

세미나

2024.04.04(목)

모바일시스템공학과 20학번 조민혁

INDEX

01

논문 리뷰

02

2주간 해온 공부

03

앞으로의 공부 방향

01

논문 리뷰

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

❖ 서론

- 사용자에게 다양한 정보 및 기능을 제공하는 '**커넥티드 카**' 등장
 - 모바일 기기 분야 -> **Android Auto, Apple CarPlay** 플랫폼 제공
 - 또한, **OBD-II**를 통한 차량 진단 정보를 모바일 App을 통해 확인할 수 있음
 - ICT 기술이 많이 적용되면서 **사이버 보안 위협 및 사고 가능성이 높아짐**
- * OBD-II : 고수준의 통신 프로토콜, OBD-II 스캐너가 Bluetooth 또는 Wi-Fi 방식으로 사용자에게 전달
- * 인포카 앱 : OBD-II 기반 모바일 App, 차량 진단 Data를 저장하고 가공하여 사용자에게 전달

❖ 목표

- " 사용자 이벤트 기반 시나리오 설계 후 데이터 획득 이후 데이터 분석 "
- " 이를 통한 사용자의 동선 및 행위 재구성 "

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

Time	Function	Event	OBD-II Scanner
17:19	Power on	차량 시동 On	
17:20	Bluetooth Connection	블루투스로 인포카 앱과 연동	OBD-II 포트와 연결
17:23	Drive	주행 시작	
17:38	Power off	차량 시동 Off	OBD-II 포트에서 제거하지 않음
17:39	Power on	차량 시동 On	
17:39	Bluetooth Connection	블루투스로 인포카 앱과 연동	OBD-II 포트와 연결
17:40	Drive	주행 시작	
17:54	Power off	차량 시동 Off	
17:55			OBD-II 포트에서 제거

[표 2]. 인포카 앱 데이터 생성 시나리오

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

❖ 인포카 App 포렌식 과정

Step1) 루팅 진행

- OEM 잠금 해제 -> 커스텀 리커버리 이미지 플래시 -> Magisk 설치 후 루트 권한 획득

Step2) 루팅 후 데이터 추출 (/data 파티션의 시스템 이미지 획득)

- CMD1에서 'adb push' 명령을 통해 Busybox App을 모바일 기기에 설치 (App의 다양한 명령어 유틸리티 사용 가능)
- mount 명령어를 통해 /data 파티션의 마운트 위치 확인
- CMD2에서 'adb forward tcp:8888 tcp:8888' 명령을 통해 '안드로이드 기기'와 'PC'의 통신 환경 구축
- CMD1에서 'dd if=/dev/block/sda24 | busybox nc -l -p 8888' 명령을 통해 /data 파티션 이미징
- CMD2에서 이미지 파일을 8888 포트를 통해 PC에서 수신하도록 함
- PC 작업 영역에 수신 확인

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

❖ 실습 - 통신 환경 구성 및 dd 명령어를 통한 데이터 획득

```
PS C:\Users\cgumg\Desktop\platform-tools-latest-windows\platform-tools> .\adb.exe forward tcp:8888 tcp:8888
PS C:\Users\cgumg\Desktop\platform-tools-latest-windows\platform-tools> nc 127.0.0.1 8888 > test_data.dd
```

통신 환경 구성

```
dreamlteks:/data/data # dd if=/dev/block/sda24 | busybox nc -l -p 8888
```

dd 명령어를 통한 메모리 이미징

test_data.dd

2024-04-03 오후 6:33

DD 파일

398,108KB

생성된 dd 파일 확인

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

❖ 인포카 App 데이터 분석 - Autopsy를 통한 분석

- 인포카 App 데이터 위치 : '/data/mureung.obdproject'
- "/data/mureung.obdproject/databases" 하위 경로에 DB 파일 목록 존재
 - * WMI.db : 차량의 제조업체 정보
 - * InfoCar.db : 운전자의 주행 정보 & 개인정보, 차량 정보 (주목해야할 DB 파일)
 - * DTC.db : 차량 진단 코드명
- InfoCar.db의 테이블
 - * USERINFO : 사용자 이름, 이메일, 차량 연식, 차량 모델명, 연료 확인 가능
 - * DRVREC : 운행 시작/종료 시각, 총 주행거리, 도착지점 및 출발지점, 연비, 평균 주행속도 확인 가능
 - * SRCREC : RPM, APS, TPS, RPS, 위/경도 등의 데이터 1초 단위로 저장

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

Table 1 entries Page 1 of 1 [Export to CSV](#)

userName	userEmail	carName	carMaker	carYear	carModel	carFuelT...	allDistan...	avrFuel...	obdSN	drvFinishTime
장지현	cpooduam82730@gmail.com	cn7	현대	2020	아이반	Gasoline	40.500614	8.671122	661E11090100	20220714175507

[그림 6]. USERINFO 테이블

Table	SRCREC	5342 entries	Page 34 of 54	↔	Export to CSV					
Jid	srcValue	userSN	realTime	srcLatit...	srcLong...	srcSpeed	srcRPM	srcAPS	srcTPS	srcRPS
3328	22071401	1	20220714161111	37.3244442	127.12603...	0.0	0.0	14.90196	14.509804	0.0
3329	22071401	1	20220714161112	37.3244442	127.12603...	0.0	0.0	14.90196	14.509804	0.0
3330	22071401	1	20220714161113	37.3244442	127.12603...	0.0	0.0	14.90196	14.509804	0.0
3331	22071401	1	20220714161114	37.3244442	127.12603...	0.0	0.0	14.90196	14.509804	0.0
3332	22071402	1	20220714172026	0.0	0.0	0.0	704.0	14.90196	0.0	0.0
3333	22071402	1	20220714172027	0.0	0.0	0.0	704.0	14.90196	0.0	0.0
•										
•										
•										
3513	22071402	1	20220714172327	0.0	0.0	0.0	700.0	14.90196	14.509804	0.0
3514	22071402	1	20220714172328	0.0	0.0	0.0	700.0	14.90196	14.509804	0.0
3515	22071402	1	20220714172329	0.0	0.0	0.0	676.5	14.90196	14.509804	0.0
3516	22071402	1	20220714172330	37.2395975	127.179017	2.0	676.5	14.90196	14.509804	0.0
3517	22071402	1	20220714172331	37.2395841	127.179023	2.0	676.5	14.90196	14.509804	0.0

[그림 8]. SRCREC 테이블

Table	DRVREC	4 entries	Page 1 of 1	Export to CSV					
Jid	drvValue	userSN	drvKey	drvHid...	drvStartTime	drvFinishTime	drvLatitude	drvLongitude...	drvAddress
1	22070801	1	9874806	0	20220708155815	20220708163752	37.361026	127.1194287	대한민국 경기도 성남시 분당구 결자동 202
2	22063001	1	9881908	0	20220630154426	20220630161410	37.381182	127.1230882	대한민국 평남시 서천고등학교
3	22071401	1	9929943	0	20220714151436	20220714161115	37.3244442	127.1260347	대한민국 경기도 용인시 수지구 죽전1동 1399-5
4	22071402	1	null	0	20220714172026	20220714175507	37.3236212	127.1242471	대한민국 경기도 용인시 수지구 죽전1동 147

[그림 7]. DRVREC 테이블

논문 리뷰 - 사용자 이벤트 기반 안드로이드 인포카 앱 포렌식 연구

❖ 결론

- 논문에서 제시한 목표 : 인포카 앱 Data를 통한 사용자 동선 및 행위 검증
- 'InforCar.db' 가 목표에 충족되는 의미있는 DB 파일 이었음
- 이전 장에서 보았던 것처럼 데이터 비교 분석을 통해 목표를 충족할 수 있었음

❖ 소감

- 해당 논문을 포렌식 공부하기 전, 공부한 후 총 2번 반복하여 읽었음
- 공부하기 전에 읽었을 당시 모르는 용어들이 많아 실험 흐름에 대해 깨닫지를 못했음
- 공부한 후 읽었을 때는 직접 실습해본 내용도 있기에 이해하기에 수월했음
- " 논문을 통해 루팅 적용 방법 및 데이터 획득, 분석 방법을 깨달음 "
- " DB 파일에 대한 분석을 위해 SQL 관련 공부가 필요하다고 느낌 "

02

2주 동안 공부한 내용

메모리 포렌식 - CriDex

❖ CriDex 문제 풀이(1) - Volatility 분석 도구 활용

Step1) 운영 체제 식별

-> volatility_2.6_win64_standalone.exe -f <이미지 파일 이름> imageinfo

Step2) 프로세스 로그 생성

-> pslist, psscan, pstree, psxview

Step3) 네트워크 분석

-> connections.log

* pslist : 시간 순서대로 프로세스 목록 출력

* psscan : offset 순서대로 프로세스 목록 출력

* pstree : PID와 PPID 기반 구조화

* psxview : pslist와 psscan을 한눈에 볼 수 있음

메모리 포렌식 - CriDex

❖ 운영체제 식별에 관한 부분

```
PS C:\Users\cgumg\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG search...
          : Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
          : AS Layer1           : IA32PagedMemoryPae (Kernel AS)
          : AS Layer2           : FileAddressSpace (C:\Users\cgumg\Desktop\cridex\cridex.vmem)
          : PAE type            : PAE
          : DTB                 : 0x2fe000L
          : KDBG                 : 0x80545ae0L
          : Number of Processors : 1
          : Image Type (Service Pack) : 3
          : KPCR for CPU 0       : 0xffdff000L
          : KUSER_SHARED_DATA    : 0xffdf0000L
          : Image date and time  : 2012-07-22 02:45:08 UTC+0000
          : Image local date and time : 2012-07-21 22:45:08 -0400
```

-> 'WinXPSP2x86' 인 것을 확인할 수 있음

메모리 포렌식 - CriDex

❖ 프로세스 로그 파일 - NotePad++를 통한 분석

1	Offset (V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
2										
3	0x823c89c8	System	4	0	53	240	-----	0		
4	0x822f1020	smss.exe	368	4	3	19	-----	0	2012-07-22 02:42:31 UTC+0000	
5	0x822a0598	csrss.exe	584	368	9	326	0	0	2012-07-22 02:42:32 UTC+0000	
6	0x82298700	winlogon.exe	608	368	23	519	0	0	2012-07-22 02:42:32 UTC+0000	
7	0x81e2ab28	services.exe	652	608	16	243	0	0	2012-07-22 02:42:32 UTC+0000	
8	0x1e2a3b8	lsass.exe	664	608	24	330	0	0	2012-07-22 02:42:32 UTC+0000	
9	0x82311360	svchost.exe	824	652	20	194	0	0	2012-07-22 02:42:33 UTC+0000	
10	0x81e29ab8	svchost.exe	908	652	9	226	0	0	2012-07-22 02:42:33 UTC+0000	
11	0x823001d0	svchost.exe	1004	652	64	1118	0	0	2012-07-22 02:42:33 UTC+0000	
12	0x821dfda0	svchost.exe	1056	652	5	60	0	0	2012-07-22 02:42:33 UTC+0000	
13	0x8229650	svchost.exe	1220	652	15	197	0	0	2012-07-22 02:42:35 UTC+0000	
14	0x821dea70	explorer.exe	1484	1464	17	415	0	0	2012-07-22 02:42:36 UTC+0000	
15	0x81eb17b8	spoolsv.exe	1512	652	14	113	0	0	2012-07-22 02:42:36 UTC+0000	
16	0x81e7bda0	reader_sl.exe	1640	1484	5	39	0	0	2012-07-22 02:42:36 UTC+0000	
17	0x820e8da0	alg.exe	788	652	7	104	0	0	2012-07-22 02:43:01 UTC+0000	
18	0x821fcd80	wuauclt.exe	1136	1004	8	173	0	0	2012-07-22 02:43:46 UTC+0000	
19	0x8205bda0	wuauclt.exe	1588	1004	5	132	0	0	2012-07-22 02:44:01 UTC+0000	
20										

1	Offset (P)	Name	PID	PPID	PDB	Time created	Time exited
2							
3	0x00000000002029ab8	svchost.exe	908	652	0x079400e0	2012-07-22 02:42:33 UTC+0000	
4	0x0000000000202a3b8	lsass.exe	664	608	0x079400a0	2012-07-22 02:42:32 UTC+0000	
5	0x0000000000202ab28	services.exe	652	608	0x07940080	2012-07-22 02:42:32 UTC+0000	
6	0x0000000000207bda0	reader_sl.exe	1640	1484	0x079401e0	2012-07-22 02:42:36 UTC+0000	
7	0x000000000020b17b8	spoolsv.exe	1512	652	0x079401c0	2012-07-22 02:42:36 UTC+0000	
8	0x0000000000225bda0	wuauclt.exe	1588	1004	0x07940200	2012-07-22 02:44:01 UTC+0000	
9	0x0000000000225bda0	alg.exe	788	652	0x07940140	2012-07-22 02:43:01 UTC+0000	
10	0x000000000023dea70	explorer.exe	1484	1464	0x079401a0	2012-07-22 02:42:36 UTC+0000	
11	0x000000000023dfda0	svchost.exe	1056	652	0x07940120	2012-07-22 02:42:33 UTC+0000	
12	0x000000000023fcd80	wuauclt.exe	1136	1004	0x07940180	2012-07-22 02:43:46 UTC+0000	
13	0x00000000002495650	svchost.exe	1220	652	0x07940160	2012-07-22 02:42:35 UTC+0000	
14	0x00000000002498700	winlogon.exe	608	368	0x07940060	2012-07-22 02:42:32 UTC+0000	
15	0x000000000024a0598	csrss.exe	584	368	0x07940040	2012-07-22 02:42:32 UTC+0000	
16	0x000000000024f1020	smss.exe	368	4	0x07940020	2012-07-22 02:42:31 UTC+0000	
17	0x000000000025001d0	svchost.exe	1004	652	0x07940100	2012-07-22 02:42:33 UTC+0000	
18	0x00000000002511360	svchost.exe	824	652	0x079400c0	2012-07-22 02:42:33 UTC+0000	
19	0x000000000025c89c8	System	4	0	0x002fe000		
20							

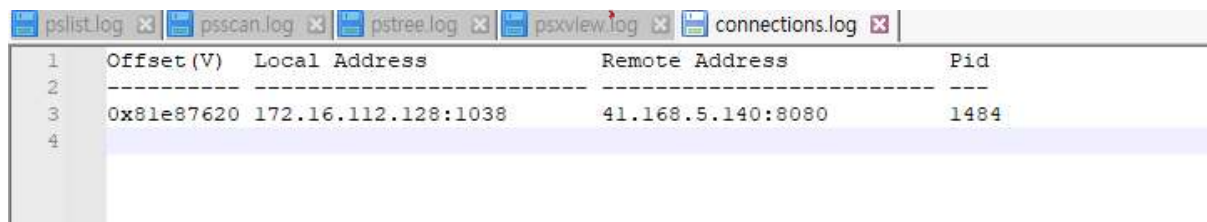
1	Name	Pid	PPid	Thds	Hnds	Time
2						
3	0x823c89c8:System	4	0	53	240	1970-01-01 00:00:00 UTC+0000
4	.. 0x822f1020:smss.exe	368	4	3	19	2012-07-22 02:42:31 UTC+0000
5	... 0x82298700:winlogon.exe	608	368	23	519	2012-07-22 02:42:32 UTC+0000
6	... 0x81e2ab28:services.exe	652	608	16	243	2012-07-22 02:42:32 UTC+0000
7 0x821dfda0:svchost.exe	1056	652	5	60	2012-07-22 02:42:33 UTC+0000
8 0x81eb17b8:spoolsv.exe	1512	652	14	113	2012-07-22 02:42:36 UTC+0000
9 0x81e29ab8:svchost.exe	908	652	9	226	2012-07-22 02:42:33 UTC+0000
10 0x823001d0:svchost.exe	1004	652	64	1118	2012-07-22 02:42:33 UTC+0000
11 0x8205bda0:wuauclt.exe	1588	1004	5	132	2012-07-22 02:44:01 UTC+0000
12 0x821fcd80:wuauclt.exe	1136	1004	8	173	2012-07-22 02:43:46 UTC+0000
13 0x82311360:svchost.exe	824	652	20	194	2012-07-22 02:42:33 UTC+0000
14 0x820e8da0:alg.exe	788	652	7	104	2012-07-22 02:43:01 UTC+0000
15 0x8229650:svchost.exe	1220	652	15	197	2012-07-22 02:42:35 UTC+0000
16	... 0x81e2a3b8:lsass.exe	664	608	24	330	2012-07-22 02:42:32 UTC+0000
17	.. 0x822a0598:csrss.exe	584	368	9	326	2012-07-22 02:42:32 UTC+0000
18	0x821dea70:explorer.exe	1484	1464	17	415	2012-07-22 02:42:36 UTC+0000
19	.. 0x81e7bda0:reader_sl.exe	1640	1484	5	39	2012-07-22 02:42:36 UTC+0000
20						

1	Offset (P)	Name	PID	pslist	psscan	thrdproc	pspid	csrss	session	deskthrd	ExitTime
2											
3	0x02498700	winlogon.exe	608	True	True	True	True	True	True	True	True
4	0x02511360	svchost.exe	824	True	True	True	True	True	True	True	True
5	0x022e8da0	alg.exe	788	True	True	True	True	True	True	True	True
6	0x020b17b8	spoolsv.exe	1512	True	True	True	True	True	True	True	True
7	0x0202ab28	services.exe	652	True	True	True	True	True	True	True	True
8	0x02495650	svchost.exe	1220	True	True	True	True	True	True	True	True
9	0x0207bda0	reader_sl.exe	1640	True	True	True	True	True	True	True	True
10	0x025001d0	svchost.exe	1004	True	True	True	True	True	True	True	True
11	0x02029ab8	svchost.exe	908	True	True	True	True	True	True	True	True
12	0x023fcd80	wuauclt.exe	1136	True	True	True	True	True	True	True	True
13	0x0225bda0	wuauclt.exe	1588	True	True	True	True	True	True	True	True
14	0x0202a3b8	lsass.exe	664	True	True	True	True	True	True	True	True
15	0x023dea70	explorer.exe	1484	True	True	True	True	True	True	True	True
16	0x023dfda0	svchost.exe	1056	True	True	True	True	True	True	True	True
17	0x024f1020	smss.exe	368	True	True	True	True	True	True	True	True
18	0x025c89c8	System	4	True	True	True	True	False	False	False	False
19	0x024a0598	csrss.exe	584	True	True	True	True	False	True	True	True
20											

메모리 포렌식 - CriDex

❖ 네트워크 분석에 대한 부분

```
PS C:\Users\cgumg\Desktop\crindex> volatility_2.6_win64_standalone.exe -f crindex.vmem --profile=WinXPSP2x86 co  
nnections > connections.log
```



	Offset (V)	Local Address	Remote Address	Pid
1				
2				
3	0x81e87620	172.16.112.128:1038	41.168.5.140:8080	1484
4				

-> '8080' 포트를 통해 41.168.5.140 IP와 통신하고 있는 것을 알 수 있음

메모리 포렌식 - CriDex

❖ CriDex 문제 풀이(2) - Volatility 분석 도구 활용

Step4) cmd 분석

-> cmdline, cmdscan, consoles

Step5) 파일 분석 및 덤프

-> filescan, dumpfiles

Step6) 프로세스 세부 분석

-> procdump, memdump

Step7) 로그 분석을 통한 시나리오 작성

- 메모리 포렌식, 로그 분석으로 시나리오를 작성 가능해짐

- 결과적으로 악성 PDF 파일 읽어드림 -> 취약점 -> 은행 관련 피싱이라는 것을 알 수 있었음

* cmdline : 프로세스가 실행될 때의 인자값 확인 가능

* cmdscan, consoles : 콘솔에 입력한 값 확인 가능

* filescan : 메모리에서 파일 시스템 정보 스캔

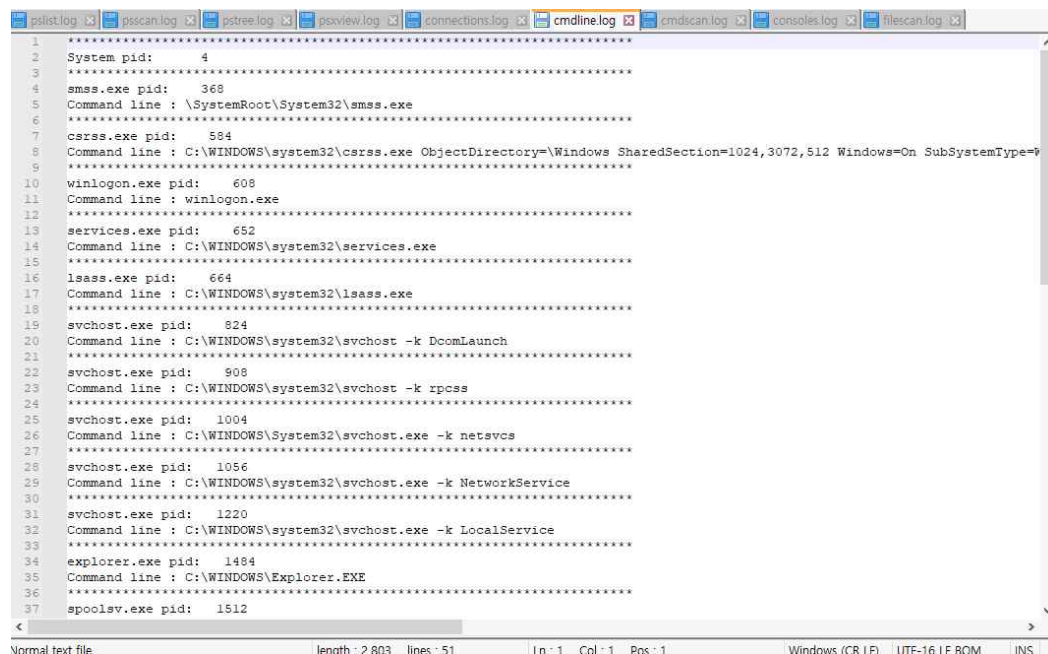
* dumpfiles : 메모리 덤프 파일에서 특정 파일 추출

* memdump : 메모리 덤프 파일에서 특정 프로세스 메모리
추출

* procdump : 프로세스의 실행 파일 추출

메모리 포렌식 - CriDex

❖ cmd에 대한 부분



```
1 *****
2 System pid: 4
3 *****
4 smss.exe pid: 368
5 Command line : \SystemRoot\System32\smss.exe
6 *****
7 csrss.exe pid: 594
8 Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512 Windows=On SubSystemType=
9 *****
10 winlogon.exe pid: 608
11 Command line : winlogon.exe
12 *****
13 services.exe pid: 652
14 Command line : C:\WINDOWS\system32\services.exe
15 *****
16 lsass.exe pid: 664
17 Command line : C:\WINDOWS\system32\lsass.exe
18 *****
19 svchost.exe pid: 824
20 Command line : C:\WINDOWS\system32\svchost -k DoomLaunch
21 *****
22 svchost.exe pid: 908
23 Command line : C:\WINDOWS\system32\svchost -k rpcss
24 *****
25 svchost.exe pid: 1004
26 Command line : C:\WINDOWS\system32\svchost.exe -k netsvcs
27 *****
28 svchost.exe pid: 1056
29 Command line : C:\WINDOWS\system32\svchost.exe -k NetworkService
30 *****
31 svchost.exe pid: 1220
32 Command line : C:\WINDOWS\system32\svchost.exe -k LocalService
33 *****
34 explorer.exe pid: 1484
35 Command line : C:\WINDOWS\Explorer.EXE
36 *****
37 spoolsv.exe pid: 1512
38 *****
```

```
PS C:\Users\cgung\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem --profile=WinXPSP2x86 cm
dline > cmdline.log
Volatility Foundation Volatility Framework 2.6
PS C:\Users\cgung\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem --profile=WinXPSP2x86 cm
dscan > cmdscan.log
Volatility Foundation Volatility Framework 2.6
PS C:\Users\cgung\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem --profile=WinXPSP2x86 co
nsoles > consoles.log
```

메모리 포렌식 - CriDex

❖ 파일 분석 및 덤프에 대한 부분 - dumpfiles

```
PS C:\Users\cgumg\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x00000000023ccf90 -D .\files\ -n
Volatility Foundation Volatility Framework 2.6
ImageSectionObject 0x023ccf90 None \Device\HarddiskVolume1\Program Files\Adobe\Reader 9.0\Reader\reader_sl.exe
DataSectionObject 0x023ccf90 None \Device\HarddiskVolume1\Program Files\Adobe\Reader 9.0\Reader\reader_sl.exe
```

<dumpfiles 하는 과정>

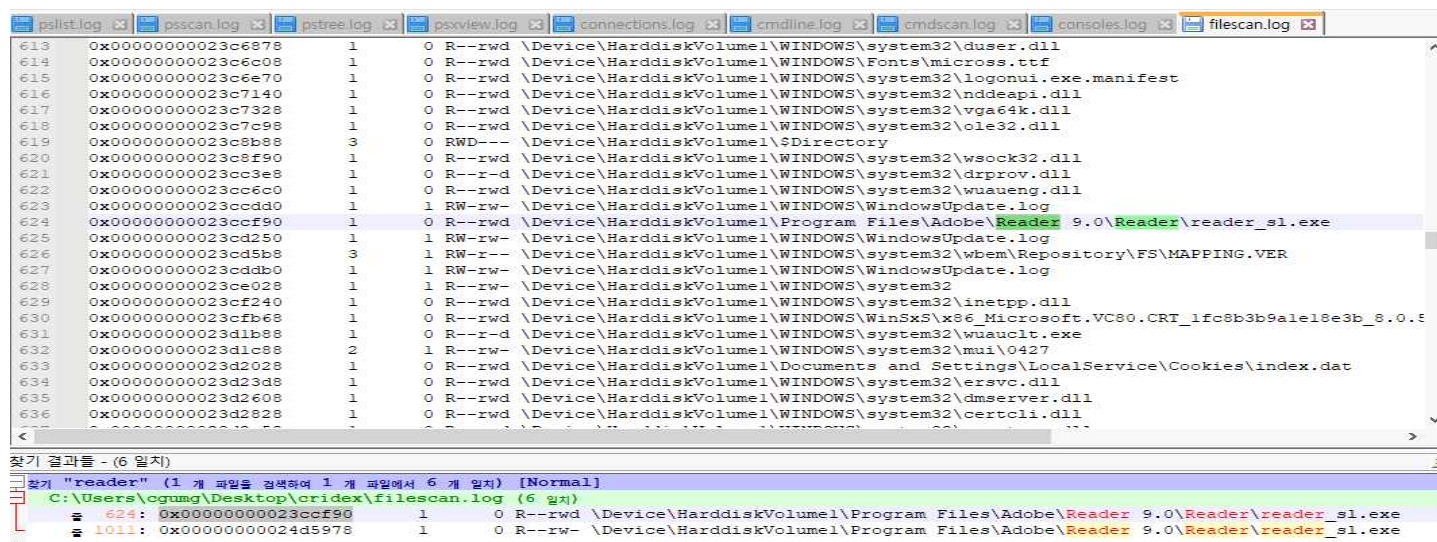
* -Q : Offset 지정, -D: 저장할 경로, -n: dumpfile 대상 이름

 file.None.0x82137c08.reader_sl.exe	2024-04-03 오후 7:35	디스크 이미지 파일	31KB
 file.None.0x822116f0.reader_sl.exe.dat	2024-04-03 오후 7:35	DAT 파일	32KB

<dumpfiles로 추출된 메모리 영역에서의 특정 파일>

메모리 포렌식 - CriDex

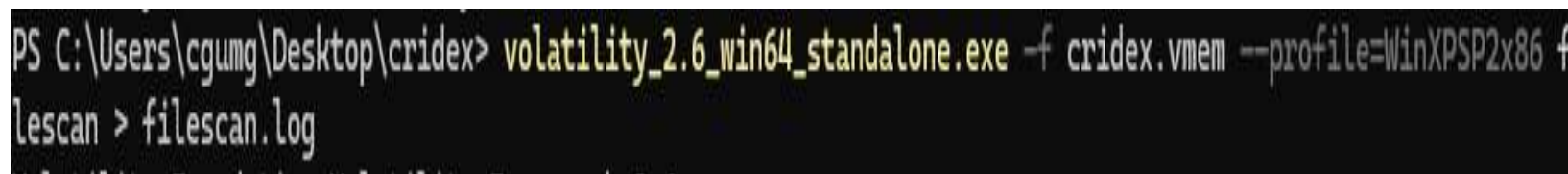
❖ 파일 분석 및 덤프에 대한 부분 - filescan



The screenshot displays the filescan.log file, which contains a list of file operations. The search window at the bottom shows the following results:

Offset	File Path
624: 0x0000000023ccf90	\\Device\\HarddiskVolume1\\Program Files\\Adobe\\Reader 9.0\\Reader\\reader_sl.exe
1011: 0x0000000024d5978	\\Device\\HarddiskVolume1\\Program Files\\Adobe\\Reader 9.0\\Reader\\reader_sl.exe

<filescan.log에서 의심 프로세스를 검색하여 경로를 확인하고 Offset을 확인하는 과정>



```
PS C:\Users\cgumg\Desktop\crindex> volatility_2.6_win64_standalone.exe -f crindex.vmem --profile=WinXPSP2x86 filescan > filescan.log
```

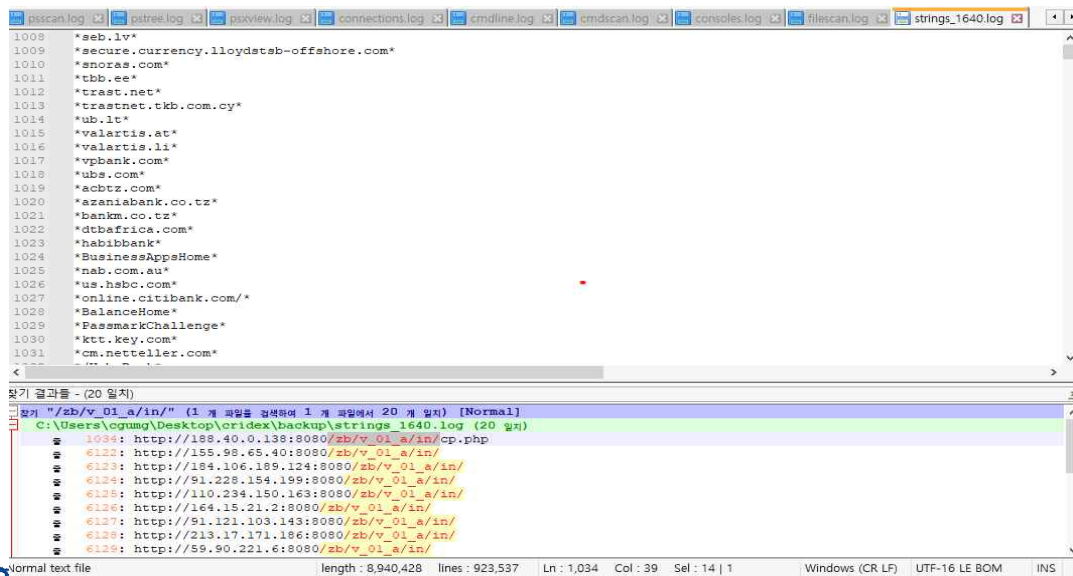
<filescan하는 과정>

메모리 포렌식 - CriDex

❖ 프로세스 세부 분석에 대한 부분 - memdump

```
PS C:\Users\cgumg\Desktop\crindex> volatility_2.6_win64_standalone.exe -f crindex.vmem --profile=WinXPSP2x86 me
mdump -p 1640 -D .\dumps\
Volatility Foundation Volatility Framework 2.6
*****
Writing reader_sl.exe [ 1640] to 1640.dmp
```

< memdump하는 과정 >



< memdump로 추출된 dmp 파일 string 변환 후 분석 >

메모리 포렌식 - CriDex

❖ 프로세스 세부 분석에 대한 부분 - procdump

```
PS C:\Users\cgumg\Desktop\cridex> volatility_2.6_win64_standalone.exe -f cridex.vmem --profile=WinXPSP2x86 procdump -p 1640 -D .\dumps
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase Name Result
-----
0x81e7bda0 0x00400000 reader_sl.exe OK: executable.1640.exe
```

<procdump를 통한 특정 프로세스 실행 파일 추출 과정>



2024-04-03 오후 7:52 응용 프로그램

29KB

<생성된 특정 프로세스의 실행 파일>

이외에도 공부한 내용

1. CriDex와 같은 방식으로 GrrCon2015 분석

2. 갤럭시 s8 루팅 실습 (루팅 후 이미 루팅이 완료된 휴대폰이란 걸 알았음)

* 스스로 실습해본 루팅의 과정

- TWRP 펌웨어 파일 가져오기 -> 다운로드모드 접속 -> Odin 실행 -> 리커버리 모드 접속 ->
Magisk에서 권한 획득 -> adb에서 su 명령어를 통해 root 권한 가져옴

3. 윈도우 레지스트리의 개념 및 분석 기법 공부 (복습 필요)

4. 모바일 포렌식 - 안드로이드 포렌식 부분 공부 (진행중)

03

앞으로의 공부 방향

앞으로의 공부 방향

❖ 자동차 포렌식

- 책을 통한 안드로이드 포렌식 관련 공부를 하면서 실습을 진행 예정
- 이전에 이해하지 못했던 논문을 복습하며 깨닫기
- 논문에 주어진 실험 과정 직접 실습해보기
- 현재 하고 있는 공부를 마친 후 주어진 과제를 해결하면서 실력 쌓기

❖ 메모리 포렌식

- 남아있는 메모리 포렌식 강의를 마저 수강하기
- 메모리 포렌식을 통한 로그 분석에 대한 실력을 향상시키기

앞으로의 공부 방향

❖ 모바일 포렌식 분야 중 인스타그램, 카카오톡과 같은 SNS 포렌식 연구

- 여러 정보를 찾아본 결과 해당 분야에 대한 연구 사례가 적음
- 최근 해당 주제에 대해 사회적 문제들이 많이 발생하고 있음
Ex) 개인 정보 유출, 사기, 스토킹, 악성코드 유포
- 해당 연구를 통해 사용자 행위 분석이 가능할 것이라고 기대



들어주셔서 감사합니다 !
