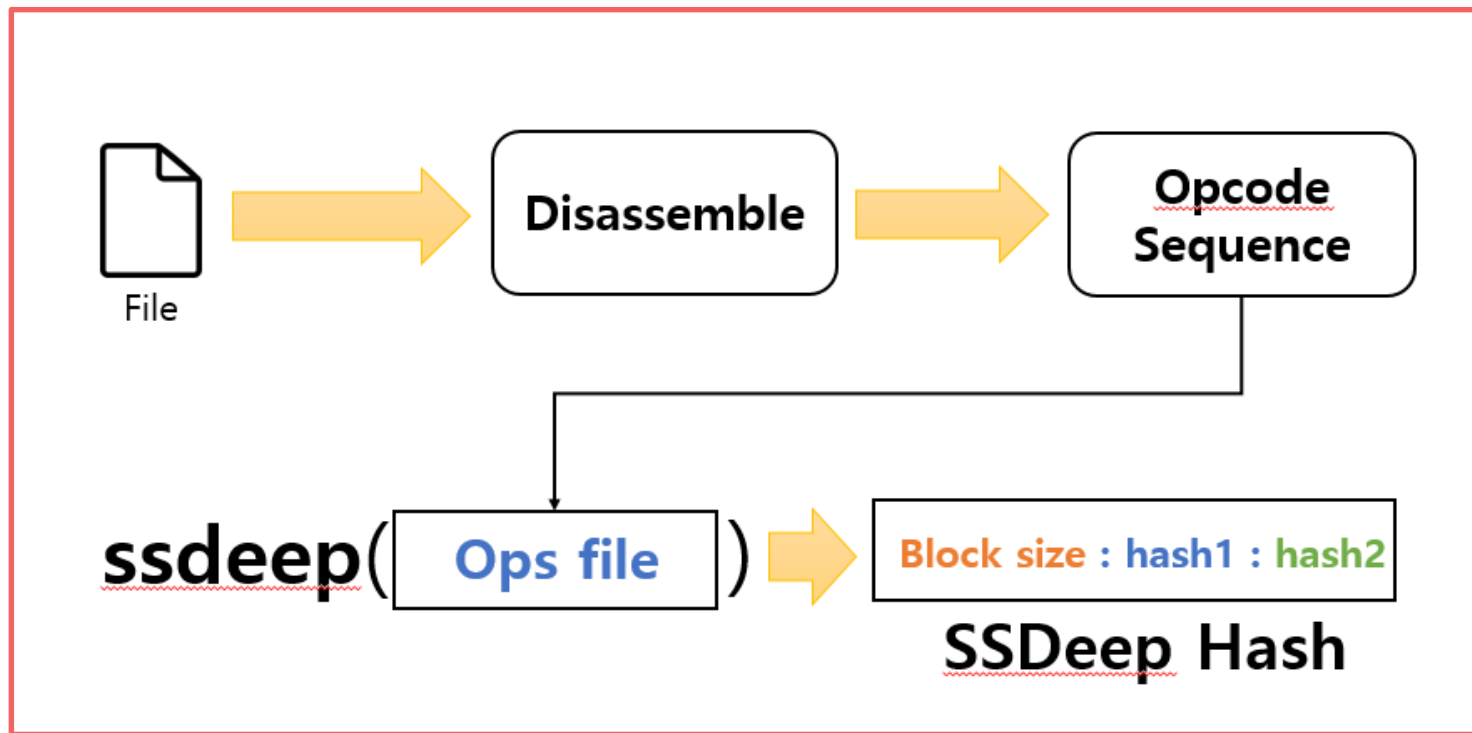
02. 수행 내용

데이터 처리

업로드 한 파일과 기존 DB에 있는 파일에서 유사한 파일을 어떻게 찾아서 보여줄 것인가?

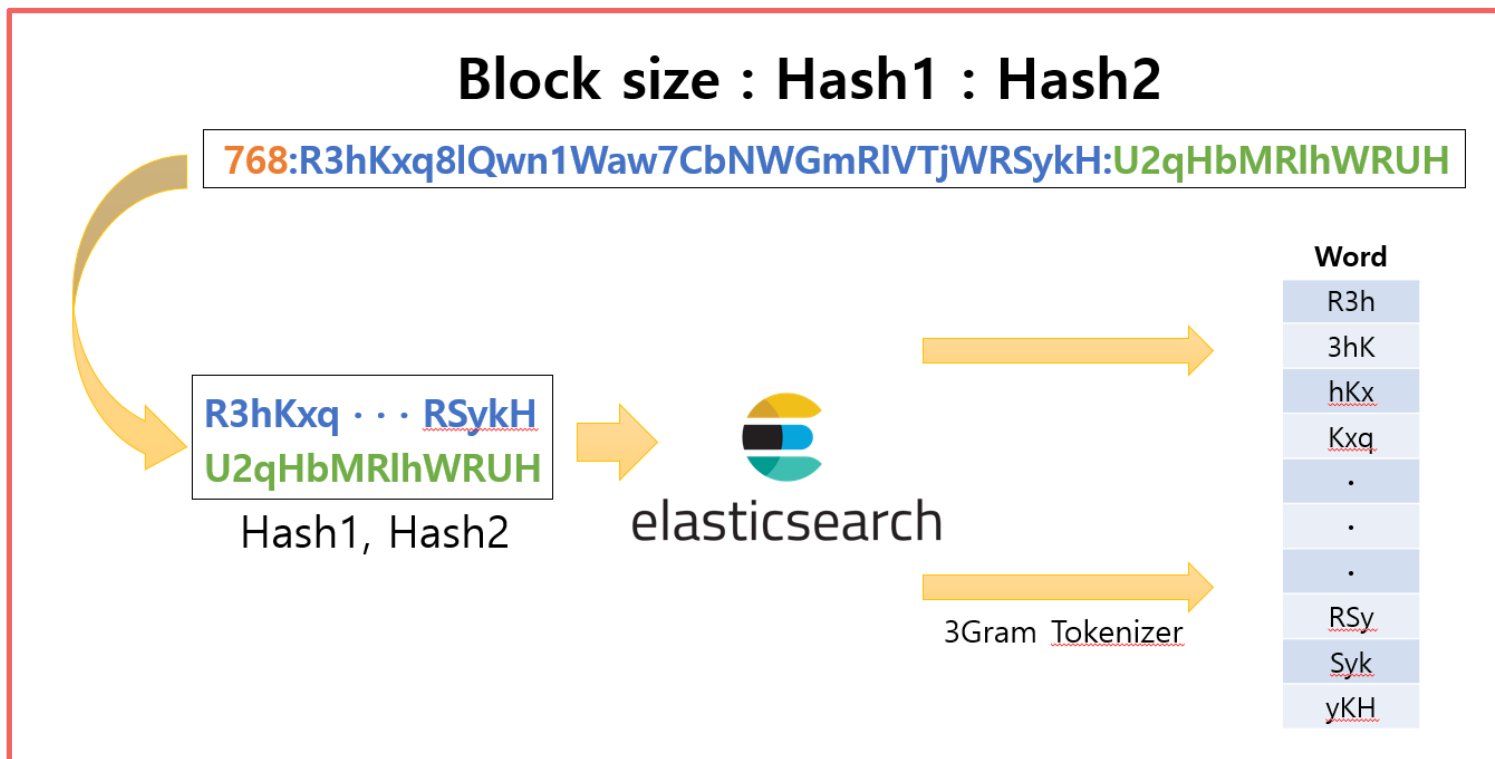


03 어떤 방법으로 유사도 측정을 할 것인가?



SSDeep 유사도 측정 툴을 이용한 Hash값 도출

04 도출된 Hash값 간의 유사도 비교를 위한 전 처리



스트링 매칭을 하기 위해 N-Gram Tokenize 를 이용하여 전 처리

05 빠른 검색 위한 방법

Doc ID = 1

Word

R3h
3hK
<u>hKx</u>
<u>Kxq</u>
.
.
.
<u>RSy</u>
<u>Syk</u>
<u>yKH</u>



elasticsearch



Mapping

Term

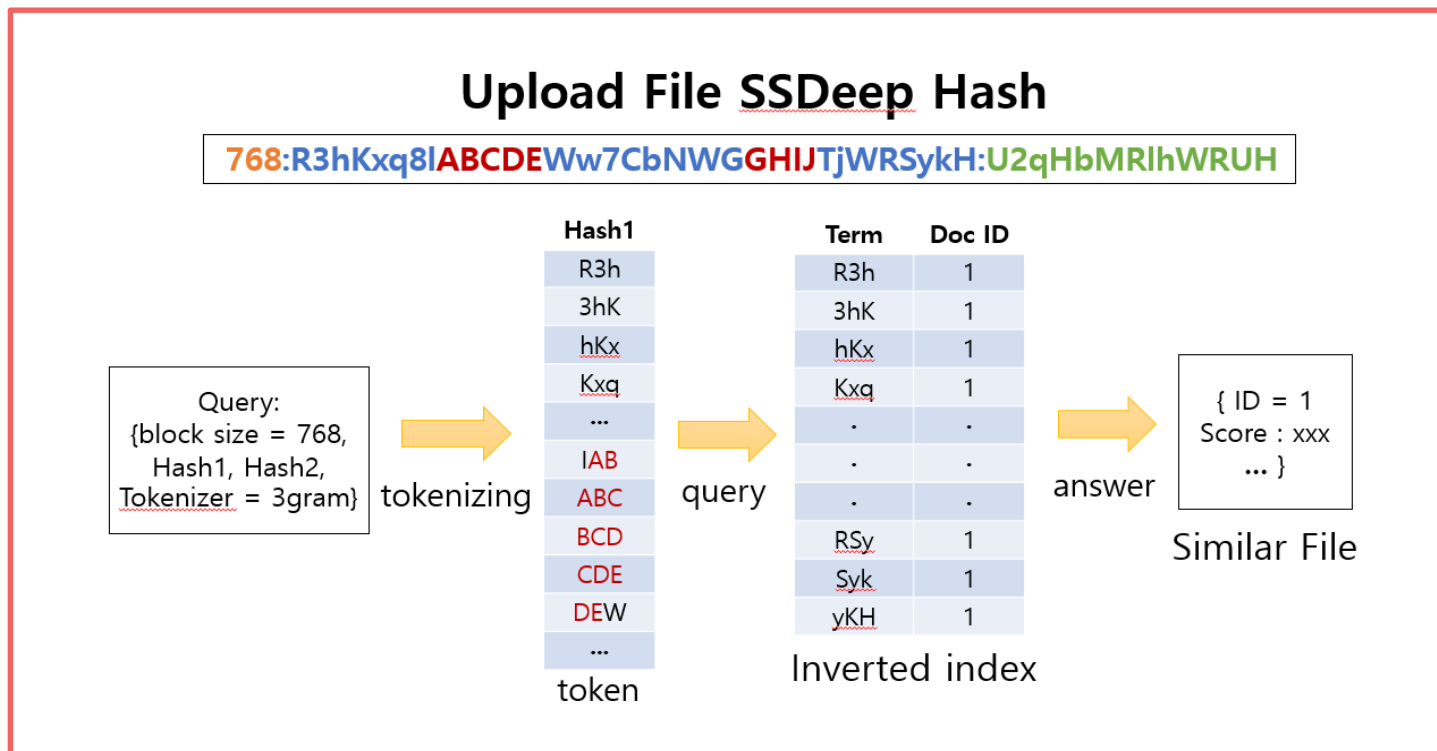
Doc ID

R3h	1
3hK	1
<u>hKx</u>	1
<u>Kxq</u>	1
.	.
.	.
.	.
<u>RSy</u>	1
<u>Syk</u>	1
<u>yKH</u>	1

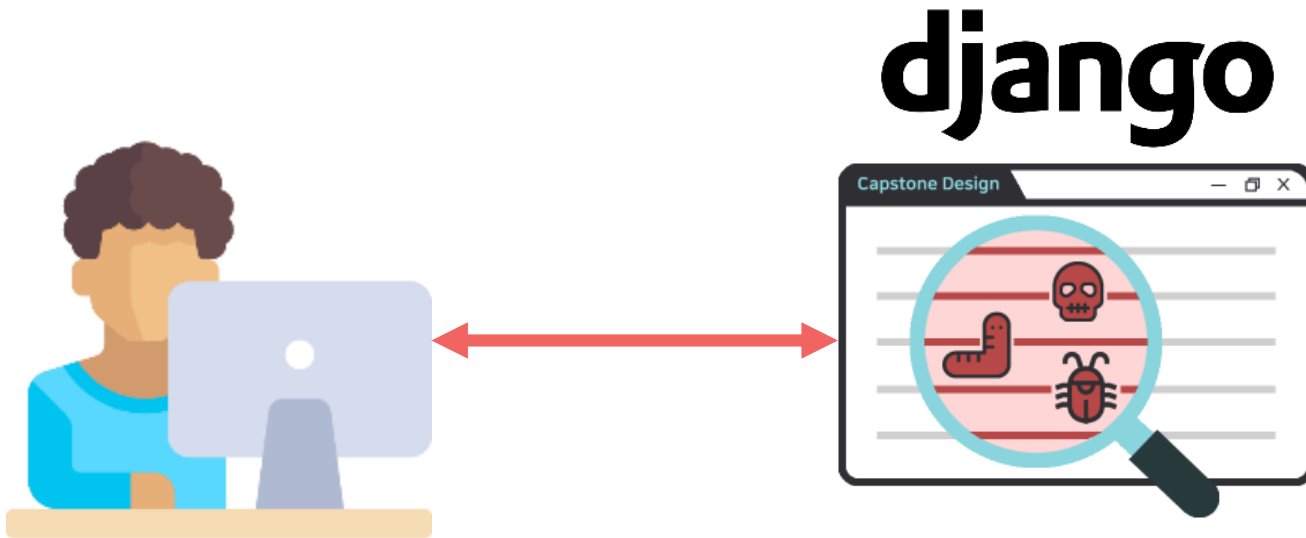
Inverted index

Tokenize된 word를 빠른 검색을 위해 역인덱싱

06 업로드 한 파일과 유사한 파일 검색



유사한 파일을 찾는 질의의 과정



01 Bootstrap을 이용한 반응형 웹 제작



브라우저의 폭 768px 이상일 때 웹 초기화면



브라우저의 폭 768px 이하일 때 변경된 네비게이션 바

02 파일 업로드 방법



NCNP: MASK

분석

통계

Sign in

파일 정보

파일 MD5	virusign.com_00cee1918c8925a2a7782f1a0d5759c7
바이러스 유무	True
바이러스 분석 결과	None
리포트 수집 날짜	None

AI

SAFE

100%

시그나처

Primary

유사 파일

사용자에게 보여지는 분석 결과 화면

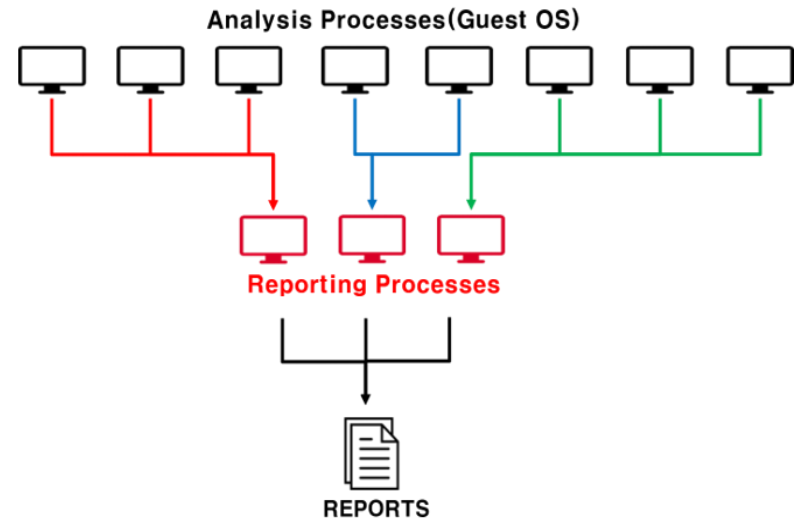
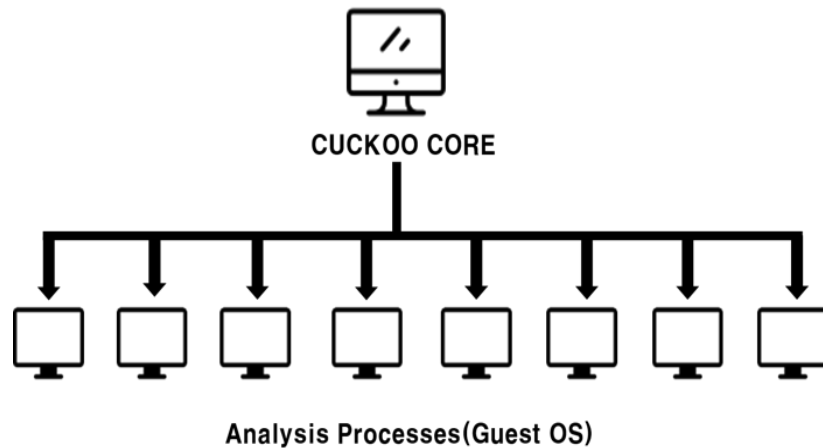
크롤링 자동화 >> 수동화

크롤러를 개발하여 악성코드를 수집하려 하였으나
유료서비스나 토렌트 등을 이용해야 하기 때문에 자동화가 어려움
수집 채널을 수동으로 변경함

04. 향후 추진계획

동적 분석

분석 프로세스와 리포팅 프로세스를 분리함으로써 시스템의 안정성을 높임



04. 향후 추진계획

라벨링

더 정확한 악성코드의 특징에 따른 분류를 위해 세분화된 라벨을 제작

현재 사용중인 7가지의 라벨을 기준으로 잡고 세분화
악성코드의 정적/동적 분석 정보에서 뽑은 특징점을 이용한 실험

AWS 인스턴스를 이용한 악성코드 분산 분석 자동화

인스턴스를 프로그램을 통해 조작 (python boto3 라이브러리...)
인스턴스에 악성코드를 분산 업로드 (신규 분석 요청시 인스턴스에서 직접 파일 보냄)

04. 향후 추진계획

딥러닝

피처 해싱을 다양하게 해볼 예정

다양한 기법으로 피처 해싱을 시도하여 모델의 성능 향상을 시도함

하이브리드 모델 개발

동적 분석 결과로부터 추출된 피처도 적용되는 동적 분석 모델과
정적 분석 결과와 동적 분석 결과를 복합적으로 사용하는 하이브리드 모델 개발을 목표로 함

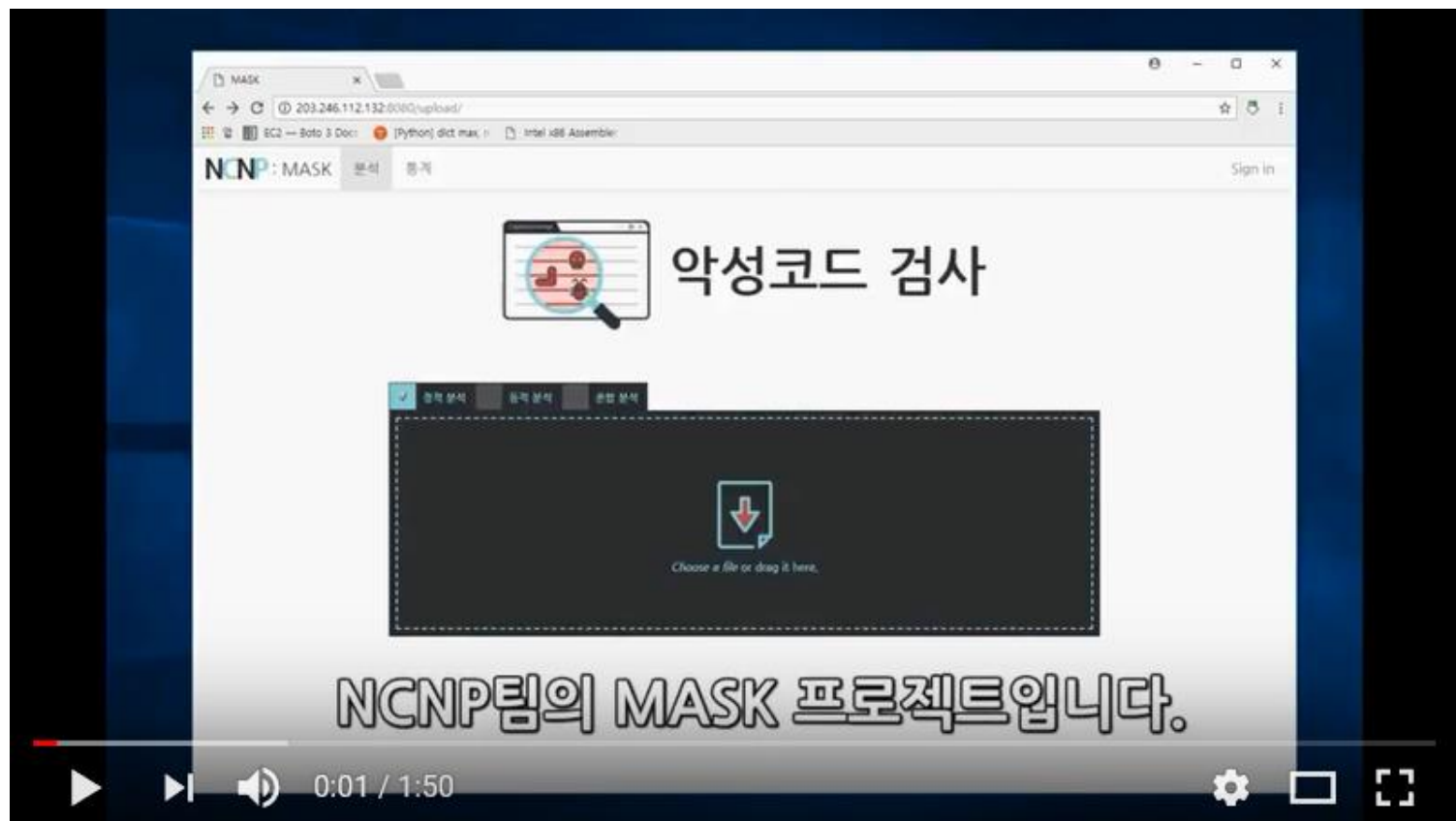
파일의 정적, 동적, 혼합 분석 기능 제공

제공중인 정적 분석을 포함하여 동적, 혼합 분석 기능을 모두 제공할 예정

분석 결과 화면에 정보 추가

동적 분석 시에 나타나는 파일의 C&C 서버를 시각화하여 제공
매일 분석되는 악성코드의 수와 발견되는 종류를 도식화 한 표 등 각종 자료 제공

05. 시연 영상



THANK YOU
Q&A