

203 – Where did the data go?

Team Information

Team Name : LuckyVicky

Team Member : Eungchang Lee, Hyun Yi, Juho Heo, Dongkyu Lee

Email Address : dfc_luckyvicky@googlegroups.com

Instructions

Description The file is selectively collected data from Logan's PC. Analyze the data to find clues about the leaked files.

Target	Hash (MD5)
Artifacts.zip	79bd1bea60d5a8ef879cc04c33eea5f8

Questions

1. Logan leaked a file to Google Drive. What is Logan's Google account name? (20 points)
2. Logan leaked the source code to Github. What is Logan's Github account email address? (20 points)
3. Logan leaked files to Dropbox. What files did Logan upload to Dropbox? (30 points)
4. Logan downloaded a file from M365. Analyze Logan's M365 email account name and the file he downloaded from M365. (40 points)
5. Logan leaked files via Slack. Analyze the following: (60 points)
 - A. Name of the workspace used for the leak
 - B. Name of the channel used for the leak
 - C. Logan's Slack account email
 - D. The Slack account email of the person he leaked to

E. Files leaked

6. Logan uploaded a file to the temp.sh site. Which file did he upload?
(30 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:	DB Browser for SQLite	Publisher:	sqlitebrowser
Version:	3.12.2		
URL:	https://sqlitebrowser.org/		

Name:	ChromeCacheView	Publisher:	Nirsoft
Version:	2.21		
URL:	https://www.nirsoft.net/		

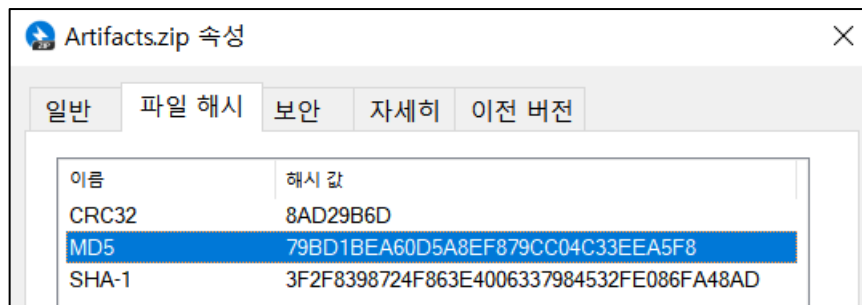
Name:	HashTab	Publisher:	Implbits Software
Version:	6.0.0		
URL:	https://implbits.com		

Name:	hindsight	Publisher:	obsidianforensics
Version:	v2023.03		
URL:	https://github.com/obsidianforensics/hindsight		

Name:	Slack-Parser	Publisher:	0xHasanM
Version:	-		
URL:	https://github.com/0xHasanM/Slack-Parser		

Name:	HxD	Publisher:	Maël Hörz
Version:	2.5.0.0		
URL:	https://www.mh-nexus.de		

Step-by-step methodology:



[그림 1] 주어진 Target 파일의 md5 해시 값 확인

주어진 target파일인 Artifacts.zip 파일의 md5해시가 일치함을 확인하였습니다.

1. Logan leaked a file to Google Drive. What is Logan's Google account name? (20 points)

Hindsight라는 웹 브라우저 아티팩트 포렌식 도구를 통해 주어진 target 파일 내 Google Default 폴더 내 아티팩트들의 분석을 진행하여 xlsx 파일로 추출하였습니다.

url	2024-07-18 09:15:03.889	https://app.slack.co	로그인 Slack	
url	2024-07-18 09:15:05.095	https://app.slack.co	로그인 Slack	
url	2024-07-18 09:15:05.159	https://app.slack.co	로그인 Slack	
autofill	2024-07-18 09:15:24.000			email loganlee9124@gmail.com
url	2024-07-18 09:15:25.394	https://app.slack.co	로그인 Slack	

[그림 2] gmail 계정 확인

Logan의 PC에서 선별 수집한 데이터 중 Chrome 웹 아티팩트에서 loganlee9124@gmail.com이라는 이메일을 추출한 xlsx파일에서 확인하였습니다.

테이블(T): autofill						
	name	value	value_lower	date_created	date_last_used	count
	필터	필터	필터	필터	필터	필터
1	email	loganlee9124@gmail.com	loganlee9124@gmail.com	1721294124	1721294124	1
2	identifier	loganlee9124@gmail.com	loganlee9124@gmail.com	1721297554	1721297554	1
3	identifier	namim7482@gmail.com	namim7482@gmail.com	1721297788	1721297788	1
4	susi_email	namim	namim	1721303789	1721303789	1
5	login	namim7482@gmail.com	namim7482@gmail.com	1721306222	1721306222	1

[그림 3] Web Data의 autofill 테이블 내 계정 확인

해당 이메일은 Default 폴더 내 Web Data Sqlite3 db 파일의 autofill 테이블에서 식별 가능하였습니다.

답 : loganlee9124@gmail.com

2. Logan leaked the source code to Github. What is Logan's Github account email address? (20 points)

Logan의 Github 계정 이메일 주소는 앞서 살펴본 [그림 3] 내의 또 다른 이메일 주소를 통해, 다른 아티팩트를 살펴보았습니다.

```
C: > Users > ehfeh > Desktop > 2024dfc > 203 - Where did the data go > Artifacts > C > Users > user > .gitconfig
1  [core]
2  |   editor = \"C:\\Users\\user\\AppData\\Local\\Programs\\Microsoft VS Code\\bin\\code\" --wait
3  [user]
4  |   name = Logan Lee
5  |   email = namim7482@gmail.com
```

[그림 4] .gitconfig 파일 내용 확인

주어진 Artifacts 파일의 C:\Users\user\.gitconfig 파일에서 이름과 이메일의 정보를 통해 namim7482@gmail.com이 Logan의 Github 계정 이메일 주소라고 판단하였습니다.

답 : namim7482@gmail.com

3. Logan leaked files to Dropbox. What files did Logan upload to Dropbox? (30 points)

sync_history							
모든 열에서 필터링							
nt_type	direction	file_id	local_path	server_path	other_user	timestamp	
	필터	필터	필터	필터	필터	필터	필터
	upload	I21iz5zSWo0AAAAAAGQ	C:\Users\User\Dropbox\All-In-One-Python-Projects.zip	3288824595/All-In-One-Python-Projects.zip	0	1721304061	
	upload	I21iz5zSWo0AAAAAAGg	C:\Users\User\Dropbox\kodocs.io.zip	3288824595/kodocs.io.zip	0	1721304106	
	upload	I21iz5zSWo0AAAAAAGw	C:\Users\User\Dropbox\Python-Speech-Recognition.zip	3288824595/Python-Speech-Recognition.zip	0	1721304193	

[그림 5] sync_history.db 파일

Artifacts 폴더 내 C:\Users\user\AppData\Local\Dropbox\instance1 경로에서 sync_history.db 파일을 DB Browser for SQLite로 열어보았을때, 위 그림과 같이 세 개의 파일을 확인할 수 있습니다. 세 개의 파일들은 모두 direction column이 upload로 기록된 server_path에 timestamp에 적힌 시간으로 업로드 되었음을 알 수 있습니다.

Logan이 Dropbox에 유출한 파일	
C:\Users\user\Dropbox\All-In-One-Python-Projects.zip	
C:\Users\user\Dropbox\kodocs.io.zip	
C:\Users\user\Dropbox\Python-Speech-Recognition.zip	

4. Logan downloaded a file from M365. Analyze Logan's M365 email account name and the file he downloaded from M365. (40 points)

Hindsight Internet History Forensics (v2023.03)				
Type	Timestamp (UTC)	URL	Title / Name / Status	Data / Value / Path
site setting (modified)	2024-07-18 09:54:45.371	https://www.office.com/media_engagement	[in Preferences.profile]	{'expiration': '13373546085371848', 'last_modified': '13373546085371848'}
url	2024-07-18 10:06:55.619	https://byternd-my.	내 파일 - OneDrive	
url	2024-07-18 10:06:57.638	https://byternd-my.	Document - OneDrive	
download	2024-07-18 10:07:25.429	https://koreacentral.Complete - 100% [1076882/1076882]		C:\Users\User\Downloads\OneDrive_2024-07-18.zip

[그림 6] download 기록 확인

앞서 살펴본 hindsight를 통한 chrome 웹 아티팩트에서 download기록을 확인할 수 있습니다.

테이블(T): downloads		
id	guid	current_path
...	필터	필터
1	1 decdf14e-f5bf-41bb-918f-352ff36ae9ae	C:\Users\User\Downloads\OneDrive_2024-07-18.zip
2	2 6e7cd81b-55ba-4552-9de7-1db3b857...	C:\Users\User\Downloads\DropboxInstaller.exe

[그림 7] History 파일의 downloads 테이블

History 파일에서 downloads 테이블을 살펴보면 Logan이 크롬 브라우저를 통해 OneDrive_2024-07-18.zip을 다운로드 받은 것을 알 수 있습니다.

URL	Interpretation
https://portal.office.com/onedrive?msafed=0&wsucxt=2&username=leeaustin%40BYTERND.onmicrosoft.com	msafed: 0 wsucxt: 2 username: leeaustin@BYTERND.onmicrosoft.com [Query String Parser]
https://portal.office.com/login?login_hint=leeaustin%40BYTERND.onmicrosoft.com&ru=%2Fonedrive%3Fmsafed%3D0%26wsucxt%3D2%26username%3Dleeaustin%2540BYTERND.onmicrosoft.com	login_hint: leeaustin@BYTERND.onmicrosoft.com ru: /onedrive?msafed=0&wsucxt=2&username=leeaustin%40BYTERND.onmicrosoft.com [Query String Parser]

[표 1] Logan의 로그인 시도 기록 in Chrome Web Artifact

그리고 OneDrive_2024-07-18.zip을 다운로드 받기 전의 기록을 살펴보면, Logan의 로그인 시도 기록을 발견할 수 있었습니다. 이를 통해 Logan의 M365 이메일 계정 이름은 leeaustin@BYTENRND.onmicrosoft.com임을 알 수 있습니다.

답 : leeaustin@BYTENRND.onmicrosoft.com, OneDrive_2024-07-18.zip

5. Logan leaked files via Slack. Analyze the following: (60 points)

Slack을 통해 유출된 정보들을 찾기 위해 다음의 경로에 있는 아티팩트를 Slack-Parser라는 도구와 HxD 도구를 활용하여 분석하였습니다.

[path]

C:\Users\user\AppData\Roaming\Slack\IndexedDB\https_app.slack.com_0.indexeddb.blob

A. Name of the workspace used for the leak

```
Insert the data you want (users, messages, workspace): workspace
domain: myworkspace-yhp4917
channels: ['secretconversation']
```

[그림 8] slack 아티팩트에 기록된 workspace

000015D0	7B 22 74 65 61 6D 73 22 3A 7B 22 54 30 37 44 43	{"teams":{"T07DC
000015E0	4A 57 44 42 33 33 22 3A 7B 22 69 64 22 3A 22 54	JWDB33":{"id":"T
000015F0	30 37 44 43 4A 57 44 42 33 33 22 2C 22 6E 61 6D	07DCJWDB33","nam
00001600	65 22 3A 22 4D 79 20 57 6F 72 6B 73 70 61 63 65	e":"My Workspace
00001610	22 2C 22 75 72 6C 22 3A 22 68 74 74 70 73 3A 2F	","url":"https:/
00001620	2F 6D 79 77 6F 72 6B 73 70 61 63 65 2D 79 68 70	/myworkspace-yhp
00001630	34 39 31 37 2E 73 6C 61 63 6B 2E 63 6F 6D 2F 22	4917.slack.com/"
00001640	2C 22 64 6F 6D 61 69 6E 22 3A 22 6D 79 77 6F 72	,"domain":"mywor
00001650	6B 73 70 61 63 65 2D 79 68 70 34 39 31 37 22 2C	kspace-yhp4917",

[그림 9] Local Storage\leveldb 내 파일에 기록된 workspace

Slack 아티팩트에 기록된 workspace는 위 두 그림과 같이 IndexedDB와 Local Storage 등의 폴더에서 **myworkspace-yhp4917**로 확인됩니다.

답 : myworkspace-yhp4917

B. Name of the channel used for the leak

유출 채널의 이름의 경우 A에서 workspace를 확인하면서 함께 **secretconversation**으로 확인할 수 있었습니다.

답 : **secretconversation**

C. Logan's Slack account email

Users	Profile Picture	Email	is_bot	is_admin	is_owner	primary_owner
loganlee9124		loganlee9124@gmail.com	False	True	True	True
raymondhong61		https://ca.slack-edge.com/T07DCJWDB33-U0702DJ6T3N-g359a-9675e8-102	False	False	False	False
Slackbot		https://ca.slack-edge.com/T07DCJWDB33-U07CZUYP0S-g422fd27d8c7-102	False	False	False	False
		404 Not Found	False	False	False	False
		https://ca.slack-edge.com/T07DCJWDB33-USLACKBOT-sv41d8cd98f0-102	False	False	False	False

[그림 10] Slack 아티팩트 내 저장된 user 기록

Slack에 저장된 user 중, Logan의 슬랙 이메일 계정은 위 그림과 같이 **loganlee9124@gmail.com**으로 확인할 수 있습니다.

답 : **loganlee9124@gmail.com**

D. The Slack account email of the person he leaked to

```
Hi hong{${}${$
Hi hong{@${}${$
Hi Lee{${}${$
Hi Lee{@${}${$
,It's a list of source codes I have access to{${}${$
,It's a list of source codes I have access to{@${}${$
Okay{${}${$
Okay{@${}${$
Send 'All-In-One-Projects' code{${}${$
Send 'All-In-One-Projects' code{@${}${$
```

[그림 11] Slack 아티팩트 내 기록된 대화 내역

Slack에 저장된 대화 내역을 살펴보면 Logan이 hong이라는 사람과 대화를 나눈 기록을 확인할 수 있습니다. 그리고, C에서 확인한 user 내역을 통해 Logan이 파일을 hong이라는 사람에게 유출했다는 사실을 알 수 있고, 그의 이메일 계정은 **raymondhong61@gmail.com**임을 알 수 있습니다.

답 : **raymondhong61@gmail.com**

E. Files leaked

그리고 앞서 D에서 살펴본 대화 내역에서, 맥락 상 Logan이 Send 'All-In-One-Projects' code라는 말을 통해 파일을 보내겠다는 말을 남겼습니다.

```
"id" "F07D04VFM0A"
is_tombstonedF "BELcreatedN~9!ÜA"
timestampN~9!ÜA"name "FileList.PNG"
title"FFFileList.PNG"BS
mimetype" image/png"BS
filetype"ETXpng"VT
pretty_type"ETXPNG"
user"VTU07D2DJ6T3N"
user_team"VT07DCJWDB33"BS
editableF"EOTsizeIEPNAK"
"EOTmode"ACKhosted"VT
is_externalF"
external_type"~"
is_publicT"DC1public_url_sharedF"SO
display_as_botF"BS
username"~"VTurl_private"Fhttps://files.slack.com/files-pri/T07DCJWDB33-F07D04VFM0A/filelist.png"DC4
url_private_download"Ohttps://files.slack.com/files-pri/T07DCJWDB33-F07D04VFM0A/download/filelist.png"DC2
```

[그림 12] https_app.slack.com_0.indexeddb.blob 내 3 파일 내용 중 일부 - 1

HxD를 통해 IndexedDB 내 id U07D2DJ6T3N를 가진 Logan과 관련된 기록을 보면 filelist.png라는 사진을 올렸음을 유추해볼 수 있습니다.

```
"STXid"VT "F07DCQYDK25"
is_tombstonedF "BELcreatedN~9!ÜA" timestampN~9!ÜA"EOT"
name"RSAll-In-One-Python-Projects.zip"ENQ
title"RSAll-In-One-Python-Projects.zip"BS
mimetype"STapplication/zip"BS
filetype"ETXzip"VT
pretty_type"ETXZip"EOT
user"VTU07D2DJ6T3N"
user_team"VT07DCJWDB33"BS
editableF"EOTsizeI~ACK"EOT
mode"ACKhosted"VT
is_externalF"
external_type"~" is_publicT"DC1
public_url_sharedF"SOdisplay_as_botF"BS
username"~"VT
url_private"Xhttps://files.slack.com/files-pri/T07DCJWDB33-F07DCQYDK25/all-in-one-python-projects.zip"DC4
url_private_download"ahhttps://files.slack.com/files-pri/T07DCJWDB33-F07DCQYDK25/download/all-in-one-python-pro
media_display_type"BEIunknown"
permalink"bhhttps://myworkspace-yhp4917.slack.com/files/U07D2DJ6T3N/F07DCQYDK25/all-in-one-python-projects.zip"
```

[그림 13] https_app.slack.com_0.indexeddb.blob 내 3 파일 내용 중 일부 - 2

또한, all-in-one-python-projects.zip 이라는 파일도 Logan user의 id인 U07D2DJ6T3N로 Slack 아티팩트에 남아있는 것을 확인하였습니다.

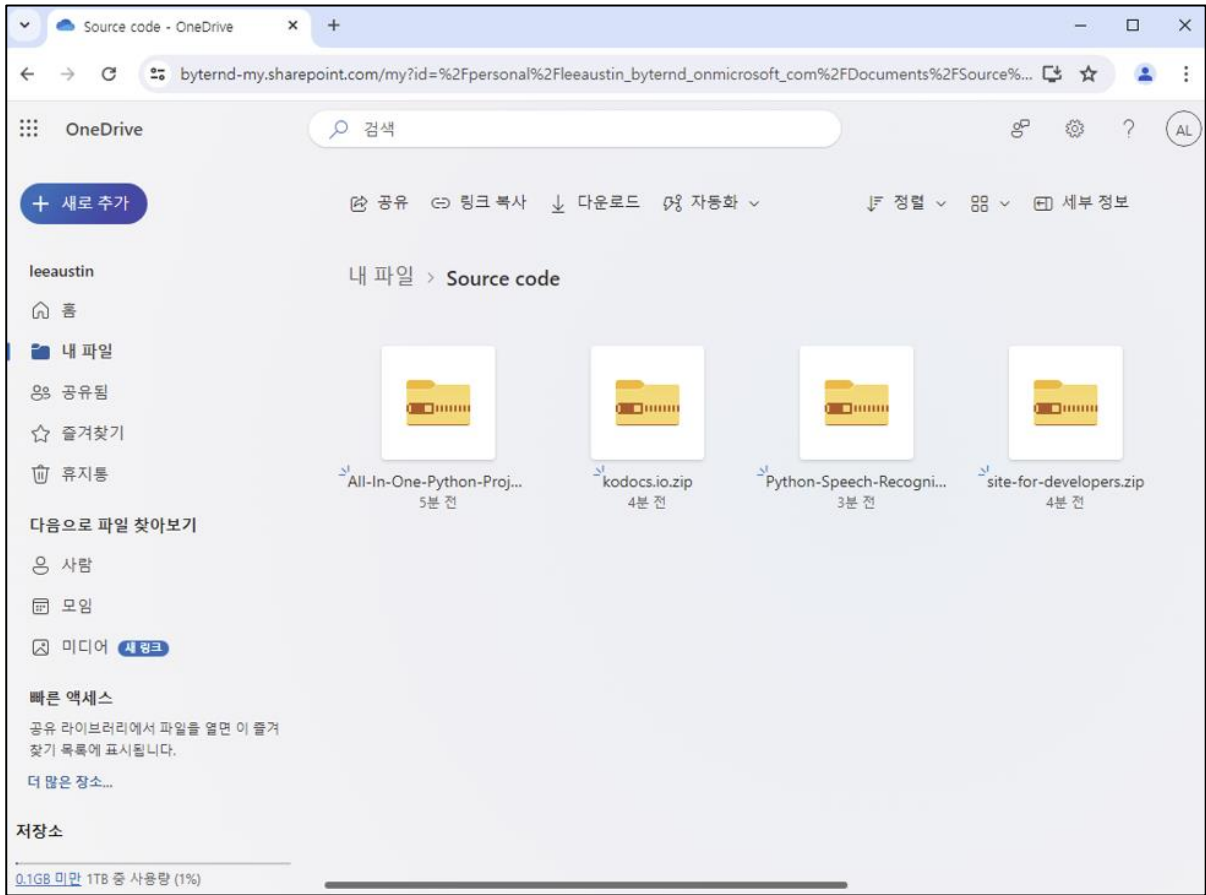
```

C:\Users\hyunnn\Documents\Github\Slack-Parser\python_Slack-Parser.py
Slack-Parser
By: 0xM0h0m3d
Insert Slack Database Path (%appdata%\Slack\IndexedDB\http-app.slack.com.0.indexeddb.blob.v): 3
Insert the data you want (users, messages, files, workspace): users
Users:
logalee9124 email logalee9124@gmail.com is_bot False is_admin True is_owner True primary_owner True Profile Picture https://ca.slack-edge.com/T07DC3M0833-U0702D36T3M-g3593c9g75e8-102
raymondhang61 email raymondhang61@gmail.com is_bot False is_admin False is_owner False primary_owner False https://ca.slack-edge.com/T07DC3M0833-U0702D36T3M-g422f427d8e7-102
Slackbot email Not found is_bot True is_admin False is_owner False primary_owner False https://ca.slack-edge.com/T07DC3M0833-U0702D36T3M-g422f427d8e7-102
Insert the data you want (users, messages, files, workspace): files
Name: filelist.png Download URL: https://files.slack.com/files-prl/T07DC3M0833-F07D0HVF0MA/download/filelist.png Normal link: https://myworkspace-yhp4917.slack.com/files/U0702D36T3M/F07D0HVF0MA/filelist.png
All-In-One-Python-Projects.zip Download URL: https://files.slack.com/files-prl/T07DC3M0833-F07D0HVF0MA/download/all-in-one-python-projects.zip Normal link: https://myworkspace-yhp4917.slack.com/files/U0702D36T3M/F07D0HVF0MA/all-in-one-python-projects.zip
Insert the data you want (users, messages, files, workspace):

```

[그림 14] Slack-parser.py 오픈소스를 수정한 결과

그리고, 추가로 Slack-parser에서 파싱해올 수 없었던 파일 부분을 가져오기 위해 코드를 수정해서 실행한 결과 logan의 Userid값으로 기록된 파일을 조회할 수 있었습니다. 그 결과, filelist.png와 all-in-one-python-projects.zip 파일을 확인할 수 있었습니다.

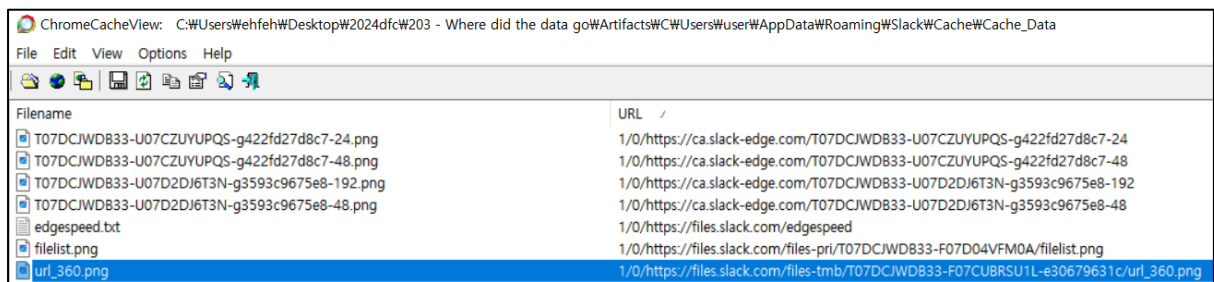


[그림 15] Filelist.png

ChromeCacheview로 확인한 Filelist.png는 대화 내용 중에 Logan이 언급한 It's a list of source codes I have access to를 통해 보낸 Source code 리스트 사진으로 파악됩니다. 다만, Slack 아티팩트 상에 남아있는 기록으로 판단하건대, 유출된 것으로 파악되는 파일은 **all-in-one-python-projects.zip**으로 보입니다.

답 : All-In-One-Python-Projects.zip

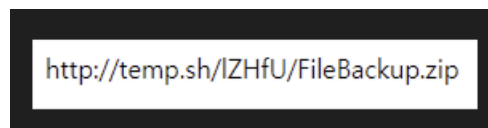
6. Logan uploaded a file to the temp.sh site. Which file did he upload? (30 points)



Filename	URL
T07DCJWDB33-U07CZUYUPQS-g422fd27d8c7-24.png	1/0/https://ca.slack-edge.com/T07DCJWDB33-U07CZUYUPQS-g422fd27d8c7-24
T07DCJWDB33-U07CZUYUPQS-g422fd27d8c7-48.png	1/0/https://ca.slack-edge.com/T07DCJWDB33-U07CZUYUPQS-g422fd27d8c7-48
T07DCJWDB33-U07D2DJ6T3N-g3593c9675e8-192.png	1/0/https://ca.slack-edge.com/T07DCJWDB33-U07D2DJ6T3N-g3593c9675e8-192
T07DCJWDB33-U07D2DJ6T3N-g3593c9675e8-48.png	1/0/https://ca.slack-edge.com/T07DCJWDB33-U07D2DJ6T3N-g3593c9675e8-48
edgespeed.txt	1/0/https://files.slack.com/edgespeed
filelist.png	1/0/https://files.slack.com/files-pri/T07DCJWDB33-F07D04VFM0A/filelist.png
url_360.png	1/0/https://files.slack.com/files-tmb/T07DCJWDB33-F07CU8RSU1L-e30679631c/url_360.png

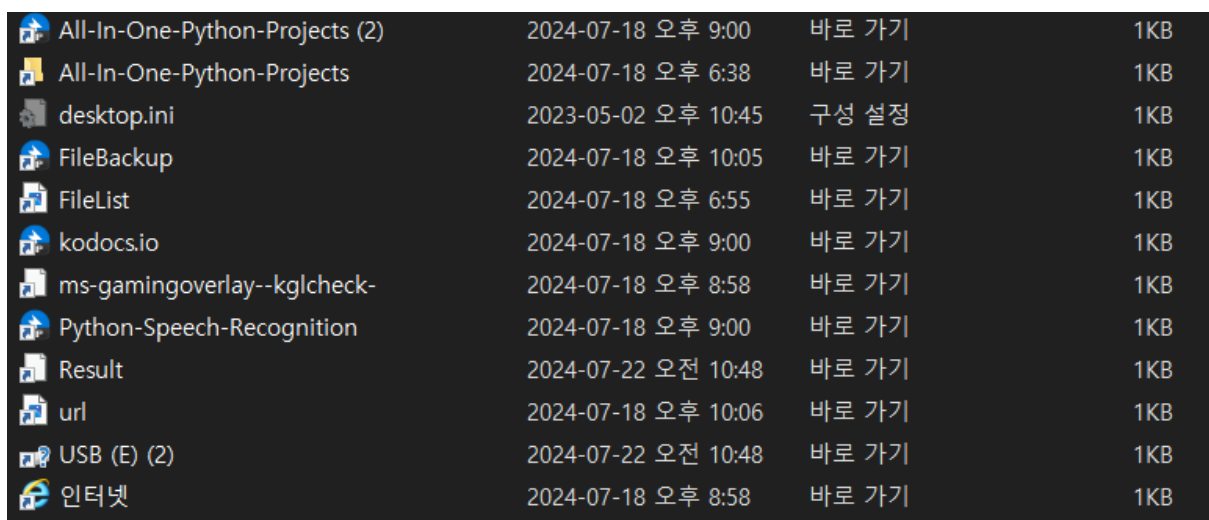
[그림 16] chromecacheview를 통해 Slack 아티팩트 내 Cache Data 확인

Slack 아티팩트 내 Cache Data에서 Logan이 temp.sh 사이트에 업로드한 파일로 판단되는 이미지 파일을 위 그림과 같이 찾았습니다.



[그림 17] url_360.png

url_360.png 파일을 열면 위 그림과 같이 http://temp.sh/IZHfU/FileBackup.zip 이라는 URL을 확인할 수 있습니다.



All-In-One-Python-Projects (2)	2024-07-18 오후 9:00	바로 가기	1KB
All-In-One-Python-Projects	2024-07-18 오후 6:38	바로 가기	1KB
desktop.ini	2023-05-02 오후 10:45	구성 설정	1KB
FileBackup	2024-07-18 오후 10:05	바로 가기	1KB
FileList	2024-07-18 오후 6:55	바로 가기	1KB
kodocs.io	2024-07-18 오후 9:00	바로 가기	1KB
ms-gamingoverlay--kglcheck-	2024-07-18 오후 8:58	바로 가기	1KB
Python-Speech-Recognition	2024-07-18 오후 9:00	바로 가기	1KB
Result	2024-07-22 오전 10:48	바로 가기	1KB
url	2024-07-18 오후 10:06	바로 가기	1KB
USB (E) (2)	2024-07-22 오전 10:48	바로 가기	1KB
인터넷	2024-07-18 오후 8:58	바로 가기	1KB

[그림 18] CWUsersWuserWAppDataWRoamingWMicrosoftWWindowsWRecent

Recent 폴더에서도 FileBackup.zip이 존재했던 것을 확인할 수 있으며, 이를 통해 Logan이 업로드했던 파일은 **FileBackup.zip**으로 확인됩니다.

답 : FileBackup.zip