208 – iOS Fridump

Team Information

Team Name: DogeCoin

Team Member: Dongbin Oh, Donghyun Kim, Donghyun Kim, Yeongwoong Kim

Email Address: dfc-dogecoin@naver.com

Instructions

Description Here are memory dumps of iOS's Settings and Safari.

Target	Hash (SHA-256)
Settings.zip	b13d63d56e47f8274cb38e29c9ba0b7224ceab21d4f263cae7baf50920b6f1e5
Safari.zip	ef76f60e254cc6616800a15ab3920c438c0d88a1671325651dfefb87b2afaa63

Questions

1. Identify the device model by analyzing the Settings. (25 points)
2. Identify the iOS version by analyzing the Settings. (25 points)
3. Find WiFi names recently connected by analyzing the Settings. (at least 2 WiFi names) (50 points)
4. The Safari memory dump contains the web site addresses accessed by the user. Find 10 or more, but exclude duplicates. (100 points)

Teams must:
- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).

Tools used:

Name:	scalpel	Publisher:	Brian Carrier
Version:	2.0		
URL:	https://github.com/sleuthkit/scalpel		

Name:	010editor	Publisher:	SweetScape Software.Inc
Version:	12.0		
URL:	https://www.sweetscape.com/010editor/		

Name:	plutil	Publisher:	Apple
Version:	-		
URL:	https://www.apple.com/kr/itunes/		

Step-by-step methodology:

0. Preprocessing

2 가지 관점으로 주어진 iOSFridump 의 각 프로세스 메모리 덤프로부터 아티팩트를 수집하였다. String Search 의 경우는 Fridump 로 수집된 데이터를 분석하는데 가장 일반적인 방법으로 주로 메모리 상에 ID 나 Password 와 같은 민감한 데이터가 분석 중 쉽게 노출 될 수 있는지 여부를 위주로 그동안 연구가 진행되었다.[1] 그러나 이러한 Approach 는 문자열이 어떠한 근거로 생성 되었는지에 대한 Root Cause Analysis 의 한계가 존재하여 정확성에 문제가 있을 수 있다. 보완책으로는 해당 String 의 빈도를 Count 해 우연히 발생한 String 이 아니라는 것을 확인할 수 있다. 또 다른 Approach 는 Carving 이다. Carving 의 대상은 File 이 될 수도 있고 SQLite 나 XML 의 특정 Key 구조나 Record 구조가 될 수 있다. 메모리 영역에는 필연적으로 프로그램이나 어플리케이션이 사용하는 파일의 정보를 전체 또는 일부 담고 있다. 이를 Carving 의 기법을 통해 복구 후 원본 파일시스템에 존재하는 파일과 특징을 비교하면 분석의 정확성 또한 String Search 방법보다 높일 수 있다. 본 방법에서는 Carving 과 String Approach 의 장단점을 활용하여 Hybrid 분석을 시도하되, 제한적인 상황에서만 String Search 만 사용하여 분석의 신뢰도를 보장하였다. 분석을 위해 전처리는 다음과 같이 진행하였다.

- **String 추출**

String 추출은 linux 기본 명령어인 strings 를 이용하여 다음과 같이 진행하였다.

```
(base) sin90@odongbin-ui-MacBookPro safari_dump % strings * >> ../safari_dump.txt
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x100554000_dump.data truncated or malformed object (LC_SEGMENT_64 command 2 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x1007f0000_dump.data truncated or malformed object (LC_SEGMENT_64 command 0 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x1008b0000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x100d4c000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x100d8c000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x100db8000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x1010e4000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
/Library/Developer/CommandLineTools/usr/bin/strings: object: 0x101190000_dump.data truncated or malformed object (LC_SEGMENT_64 command 1 fileoff field plus filesize field extends past the end of the file)
```

[그림 1] Strings 추출

- **File Carving**

iOS 의 경우 대부분의 아티팩트들이 SQLite 나 Plist(bplist) 같은 형식이므로, 아래와 같이 Rule 을 이용하여 Carving 을 시도하였다. Rule 은 iPhone and iOS Forensics[2]라는 책을 참고하여 진행하였다.

```
#ext case size header footer
db y 409600 SQLite\x20format
plist y 4096 <plist </plist
bplist y 4096 \x62\x70\x6c\x69\x73\x74\x30\x30
sqlitedb y 819200 SQLite\x20format
```

[그림 2] Scalpel Rule

[1] https://ch4njun.tistory.com/163?category=794672

[2] iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices / Book • 2011

Carving 도구로는 scalpel을 사용하였다. cat을 이용하여 Dump 파일을 하나로 merge하였다. 이후 도구를 통해 원하는 파일을 획득할 수 있었다.

```
(base) sin90@odongbin-ui-MacBookPro safari_dump % cat * >> ../safari_dump.dump
(base) sin90@odongbin-ui-MacBookPro safari_dump %
```

[그림 3] 메모리 덤프 병합

```
sansforensics@siftworkstation: ~/Desktop
$ scalpel -c scalpel.conf  -o ./scalpel_output_settings setiings_dump.dump
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.

Opening target "/home/sansforensics/Desktop/setiings_dump.dump"

Image file pass 1/2.
setiings_dump.dump: 100.0% |*****************************************************************************************************************************************************************|  757.0 MB    00:00 ETA
Allocating work queues...
Work queues allocation complete. Building carve lists...
Carve lists built.  Workload:
db with header "\x53\x51\x4c\x69\x74\x65\x20\x66\x6f\x72\x6d\x61\x74" and footer "" --> 1 files
plist with header "\x3c\x70\x6c\x69\x73\x74" and footer "\x3c\x2f\x70\x6c\x69\x73\x74" --> 5 files
bplist with header "\x62\x70\x6c\x69\x73\x74\x30\x30" and footer "" --> 1435 files
sqlitedb with header "\x53\x51\x4c\x69\x74\x65\x20\x66\x6f\x72\x6d\x61\x74" and footer "" --> 1 files
Carving files from image.
Image file pass 2/2.
setiings_dump.dump: 100.0% |*****************************************************************************************************************************************************************|  757.0 MB    00:00 ETA
Processing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 1442, elapsed = 2 seconds.
```

[그림 4] Scalpel 구동

1. Identify the device model by analyzing the Settings. (25 points)

2. Identify the iOS version by analyzing the Settings. (25 points)

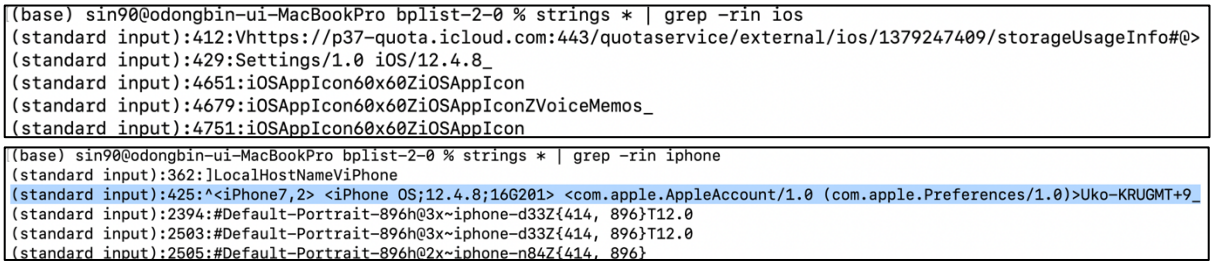
Hybrid Approach를 적용하기 위해 Scalpel을 이용하여 Memory Dump로부터 추출한 plist, bplist를 대상으로 String Search를 실행하였다. 본래의 iOS Forensics 상에서 문제에서 요구하는 Device Model과 iOS Version의 경우 아래의 그림과 같이 info.plist에서 확인할 수 있다.[3]

Key	Type	Value
Root	dict	
Applications	dict	
Build Version	string	16C101
Device Name	string	SCARIF_CITADEL_TOWER
Display Name	string	SCARIF_CITADEL_TOWER
GUID	string	CBAEB59226D11FFB1
ICCID	string	
IMEI	string	
Installed Applications	array	
Last Backup Date	date	2019-05-04 16:47:33
MEID	string	
Phone Number	string	
Product Name	string	iPhone X
Product Type	string	iPhone10,3
Product Version	string	12.1.2
Serial Number	string	
Target Identifier	string	cf88902bccf8e24459831b3eabd5c6d2462d7240
Target Type	string	Device
Unique Identifier	string	cf88902bccf8e24459831b3eabd5c6d2462d7240
iTunes Files	dict	
iTunes Settings	dict	
iTunes Version	string	12.9.2.6

[그림 5] info.plist 구조

이를 근거로 하여 scalpel을 통해 추출한 plist 파일과, bplist 파일을 대상으로 “iOS”, “iphone” 과 같은 키워드를 이용하여 검색을 수행하였다. Strings와 grep을 이용하여 수행하였으며 문제에서 원하는 정답과 가장 관련된 내용은 아래와 같다.

[3] https://farleyforensics.com/2019/04/14/forensic-analysis-of-itunes-backups/



[그림 6] String 분석을 통해 확인한 iPhone 모델 및 iOS 버전 정보

추출한 bplist 파일과 Plist 파일이 plutil 과 같은 도구[4]를 통해 명확히 XML 형식으로 변환가능하면 좀더 정확성 높은 분석을 진행할 수 있겠지만, bplist 구조 상 footer 나 end of file 을 정의할 수 없어서 아래와 같이 변환이 불가능하여 Rule 과 같이 Header 로부터 4KB 만큼 카빙한 데이터를 이용하여 그 안에서 유용한 정보를 이용하여 판단하였다.

위의 정보에 따르면 iOS version 의 경우는 iOS, iPhone OS 등과 같이 표현되고 있으며 12.4.8 을 쓰는 것을 확인할 수 있다. iPhone 의 경우는 7.2 라고 나오는데 이는 아이폰 7.2 를 나타내는 것이 아닌 model identifier 를 의미하는 것이다. 관련 identifier 를 정리해 놓은 Wiki page[5]를 통해 iPhone 7,2 는 iPhone 6 를 나타내고 있는 것을 파악할 수 있다.

또한 일부 bplist 의 경우에 knowledgeC.db 라는 데이터베이스 안에 Base64 형태로 Encode 되어 있는데[6] Dump 에서 String 만 추출한 것을 대상으로 확인해보면 “Deflate”라는 문자열을 통해 쉽게 Base64 Encoding 된 bplist 를 확인할 수 있다.

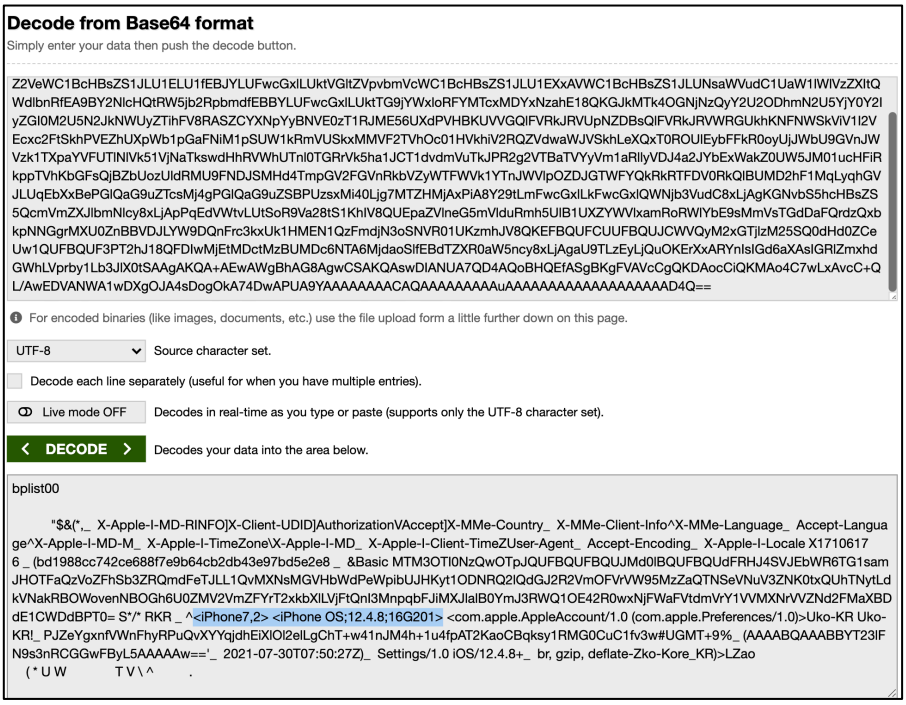
[그림 7] Base64 로 Encoding 된 bplist

위 Base64 Encoding된 파일을 Decode해 보면 아래와 같이 bplist00의 Header를 가진 bplist 구조임을 확인할 수 있다.

[4] https://www.theiphonewiki.com/wiki/Plutil

[5] https://www.theiphonewiki.com/wiki/Models

[6] https://abrignoni.blogspot.com/2019/03/ios-bplist-inception.html



[그림 8] Base64 Decoding 결과

해당 파일은 plutil로 변환하여 xml로 에러 없이 변환된다. 온전한 bplist 파일임을 확인할 수 있다. Xml로 변환된 내용은 아래와 같다.

[그림 9] bplist 로 복구하여 확인한 기기 모델 및 버전 정보

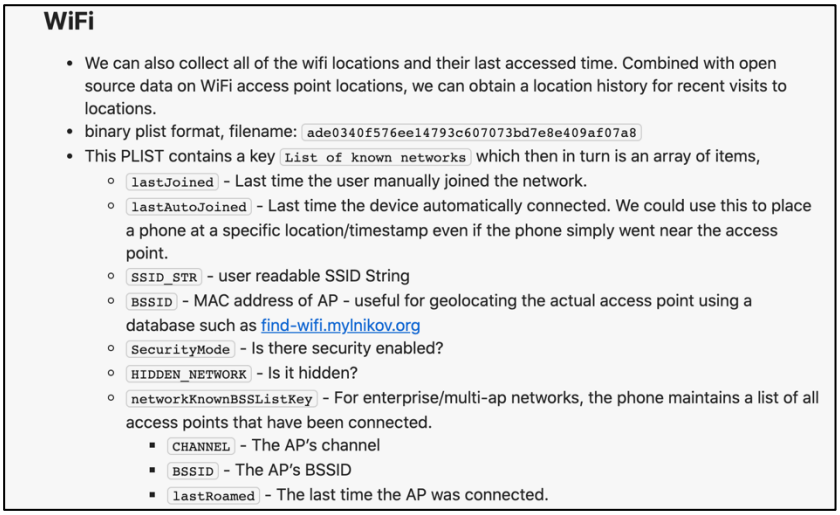
기 분석한 내용과 동일한 결과를 얻을 수 있었다.

[표 1] iPhone Device Model, iOS Version

Device Model	iPhone 6
iOS Version	12.4.8

3. Find WiFi names recently connected by analyzing the Settings. (at least 2 WiFi names) (50 points)

최근 연결한 WIFI 이름 역시 아래의 그림과 같이 binary plist 의 형태로 관리되고 있음을 확인할 수 있다.[7]



WiFi

- We can also collect all of the wifi locations and their last accessed time. Combined with open source data on WiFi access point locations, we can obtain a location history for recent visits to locations.
- binary plist format, filename: `ade0340f576ee14793c607073bd7e8e409af07a8`
- This PLIST contains a key `List of known networks` which then in turn is an array of items,
  - `lastJoined` - Last time the user manually joined the network.
  - `lastAutoJoined` - Last time the device automatically connected. We could use this to place a phone at a specific location/timestamp even if the phone simply went near the access point.
  - `SSID_STR` - user readable SSID String
  - `BSSID` - MAC address of AP - useful for geolocating the actual access point using a database such as find-wifi.mylnikov.org
  - `SecurityMode` - Is there security enabled?
  - `HIDDEN_NETWORK` - Is it hidden?
  - `networkKnownBSSListKey` - For enterprise/multi-ap networks, the phone maintains a list of all access points that have been connected.
    - `CHANNEL` - The AP's channel
    - `BSSID` - The AP's BSSID
    - `lastRoamed` - The last time the AP was connected.

[그림 10] WiFi 정보 관리

위 그림의 내용에 근거하면 plist 에는 최근 접근한 AP 에 대해 Auto 접속 여부(시간), Hidden Network 여부, SSID 이름 등을 저장하고 있는 것으로 확인할 수 있다. 이를 근거로 "lastAutoJoined", "lastJoined" 등 키워드를 이용하여 해당 키워드가 카빙된 bplist 중에 이러한 키워드가 존재하는 지 확인해보았다.

```
(base) sin90@odongbin-ui-MacBookPro bplist-2-1 % strings * |grep  lastAutoJoined
(base) sin90@odongbin-ui-MacBookPro bplist-2-1 % cd ..
(base) sin90@odongbin-ui-MacBookPro scalpel_output_settings % cd bplist-2-0
(base) sin90@odongbin-ui-MacBookPro bplist-2-0 % strings * |grep  lastAutoJoined
(base) sin90@odongbin-ui-MacBookPro bplist-2-0 % ls
```

[그림 11] "lastAutoJoined" 키워드 탐색

그러나 scalpel 을 통한 carving 결과에서는 존재하지 않아 dump 에서 해당 문자열이 존재하는 메모리 덤프 파일을 찾아 수동 carving 을 진행하였다.

키워드는 "lastAutoJoined"를 이용하였고, 10737418240_dump.data 가 매칭되어 해당 파일을 집중적으로 확인하였다.

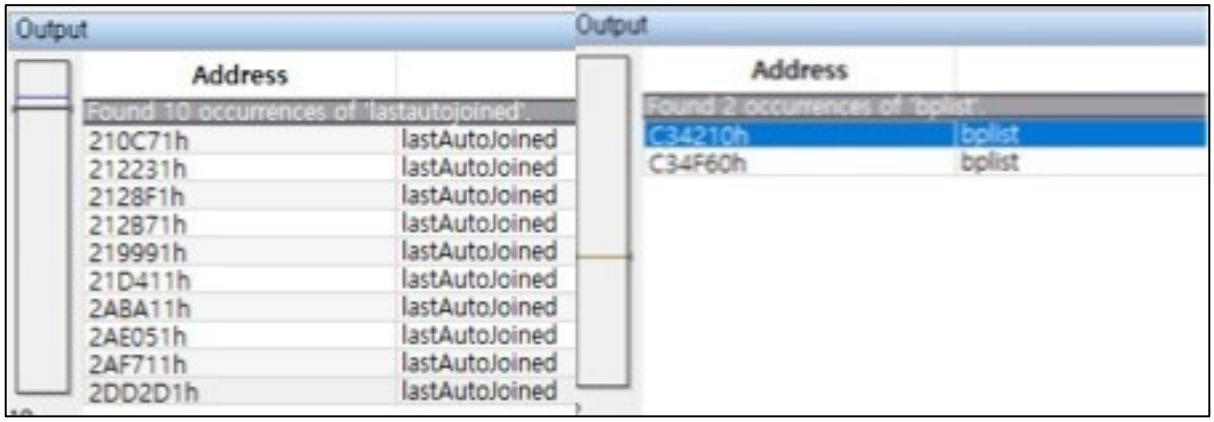
```
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cd dump
(base) sin90@odongbin-ui-MacBookPro dump % grep -n lastAutoJoined *
Binary file 10737418240_dump.data matches
(base) sin90@odongbin-ui-MacBookPro dump %
```

[그림 12] "lastAutoJoined" 매칭 Dump 탐색

해당 파일에서는 010editor 를 이용하여 bplist 의 header 가 존재하는 영역과 lastAutoJoined 주변에 존재하는 WiFi 이름 등을 확인하였다.

---

[7] https://www.richinfante.com/2017/3/16/reverse-engineering-the-ios-backup#wifi

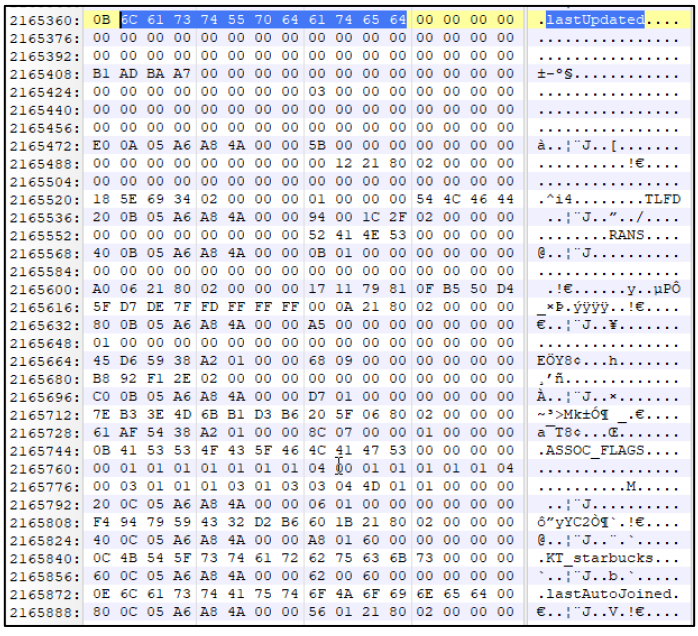


[그림 13] lastAutoJoined, bplist 존재 영역 탐색

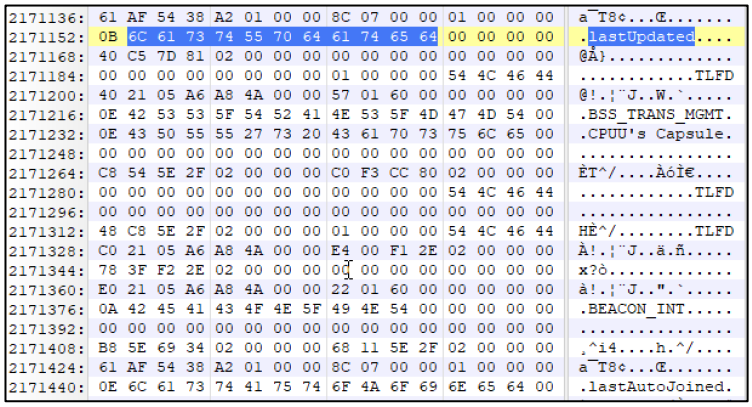
bplist Header 가 존재하는 영역과 WiFi 최근 연결내역과 관련되었다고 고려되는 영역과는 상당한 차이가 존재함을 확인할 수 있었다.

이러한 영역의 차이로 인해 bplist carving 시도 결과에 WiFi 최근 연결 내역이 포함되지 않았다는 것을 확인할 수 있다. 예상하기로 lastAutoJoined 영역에 존재 했어야 할 bplist 의 header 가 메모리 페이징으로 인해 overwrite 되었다고 추측할 수 있다.

이후에는 lastAutoJoined 근처에 존재하는 WiFi 이름을 Heuristic 하게 확인하였다. 확인 내용은 아래와 같다.



[그림 14] KT_starbucks



```
2171136: 61 AF 54 38 A2 01 00 00 8C 07 00 00 01 00 00 00  a¯T8¢...Œ.......
2171152: 0B 6C 61 73 74 55 70 64 61 74 65 64 00 00 00 00  .lastUpdated....
2171168: 40 C5 7D 81 02 00 00 00 00 00 00 00 00 00 00 00  @Å}.............
2171184: 00 00 00 00 00 00 00 00 01 00 00 00 54 4C 46 44  ............TLFD
2171200: 40 21 05 A6 A8 4A 00 00 57 01 60 00 00 00 00 00  @!.¦¨J..W.`.....
2171216: 0E 42 53 53 5F 54 52 41 4E 53 5F 4D 47 4D 54 00  .BSS_TRANS_MGMT.
2171232: 0E 43 50 55 55 27 73 20 43 61 70 73 75 6C 65 00  .CPUU's Capsule.
2171248: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2171264: C8 54 5E 2F 02 00 00 00 C0 F3 CC 80 02 00 00 00  ÈT^/....Àóì€....
2171280: 00 00 00 00 00 00 00 00 00 00 00 00 54 4C 46 44  ............TLFD
2171296: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2171312: 48 C8 5E 2F 02 00 00 00 01 00 00 00 54 4C 46 44  HÈ^/........TLFD
2171328: C0 21 05 A6 A8 4A 00 00 E4 00 F1 2E 02 00 00 00  À!.¦¨J..ä.ñ.....
2171344: 78 3F F2 2E 02 00 00 00 0C 00 00 00 00 00 00 00  x?ò.............
2171360: E0 21 05 A6 A8 4A 00 00 22 01 60 00 00 00 00 00  à!.¦¨J.."`.`.....
2171376: 0A 42 45 41 43 4F 4E 5F 49 4E 54 00 00 00 00 00  .BEACON_INT.....
2171392: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
2171408: B8 5E 69 34 02 00 00 00 68 11 5E 2F 02 00 00 00  ¸^i4....h.^/....
2171424: 61 AF 54 38 A2 01 00 00 8C 07 00 00 01 00 00 00  a¯T8¢...Œ.......
2171440: 0E 6C 61 73 74 41 75 74 6F 4A 6F 69 6E 65 64 00  .lastAutoJoined.
```

[그림 15] CPUU's Capsule

이외에 String 을 통해 아래와 같은 WiFi Name 을 확인하였으나, 최근 접속 여부를 알 수는 없다는 한계가 있었다. 기 저장된 AP 관련 정보일 수도 있겠다는 추측을 하였다.

```
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cat settings_dump.txt| grep -i "kt giga" | sort | uniq

KT GiGA WiFi
KT GiGA WiFi
KT GiGA WiFi
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cat settings_dump.txt| grep -i "kt wifi" | sort | uniq
KT WiFi
KT WiFi
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cat settings_dump.txt| grep -i "kt_starbucks" | sort | un

KT_starbucks
KT_starbucks
KT_starbucks_Secure
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cat settings_dump.txt| grep -i "kt_giga" | sort | uniq
KT_GiGA_5G_4120
KT_GiGA_5G_Wave2_397A
(base) sin90@odongbin-ui-MacBookPro 208-iOSFridump % cat settings_dump.txt| grep -i "capsule" | sort | uniq
CPUU's Capsule
```

[그림 16] 습득한 WiFi 명에 기반한 추가 데이터 탐색

[표 2] 최근 연결되었던 WiFi 이름

WiFi 이름 (1)	KT_starbucks
WiFi 이름 (2)	CPUU's Capsule

4. The Safari memory dump contains the web site addresses accessed by the user. Find 10 or more, but exclude duplicates. (100 points)

Safari 의 history 는 Library/Safari/History.plist 파일을 통하여 저장하는 것으로 알려져 있다.[8]

- *History.plist* is a binary plist file which contains the recently visited web pages. It is structured as an array of dictionaries, each of which represents a single entry. Each element has, among the other attributes, a title (corresponding to the title of the linked page), a URL, the last visited date and a visits counter.

[그림 17] History.plist 정보

해당 plist 파일은 Title 과 함께 URL, last visited date 를 함께 저장하는 것으로 확인되었다. 또한 Cookie 의 String Analysis 를 통해서도 visited sites 를 일부 식별할 수 있는 것으로 확인되었다.

- *Cookies.binarycookies*, a binary file which seems to contain cookies created by the Safari web browser. A simple string analysis using the *strings* Unix command revealed a list of visited sites with their associated cookie values (which could include, for example, session

[그림 18] Cookie 정보

iOS Version 12 에서는 SQLite 의 형태로 저장하기도 한다.[9] History 뿐만 아니라 Tab 정보까지 얻을 수 있다는 특징이 있다. 이러한 근거를 대상으로 scalpel 을 이용하여 Carving 된 plist(bplist)와 SQLite 를 대상으로 http 와 같은 History 주소의 필수요소를 기반으로 String 검색을 진행하였다. 매칭된 결과는 아래의 그림과 같다.

History. Safari browsing history is stored in the AppDomain-com.apple.mobilesafari/Library/Safari/History.db SQLite database. **Starting with iOS 12, deleted Safari history can no longer be recovered.**

Tabs. Starting with iOS 10, Safari open tabs are stored in the AppDomain-com.apple.mobilesafari/Library/Safari/BrowserState.db SQLite database. In iOS 10 and 11, open tabs would be stored indefinitely until closed, including tabs opened in the Private browsing sessions. However, even after the tabs have been closed, they could still be recovered. Starting with iOS 12, this is no longer the case. iOS 13 brings an additional protection mechanism, allowing users to specify the maximum period of time a tab can remain opened, automatically closing the tab and wiping the corresponding record after that period of time. **In iOS 12 and 13, information about closed Safari tabs can no longer be recovered.**

[그림 19] iOS 12 의 Safari History 저장 정보

[8] Piccinelli, Mario, and Paolo Gubian. "Exploring the iPhone backup made by iTunes." *Journal of Digital Forensics, Security and Law* 6.3 (2011): 4.
[9] https://blog.elcomsoft.com/2020/07/the-iphone-data-recovery-myth-what-you-can-and-cannot-recover/

```
(base) sin90@odongbin-ui-MacBookPro scalpel_output_safari % grep -rin http .
Binary file ./bplist-2-0/00000097.bplist matches
Binary file ./bplist-2-0/00000107.bplist matches
Binary file ./bplist-2-0/00000115.bplist matches
Binary file ./bplist-2-0/00000117.bplist matches
Binary file ./bplist-2-0/00000087.bplist matches
Binary file ./bplist-2-0/00000113.bplist matches
Binary file ./bplist-2-0/00000129.bplist matches
Binary file ./bplist-2-0/00000111.bplist matches
Binary file ./bplist-2-0/00000103.bplist matches
Binary file ./bplist-2-0/00000127.bplist matches
Binary file ./bplist-2-0/00000114.bplist matches
Binary file ./bplist-2-0/00000122.bplist matches
Binary file ./bplist-2-0/00000096.bplist matches
Binary file ./bplist-2-0/00000108.bplist matches
Binary file ./bplist-2-0/00000112.bplist matches
Binary file ./bplist-2-0/00000134.bplist matches
Binary file ./bplist-2-0/00000138.bplist matches
Binary file ./bplist-2-0/00000110.bplist matches
Binary file ./bplist-2-0/00000102.bplist matches
Binary file ./bplist-2-0/00000092.bplist matches
Binary file ./sqlitedb-3-0/00000152.sqlitedb matches
Binary file ./sqlitedb-3-0/00000153.sqlitedb matches
Binary file ./sqlitedb-3-0/00000151.sqlitedb matches
Binary file ./db-0-0/00000001.db matches
Binary file ./db-0-0/00000000.db matches
Binary file ./db-0-0/00000002.db matches
```

[그림 20] Carving 된 DB, SQLite 대상 http 키워드 탐색

DB 와 SQLite 에서는 아래와 같이 의미 없는 결과를 나타내었다.

```
(base) sin90@odongbin-ui-MacBookPro db-0-0 % strings * | grep http
/usr/lib/libapple_nghttp2.dylib
/usr/lib/libapple_nghttp2.dylib
http://www.apple.com
(base) sin90@odongbin-ui-MacBookPro db-0-0 % cd ../sqlitedb-3-0
(base) sin90@odongbin-ui-MacBookPro sqlitedb-3-0 % strings * | grep http
/usr/lib/libapple_nghttp2.dylib
/usr/lib/libapple_nghttp2.dylib
http://www.apple.com
(base) sin90@odongbin-ui-MacBookPro sqlitedb-3-0 %
```

[그림 21] plist, bplist 대상 http 키워드 탐색

plist 와 bplist 내에서도 http 와 관련된 문자열을 확인하였지만 실제 방문 여부를 파악할 수 없다는 한계가 존재한다.

```
(base) sin90@odongbin-ui-MacBookPro bplist-2-0 % strings * | grep http
ghttps://api.smoot.apple.com/static/installed_app_whitelist_url?version=77209e2e310a35da1cbea2e72583ce58_
https://api.smoot.apple.com/bag?agent=parsecd%2F1.0%20%28iPhone7%2C2%3B%20iPhone%20OS%2012.4.8%2016G201%29%20Parsecd%2F1.0&cc=KR&esl=ko&locale=ko-Kore_KR&tuscany=1_
rhttps://cdn.smoot.apple.com/static/spotlight_suggestions_phrase_dictionary/4/phrase-dictionary-en-US.map?version=4_
dhttps://cdn.smoot.apple.com/static/dictionary_resources_url?version=d4546b111e90a1d7005fa43d34bd2f7b_
qhttps://cdn.smoot.apple.com/static/static_corrections_dict/20200930/static_corrections_dict.json?version=20200930Rpb_
Zhttps://cdn.smoot.apple.com/static/spotlight_model_resources/2/spotlight_ml.mpak?version=2_
)https://fbs.smoot.apple.com/fb?ab_seed=42
1https://api-glb.smoot.apple.com/search?ab_seed=42
Ghttps://init.itunes.apple.com/WebObjects/MZInit.woa/wa/signSapSetupCert
6https://api-glb-apne1c.smoot.apple.com/warm?ab_seed=42#?
ghttps://cdn.smoot.apple.com/static/crowdsourcing_whitelist_url?version=032857ed01170d11e723650d446758
https://www.vertica.com/landing-page/start-your-free-trial-today/?utm_campaign=7014J000000dej3QAA&utm_funnel=action&utm_source=d&utm_medium=d&utm_campaignid=13450625514&utm_adgroupid=122906960866&utm_adid=530432504114&utm_placement=d&utm_agency=gupta&utm
_content=13450625514_122906960866_530432504114&gclid=EAIaIQobChMI9bPV8KmK8gIVUdR8Ch39wgYmEAEYASAAEgIim_D_BwE
IEC http://www.iec.ch
IEC http://www.iec.ch
ohttps://www.google.co.kr/search?q=%EB%82%98%EB%AC%B4%EC%9C%84%ED%82%A4&ie=UTF-8&oe=UTF-8&hl=ko-kr&client=safari_
ohttps://www.google.co.kr/search?q=%EB%82%98%EB%AC%B4%EC%9C%84%ED%82%A4&ie=UTF-8&oe=UTF-8&hl=ko-kr&client=safari
https://namu.wiki/
fhttps://configuration.apple.com/configurations/internetservices/cloudkit/bookmarks/bookmarks-3.0.plist#@N
https://is2-ssl.mzstatic.com/image/thumb/Purple115/v4/0a/05/34/0a0534d7-e6e2-dd5d-6fe7-ec8c9c4921b5/AppIcon-0-0-1x_U007emarketing-0-0-0-7-0-0-sRGB-0-0-0-GLES2_U002c0-512MB-85-220-0-0.png/128x128bb.png#A
https://is2-ssl.mzstatic.com/image/thumb/Purple115/v4/0a/05/34/0a0534d7-e6e2-dd5d-6fe7-ec8c9c4921b5/AppIcon-0-0-1x_U007emarketing-0-0-0-7-0-0-sRGB-0-0-0-GLES2_U002c0-512MB-85-220-0-0.png/128x128bb.png#A
Bhttps://init.itunes.apple.com/bag.xml?ix=6&os=12&locale=ko-Kore_KR#A
https://client-api.itunes.apple.com/WebObjects/MZStorePlatform.woa/wa/lookup?caller=com.apple.mobilesafari&id=808794344&p=lockup&version=1#A
https://client-api.itunes.apple.com/WebObjects/MZStorePlatform.woa/wa/lookup?caller=com.apple.mobilesafari&id=808794344&p=lockup&version=1#A
https://client-api.itunes.apple.com/WebObjects/MZStorePlatform.woa/wa/lookup?caller=com.apple.mobilesafari&id=808794344&p=lockup&version=1#A
https://client-api.itunes.apple.com/WebObjects/MZStorePlatform.woa/wa/lookup?caller=com.apple.mobilesafari&id=808794344&p=lockup&version=1#A
https://client-api.itunes.apple.com/WebObjects/MZStorePlatform.woa/wa/lookup?caller=com.apple.mobilesafari&id=808794344&p=lockup&version=1#A
https://xp.apple.com/register#A
Bhttps://init.itunes.apple.com/bag.xml?ix=6&os=12&locale=ko-Kore_KR#A
&http://certs.apple.com/apevsrsa2g1.der06
*http://ocsp.apple.com/ocsp03-apevsrsa2g1010
2https://www.apple.com/certificateauthority/public/0
Reliance on this certificate by any party assumes acceptance of the Relying Party Agreement found at https://www.apple.com/certificateauthority/public/.0
$http://crl.apple.com/apevsrsa2g1.crl0
+http://certs.apple.com/appleistca2g1_bc.cer
http://olympics.com/tokyo-2020
Khttps://p7-buy.itunes.apple.com/WebObjects/MZFinance.woa/wa/sagaAddComputer
```

[그림 22] History plist 추정 지점 탐색

3 번 문항에서 사용한 "lastJoined" 라는 키워드와 달리 Safari Web history 의 plist 구조에는 명확히 사용할 수 있는 키워드가 존재하지 않는다. 이에 Heuristic 을 이용하여 Memory Dump 내에 Safari History 와 관련된 plist 라고 추정 가능한 곳을 탐색 시도하였다.

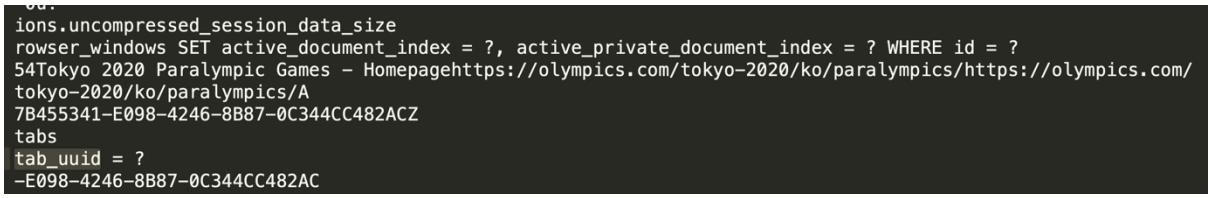
- **Tab 관련 정보**

tab_uuid 라는 키워드를 통해 검색한 결과 open 되어 있던 tab 에 대한 정보를 획득할 수 있었고 그 중 방문한 URL 까지 추출할 수 있었다. 2 건 확인 가능하였다.

해당 키워드는 CloudTabs.db 이라는 아티팩트에서 기반한 것으로, 클라우드 상에서의 탭 동기화 기능과 연관되어 있다고 추측할 수 있으며 이미 관련된 포렌식 수집도구 또한 존재한다.[10]

browser_window_uuid 의 경우에도 마찬가지로 브라우저의 "새 창"과 관련된 기능으로

각 Window 를 관리하기 위해 데이터를 일부 저장하는 것으로 추측할 수 있다.

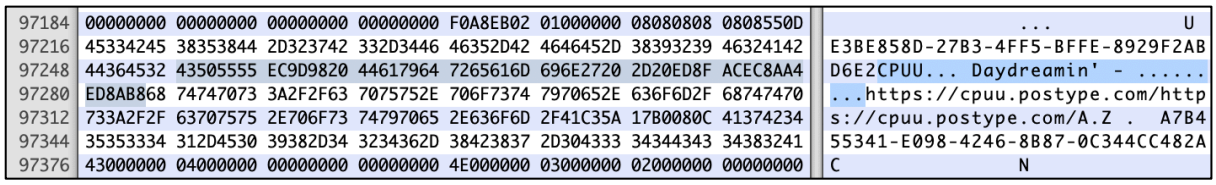


[그림 23] Tokyo 2020 올림픽 관련 URL

[표 3] tab_uuid 키워드를 통해 확보한 방문 정보

URL	https://olympics.com/tokyo-2020/ko/paralympics/
Title	Tokyo 2020 Paralympic Games - Homepage

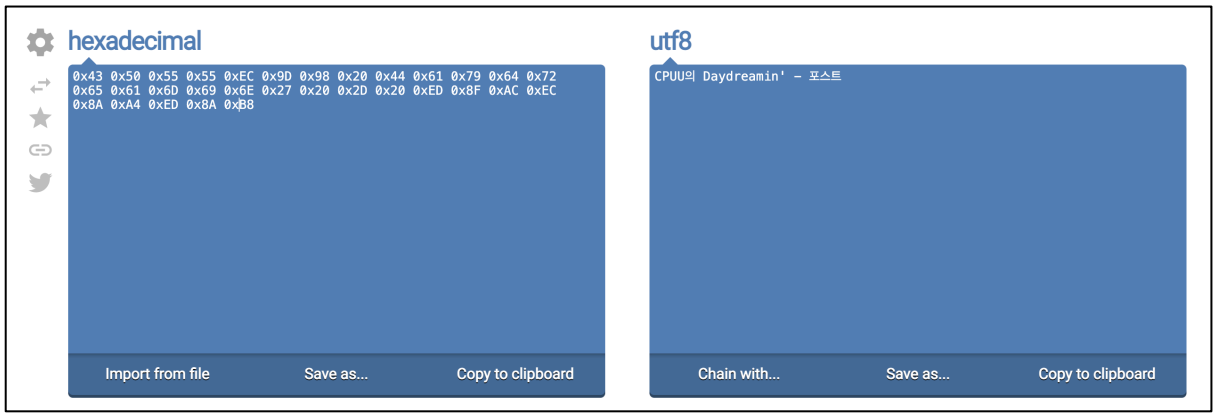
[그림 24] 블로그 관련 URL



[그림 25] Raw Data Tracking

해당 데이터의 경우 윈도우 명이 유니코드 데이터로 저장되어 있어 이를 변환하는 과정을 아래와 같이 수행하였으며, 개인 블로그 관련 방문 정보를 획득할 수 있었다.

10 https://github.com/ydkhatri/mac_apt/blob/master/plugins/safari.py



[그림 26] UTF-8 변환 과정

[표 4] browser_window_uuid 를 통해 확보한 방문 정보

URL	https://cpuu.postype.com/
Title	CPUU 의 Daydreamin' - 포스트

- History 관련 정보 (근처 string 기준 제목과 함께 나열)

```
WebProcessProxy
UpdateBackForwardItem
Digital Investigation - Journal - Elsevier
https://www.journals.elsevier.com/digital-investigation6
http://www.journals.elsevier.com/digital-investigation
about:blank
about:blank
<!--frame1-->
```

[그림 27] Digital Investigion Journal URL

[표 5] History 키워드를 통해 확보한 방문 정보 (1)

URL	https://www.journals.elsevier.com/digital-investigation
Title	Digital Investigation - Journal - Elsevier

```
https://github.com/pwn20wndstuff/Undecimus
title
GitHub - pwn20wndstuff/Undecimus: unc0ver jailbreak for iOS 11.0 - 12.4
container
#js-repo-pjax-container
fragment
https://github.com/pwn20wndstuff/Undecimus/releases/tag/v5.2.0
title
Release v5.2.0 Release
 pwn20wndstuff/Undecimus
```

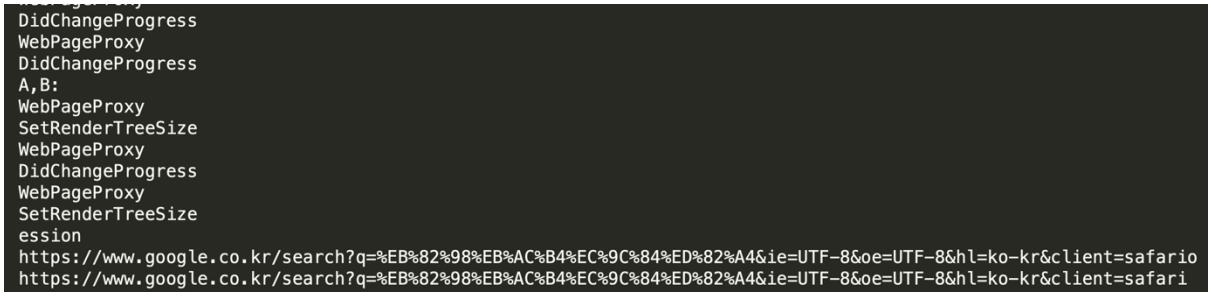
[그림 28] Github Repository URL

[표 6] History 키워드를 통해 확보한 방문 정보 (2)

URL	https://github.com/pwn20wndstuff/Undecimus https://github.com/pwn20wndstuff/Undecimus/releases/tag/v5.2.0
Title	GitHub - pwn20wndstuff/Undecimus: unc0ver jailbreak for iOS 11.0 - 12.4 Release v5.2.0 Release

- Cookie 관련 정보

Session, Render 등의 키워드로 Cookie 관련 정보임을 추측할 수 있었다.



[그림 29] Google 검색 관련 URL

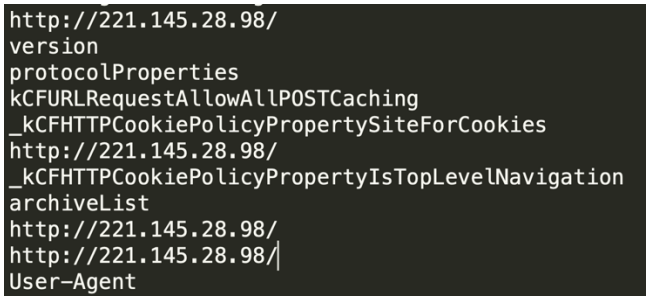
[표 7] Cookie 키워드를 통해 확보한 방문 정보 (1)

URL	https://www.google.co.kr/search?q=%EB%82%98%EB%AC%B4%EC%9C%84%ED%82%A4&ie=UTF-8&oe=UTF-8&hl=ko-kr&client=safari

[그림 30] The White House 관련 URL

[표 8] Cookie 키워드를 통해 확보한 방문 정보 (2)

URL	http://www.whitehouse.gov/



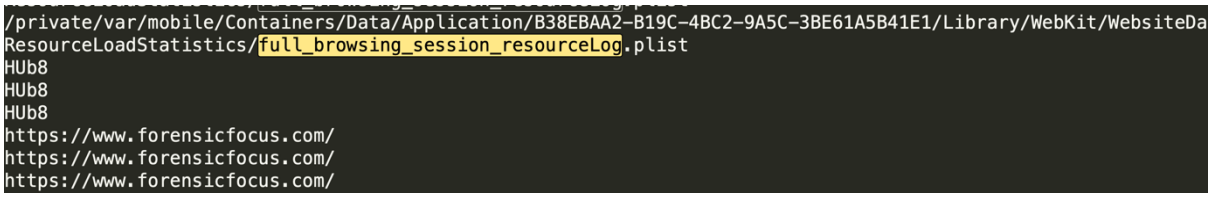
[그림 31] IP 로 구성된 URL

[표 9] Cookie 키워드를 통해 확보한 방문 정보 (2)

URL	http://221.145.28.98/

- **기타 정보**

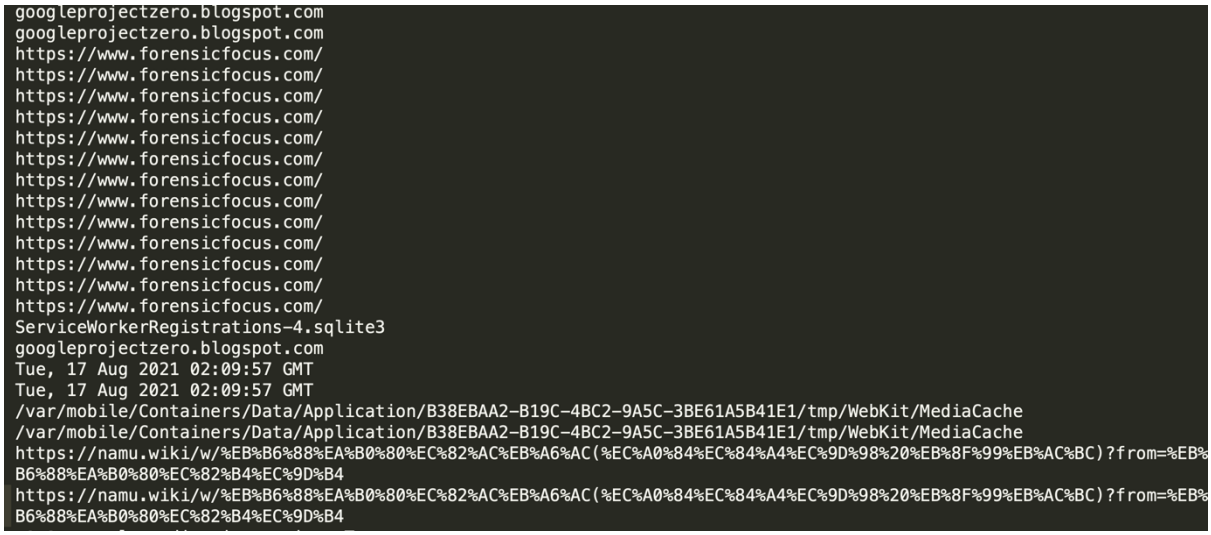
String search 를 통해 full_browsing_session_resourceLog.plist 라는 문자열을 포함하여, ServiceWorkerRegistrations-4.sqlite3 등 근처 문자열 확인을 통해 접속했을 것이라고 추측되는 일부 문자열을 추가로 발견할 수 있었다.



[그림 32] forensicfocus 관련 URL

[표 10] 키워드를 통해 수집한 URL 정보 (1)

URL	https://www.forensicfocus.com/



[그림 33] Google Project Zero 블로그 관련 URL

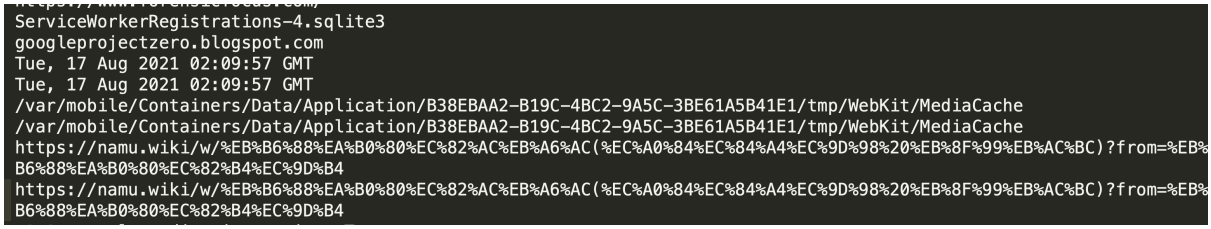
[표 11] 키워드를 통해 수집한 URL 정보 (2)

URL	googleprojectzero.blogspot.com/

[그림 34] 기타 URL

[표 12] 키워드를 통해 수집한 URL 정보 (3)

URL	https://tc.gts3.org/cs6265/2020/cal.html https://m.news.nate.com/view/20210625n35660 https://cs155.stanford.edu/syllabus.html



[그림 35] 나무위키 관련 URL

[표 13] 키워드를 통해 수집한 URL 정보 (4)

URL	https://namu.wiki/w/%EB%B6%88%EA%B0%80%EC%82%AC%EB%A6%AC(%EC%A0%84%EC%84%A4%EC%9D%98%20%EB%8F%99%EB%AC%BC)?from=%EB%B6%88%EA%B0%80%EC%82%B4%EC%9D%B4

[그림 36] NSR 채용 사이트 URL

[표 14] 키워드를 통해 수집한 URL 정보 (5)

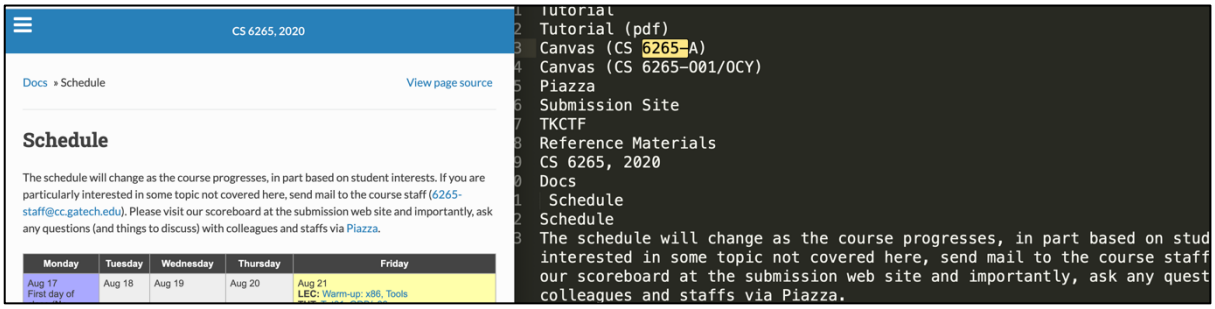
URL	https://nsr.recruiter.co.kr/appsite/company/index

위의 일련의 정보를 모두 합하여 사용자가 방문했을 거라고 예상되는 페이지는 다음과 같다.

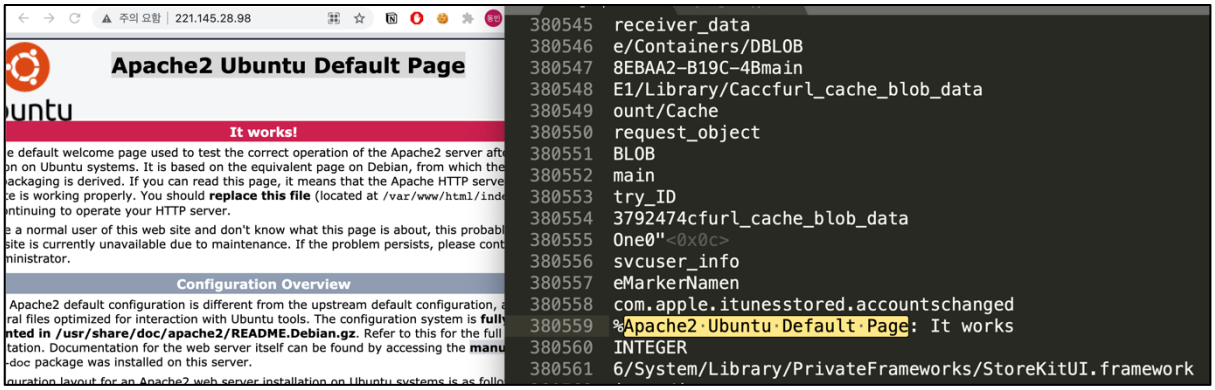
[표 15] 사용자 방문 추정 페이지 목록

URL	
	① https://nsr.recruiter.co.kr/appsite/company/index
	② https://namu.wiki/w/%EB%B6%88%EA%B0%80%EC%82%AC%EB%A6%AC(%EC%A0%84%EC%84%A4%EC%9D%98%20%EB%8F%99%EB%AC%BC)?from=%EB%B6%88%EA%B0%80%EC%82%B4%EC%9D%B4
	③ https://tc.gts3.org/cs6265/2020/cal.html
	④ https://m.news.nate.com/view/20210625n35660
	⑤ https://cs155.stanford.edu/syllabus.html
	⑥ googleprojectzero.blogspot.com/
	⑦ http://221.145.28.98/
	⑧ https://www.forensicfocus.com/
	⑨ https://www.google.co.kr/search?q=%EB%82%98%EB%AC%B4%EC%9C%84%ED%82%A4&ie=UTF-8&oe=UTF-8&hl=ko-kr&client=safari
	⑩ https://github.com/pwn20wndstuff/Undecimus
	⑪ https://github.com/pwn20wndstuff/Undecimus/releases/tag/v5.2.0
	⑫ https://www.journals.elsevier.com/digital-investigation
	⑬ https://cpuu.postype.com/
	⑭ https://olympics.com/tokyo-2020/ko/paralympics/

Tab 정보나 History 정보와 같이 구조 상 해당 페이지 접근 후 관련 웹 페이지 제목이나 기록이 존재할 수도 있지만, 메모리에 실제 웹페이지 관련 내용이 있는지와 같은 추가적인 검증을 통해 선별한 URL 이 타당성을 부여하였다. 또는 History 관련 아티팩트가 한 곳 뿐 아니라 여러 군데 발견되는지 여부를 추가 기준으로 하여 검증을 진행하였다.



[그림 37] https://tc.gts3.org/cs6265/2020/cal.html 비교



[그림 38] http://221.145.28.98/ 비교

[그림 39] http://googleprojectzero.blogspot.com/, http://cs155.stanford.edu/syllabus.html, https://cpuu.postype.com/, https://www.journals.elsevier.com/digital-investigation, http://www.whitehouse.gov/, https://olympics.com/tokyo-2020/ko/paralympics/ 교차 검증

이외에도 browserstate.db 라는 키워드를 통해 아래의 URL 을 추가로 발견할 수 있었다. 해당 파일은 Identifying private internet browsing activity in Mobile Safari[11] 에 사용되어 사용자가 접근한 URL 기록과 밀접한 연관을 갖는다. 그러나 실제로 유저가 접근했는지 확인은 되지않는다.

[표 16] 사용자 방문과 연관이 있을 것이라 추정되는 추가 확인 URL

URL	
	1. https://googleprojectzero.blogspot.com/?m=1o/
	2. https://m.news.nate.com/view/20210509n11087
	3. https://m.news.nate.com/view/20210509n07170Z/
	4. https://m.news.nate.com/view/20210509n08416Y/
	5. https://m.news.nate.com/view/20210509n03354X/
	6. https://m.news.nate.com/view/20210509n03184W/
	7. https://m.news.nate.com/view/20210509n10775V/
	8. https://m.news.nate.com/view/20210509n00018UC
	9. https://m.pann.nate.com/talk/359582291?&currMenu=talker&page=1S
	10. https://m.comm.news.nate.com/Comment/ArticleComment/ReplyList?mid=m03&artc_sq=20210509n08734&cmt_sq=220760819&beplefl=YQv
	11. https://m.mt.co.kr/renew/view.html?no=2021021006381670238&EMBA&type=outlink&ref=https%3A%2F%2Fm.news.nate.com
	12. https://m.blog.naver.com/PostView.nhn?blogId=lovetheme84&logNo=220512021544&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F
	13. https://m.news.nate.com/view/20210210n04960
	14. http://m.nate.com
	15. https://m.news.nate.com/view/20210625n35660
	16. https://m.naver.com
	17. https://m.post.naver.com/viewer/postView.nhn?volumeNo=19230233&memberNo=375594
	18. https://m.search.naver.com/search.naver?sm=mtp_sug.top&where=m&query=%ED%8F%AC%EC%BB%A4%EC%8A%A4+%EC%9A%B0%EC%8B%B8%EB%AF%B8&acq=%ED%8F%AC%EC%BB%A4%EC%8A%A4&acr=2&qdt=0A4

11 http://docplayer.net/196800552-White-paper-examining-mobile-devices-identifying-private-internet-browsing-activity-in-mobile-safari.html

19. https://m.blog.naver.com/gksmf42366/2218044331148
20. https://m.news.nate.com/view/20210509n13738O/
21. https://m.news.nate.com/view/20210509n03276N
22. https://m.search.naver.com/search.naver?sm=mtp_sly.hst&where=m&query=%ED%8F%AC%EC%BB%A4%EC%8A%A4+%EC%9A%B0%EC%8B%B8%EB%AF%B8&acr=1E
23. https://m.daum.net/?nil_top=mobilecC
24. https://m.news.nate.com/hissue/list?mid=e00&isq=7377#cid883969
25. https://m.youtube.com/results?search_query=%EC%95%84%EC%B9%A8+%ED%81%B4%EB%9E%98%EC%8B%9D
26. https://m.youtube.com/results?search_query=%EC%8B%A0%EC%83%9D%EC%95%84+%EB%B0%B1%EC%83%89%EC%86%8C%EC%9D%8C
27. https://github.com/pwn20wndstuffd
28. https://m.news.nate.com/view/20210625n35321?issue_sq=7377
29. https://accounts.google.com/signin/v2/identifier?uilel=3&continue=https%3A%2F%2Fm.youtube.com%2Fsignin%3Fnext%3D%252Fwatch%253Fv%253D69uu9As3Dt4%26hl%3Dko%26noapp%3D1%26app%3Dm%26action_handle_signin%3Dtrue&passive=true&hl=ko&service=youtube&ltmpl=mobile&flowName=GlifWebSignIn&flowEntry=ServiceLogin
30. https://m.youtube.com/channel/UCxg5eEU-VK40hm5X4zkilKg
31. https://m.youtube.com/watch?v=69uu9As3Dt4
32. http://github.com/pwn20wndstuffeg
33. https://m.youtube.com/results?search_query=%EC%8B%A0%EC%83%9D%EC%95%84+%EC%9E%90%EC%9E%A5%EA%B0%804-
34. https://m.youtube.com/watch?v=2gzgtOqVdpg
35. https://m.youtube.com/results?search_query=%23%EC%98%A4%EB%A5%B4%EA%B3%A8%EC%97%B0%EC%A3%BC%EA%B3%A1%EC%9E%90%EC%9E%A5%EA%B0%80
36. https://github.com/pwn20wndstuff/Undecimus/releases/tag/v5.2.0g
37. https://github.com/pwn20wndstuff/Undecimusf
38. https://m.news.nate.com/view/20210625n35289?issue_sq=7377ap

39. https://m.youtube.com/results?search_query=%EC%8B%A0%EC%83%9D%EC%95%84+%EB%B0%B1%EC%83%89%EC%86%8C%EC%9D%8C
40. https://m.youtube.com/watch?v=69uu9As3Dt43
41. https://m.news.nate.com/view/20210509n02025R
42. https://m.news.nate.com/view/20210509n08734P
43. https://m.youtube.com
44. https://m.search.naver.com/search.naver?where=m_video&sm=mtb_jum&query=%EA%B9%80%EA%B1%B4%EB%AA%A8+%ED%94%BC%EC%95%84%EB%85%B8G
45. https://m.search.naver.com/search.naver?sm=mtb_hty.top&where=m&oquery=%ED%8F%AC%EC%BB%A4%EC%8A%A4+%EC%9A%B0%EC%8B%B8%EB%AF%B8&tqi=hsTwdsprv2Nsscr0%2BRdsssssspd-196477&query=%EA%B9%80%EA%B1%B4%EB%AA%A8+%ED%94%BC%EC%95%84%EB%85%B8F
46. http://tweak-box.com/chimerai/
47. https://m.news.nate.com/view/20210509n12357TN
48. https://iextras.dpdcart.com/cart/view?product_id=167851&method_id=198300#
49. http://www.forensicfocus.com
50. http://cpuu.postype.com
51. http://unc0ver.dev