

Cristian Carrizales
Lab 8
11/14/2025
CS 362-Digital Forensics

Part 1 - Non Technical Summary

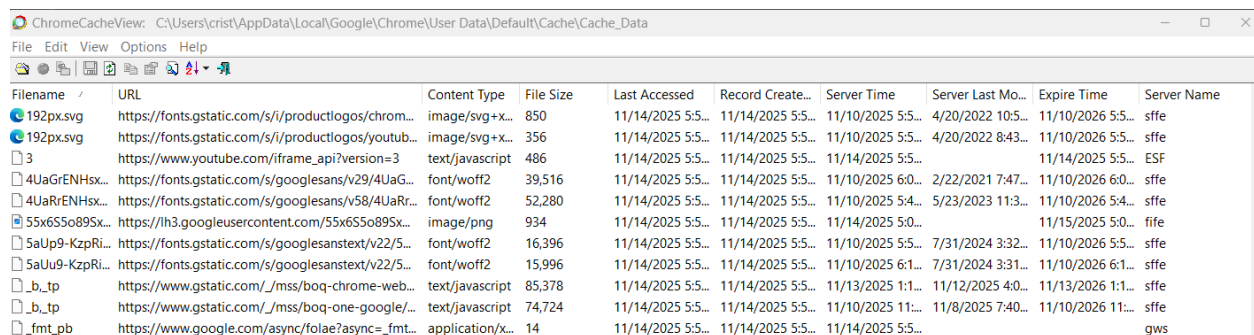
In this lab I continue to delve deeper into the world of forensic tools, using these tools to find “fingerprints” left behind on a Windows computer. Fingerprints like cached data, cookies browsing history and a log of what users searched for and clicked on for two separate browsers. Following that I looked into my systems security log through the program Event Log explorer to find an official record of when something logged onto my machine. Finally, I used some python scripts that were given in order to do several things like digging into my machines registry, peeking inside my recycle bin, and opening firefox database files in order to list every website I visited. I must say it was pretty rad to do that without the help of a GUI.

Part 2- Technical Summary

Part I: Web Browser Artifacts using NIRSOFT Tools

The beginning of this lab had me downloading a bunch of tools to gather forensic information from web browsers. The tools are called NIRSOFT tools and the browsers I will be scraping some data from are Windows and Mozilla Firefox. The data being gathered will be their respective caches, cookies, and history. Since I dont have chrome and firefox installed I will be needing to download and install them. Hopefully this keeps the information/files gathered on the lower side.

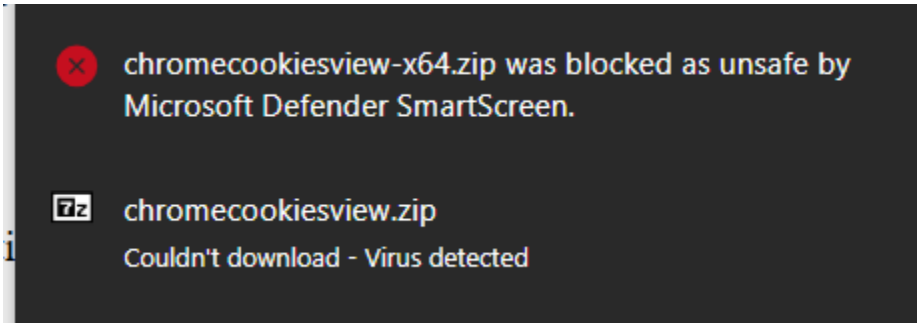
The first program opened was ChromeCacheView and in the photo below is a snippet of what appears on my screen. Unfortunately only opening chrome and downloading an ad-blocker still results in a good amount of data being generated. These are most likely common files that were downloaded alongside chrome as there are urls from youtube despite me never opening the website on my browser. Last accessed/ record created and server time help build a timeline of events for investigators even if something like browser history is deleted a cache files will be left behind



Filename	URL	Content Type	File Size	Last Accessed	Record Create...	Server Time	Server Last Mo...	Expire Time	Server Name
192px.svg	https://fonts.gstatic.com/s/i/productlogos/chrom...	image/svg+xml	850	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 5:5...	4/20/2022 10:5...	11/10/2026 5:5...	sffe
192px.svg	https://fonts.gstatic.com/s/i/productlogos/youtub...	image/svg+xml	356	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 5:5...	4/20/2022 8:43...	11/10/2026 5:5...	sffe
3	https://www.youtube.com/iframe_api?version=3	text/javascript	486	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 5:5...		11/14/2025 5:5...	ESF
4UaGrENHsx...	https://fonts.gstatic.com/s/googlesans/v29/4UaG...	font/woff2	39,516	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 6:0...	2/22/2021 7:47...	11/10/2026 6:0...	sffe
4UaRrENHsx...	https://fonts.gstatic.com/s/googlesans/v58/4UaRr...	font/woff2	52,280	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 5:4...	5/23/2023 11:3...	11/10/2026 5:4...	sffe
55x6S5o89Sx...	https://lh3.googleusercontent.com/55x6S5o89Sx...	image/png	934	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 5:0...		11/15/2025 5:0...	file
5aUp9-KzpRi...	https://fonts.gstatic.com/s/googlesansex/v22/5...	font/woff2	16,396	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 5:5...	7/31/2024 3:32...	11/10/2026 5:5...	sffe
5aUu9-KzpRi...	https://fonts.gstatic.com/s/googlesansex/v22/5...	font/woff2	15,996	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 6:1...	7/31/2024 3:31...	11/10/2026 6:1...	sffe
_b_tp	https://www.gstatic.com/_mss/boq-chrome-web...	text/javascript	85,378	11/14/2025 5:5...	11/14/2025 5:5...	11/13/2025 1:1...	11/12/2025 4:0...	11/13/2026 1:1...	sffe
_b_tp	https://www.gstatic.com/_mss/boq-one-google/...	text/javascript	74,724	11/14/2025 5:5...	11/14/2025 5:5...	11/10/2025 11:...	11/8/2025 7:40...	11/10/2026 11:...	sffe
_fmt_pb	https://www.google.com/async/foiae?async=_fmt...	application/x...	14	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 5:5...			gws

For the next step I was going to use a tool for viewing chrome cookies but after attempting to download from the website provided in the lab document my computer flagged the tools as

viruses. Every other tool I was to use was downloaded without issue so proceeding with caution I decided to skip this part of the lab.



The next two photos show my browsing history which corroborates what ive been saying previously, that ive recently downloaded chrome and the only sites I visited are a browser search and a page to download an adblock extension. Some of the more important columns are Referrer which tells the destination website exactly where the user just came from sent by the web browser to the server when a user clicks a link from one webpage to another. Typed Count = 0, meaning the user did not type any of the long urls. Transition Type = Generated means that i typed something into the address bar and the web browser generated a search url, = link meaning i clicked a link to get on “x” page

URL	Title	Visited On	Visit Count	Typed Count	Referrer	Visit Duration	Visit ID	Profile
https://chromewebstore.google.com/de...	uBlock - Chrome Web Store	11/14/2025 5:5...	1	0	https://www.google.com/search?q=...	00:00:23.989	4	Default
https://www.google.com/search?q=&so...	Google Search	11/14/2025 5:5...	1	0		00:00:00.130	1	Default
https://www.google.com/search?q=&so...	Google Search	11/14/2025 5:5...	1	0	https://www.google.com/search?q=...	00:00:04.536	2	Default
https://www.google.com/search?q=ubl...	ublock - Google Search	11/14/2025 5:5...	1	0		00:00:02.919	3	Default

URL Length	Transition Type	Transition Qualifiers	History File
86	Link	Chain Start,Chain E...	C:\Users\crist\AppData\Local\Goo...
98	Generated	Chain Start	C:\Users\crist\AppData\Local\Goo...
126	Link	Chain End,Client Re...	C:\Users\crist\AppData\Local\Goo...
241	Generated	Chain Start,Chain E...	C:\Users\crist\AppData\Local\Goo...

The next 2 photos are of what the nirsoft tools showed me for my mozilla webbrower cache and history

Like the chrome browser cache there's files downloaded for the cache during installation

Filename	Content Type	URL	File Size	Fetch Co...	Last Modified	Last Fetched	Expiration Time	Server Name	Server Response	Server Time	Server Last M...	Content Encod...	Cache Name
0.json	application/json	https://firefox.settings.services.mo...	26,982	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	nginx	HTTP/2 200	11/14/2025 5:4...	11/10/2025 1...	br	0898DD28ABA1A0...
0.json	application/json	https://firefox.settings.services.mo...	16,670	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 6:0...	nginx	HTTP/2 200	11/14/2025 5:4...	11/13/2025 4...	br	08AB6C9C55F4C0...
0.json	application/json	https://firefox.settings.services.mo...	7,500	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 6:1...	nginx	HTTP/2 200	11/14/2025 5:4...	11/14/2025 3...	br	111AE63D71402EF...
00940182-...	image/jpeg	https://firefox-settings-attachmen...	3,981	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	UploadServer	HTTP/2 200	11/14/2025 5:4...	10/7/2025 3...	identity	EAD97FA7B522D11...
00f39060-e...	application/json	https://firefox-settings-attachmen...	779	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 6:4...	UploadServer	HTTP/2 200	11/14/2025 5:4...	6/10/2025 3...	br	8364484FAE8A8C5...
025a8a1d-...	text/plain; chars...	https://incoming.telemetry.mozilla...	20	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:4...	openresty	HTTP/2 200	11/14/2025 5:4...	N/A	gzip	CEE69DD43EDA4E4...
0281c2c1-9...	text/plain; chars...	https://incoming.telemetry.mozilla...	20	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:4...	openresty	HTTP/2 200	11/14/2025 5:4...	N/A	gzip	1ED4C13377A2C38...
02841eb1-...	image/jpeg	https://firefox-settings-attachmen...	16,785	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	UploadServer	HTTP/2 200	11/14/2025 5:4...	8/6/2025 4:3...	identity	7D96914FE58C7D2...
0685a105-...	application/json	https://firefox-settings-attachmen...	22,704	1	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 6:0...	UploadServer	HTTP/2 200	11/14/2025 5:5...	3/13/2023 9...	gzip	0A5FD4027373226...
06bcc0db-...	image/jpeg	https://firefox-settings-attachmen...	6,852	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	UploadServer	HTTP/2 200	11/14/2025 5:4...	5/2/2025 1:4...	identity	012FE2463EAD21E...
06e44aaa-...	application/json	https://firefox-settings-attachmen...	26,325	1	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 6:4...	UploadServer	HTTP/2 200	11/14/2025 5:5...	3/13/2023 9...	gzip	9DA9E485979BC27...
06f399f8b4...	image/png	https://merino-images.services.mo...	827	2	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 6:2...	UploadServer	HTTP/2 200	11/14/2025 5:2...	11/4/2025 12...	identity	91A728988933805...
08926800-...	image/png	https://firefox-settings-attachmen...	20,301	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 6:0...	UploadServer	HTTP/2 200	11/14/2025 5:4...	10/27/2025 1...	identity	4E4912656FC30889...
08e4fbc-2-...	image/png	https://firefox-settings-attachmen...	812	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 6:0...	UploadServer	HTTP/2 200	11/14/2025 5:4...	3/12/2025 1...	identity	825090163ACD917...
0907bf00-f...	application/json	https://firefox-settings-attachmen...	1,947	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	UploadServer	HTTP/2 200	11/14/2025 5:4...	10/16/2025 1...	br	D7824ACC9B4E041...
0934183b-...	image/png	https://firefox-settings-attachmen...	5,984	1	11/14/2025 5:4...	11/14/2025 5:4...	11/14/2025 5:5...	UploadServer	HTTP/2 200	11/14/2025 5:4...	8/7/2025 1:0...	identity	B6111CDD9B3A028...
098599ba8...	image/png;char...	https://merino-images.services.mo...	4,155	3	11/14/2025 5:5...	11/14/2025 5:5...	11/14/2025 6:5...	UploadServer	HTTP/2 200	11/14/2025 5:5...	11/3/2025 11...	identity	651894F53A042FC...

Mozilla history shows similar results to chrome as I only searched for ublock origin in the address bar (typed url) and downloaded it from the add on page shown after clicking the link provided by the search

URL	First Visit Date	Last Visit Date	Visit Cou...	Referrer	Host Name	Title	Record I...	Visit Type	Frecency	URL Length
https://addons.mozilla.org/en-US/firefo...	N / A	11/14/2025 5:5...	1	https://www.googl...		uBlock Origin – Get this ...	3	Link	98	61
https://www.google.com/search?client=...	N / A	11/14/2025 5:5...	1			Google Search	1	Typed URL	1950	74
https://www.google.com/search?client=...	N / A	11/14/2025 5:5...	1	https://www.googl...		ublock origin firefox - G...	2	Link	98	102

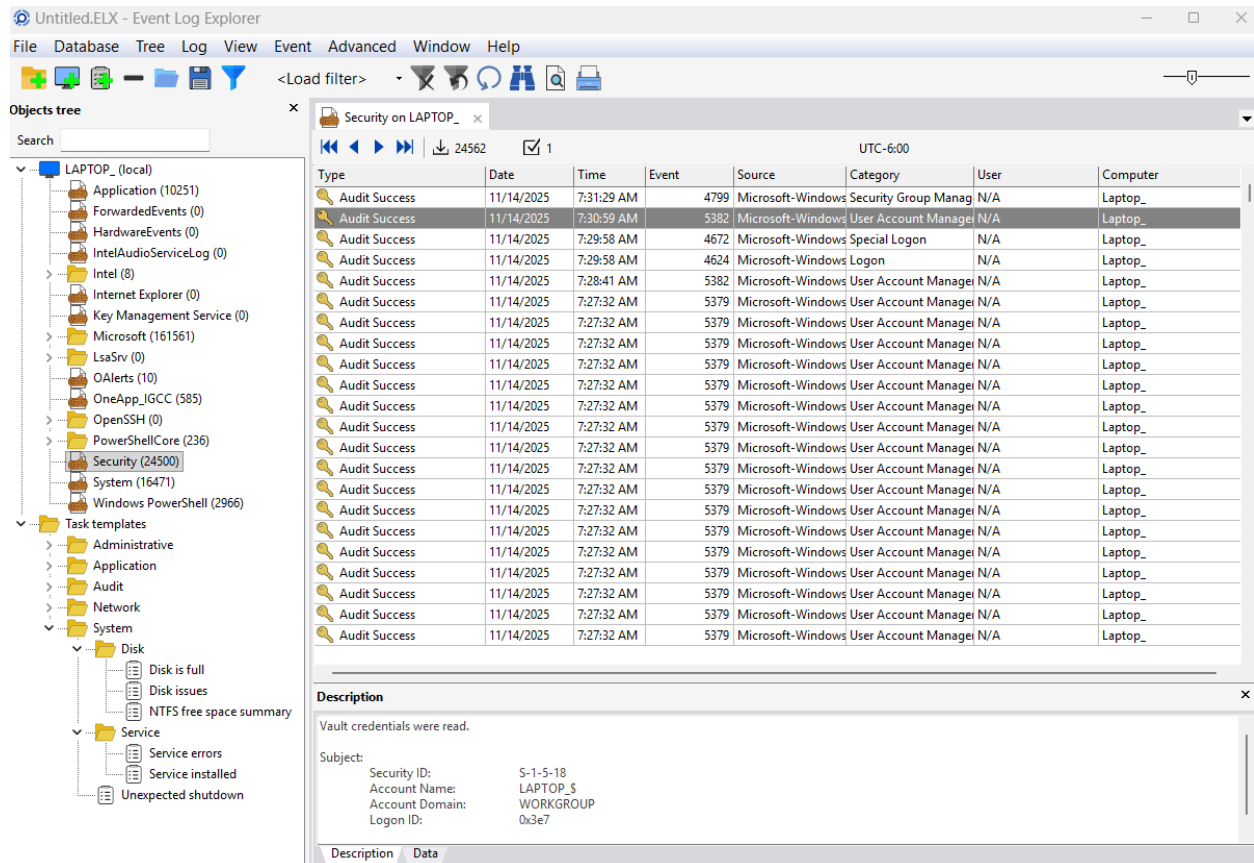
This go around I was able to see the cookies from my firefox browser. Cookies with dates far in the future are persistent cookies, used for long-term tracking or to keep me logged in. Cookies with dates in the past or no date are often session cookies, which typically expire when the browser is closed. A cookie is a forensic goldmine; it can be used to find a user's email address, username, or account ID for a specific site even if history is deleted.

Domain/Host	Path	Name	Value	Expiration Date	Secure	Domain ...	Line/ID	Last Accessed	Created Time
google.com	/	AEC	AaJma5sssZC53PC...	8/21/1987 9:00...	Yes		1	11/14/2025 5:5...	11/14/2025 5:5...
google.com	/verify	SNID	AICKk4xZzyQ_Q2d...	11/7/1995 7:55...	Yes		3	11/14/2025 5:5...	11/14/2025 5:5...
google.com	/	__Secure-STRP	ADq1D7q1eEdZBK...	3/25/2039 10:3...	No		4	11/14/2025 5:5...	11/14/2025 5:5...
google.com	/	NID	526=tB9cVZjHDuX...	11/7/1995 8:05...	Yes		6	11/14/2025 5:5...	11/14/2025 5:5...
mozilla.org	/	_ga	GA1.2.1732958301...	7/26/2045 7:44...	No		16	11/14/2025 5:5...	11/14/2025 5:5...
mozilla.org	/	_gid	GA1.2.2027086378...	12/16/2041 10:...	No		17	11/14/2025 5:5...	11/14/2025 5:5...
mozilla.org	/	_gat	1	3/23/2039 4:00...	No		18	11/14/2025 5:5...	11/14/2025 5:5...
mozilla.org	/	_ga_B9CY1C9V...	GS2.1.s1763121015...	7/26/2045 9:05...	No		19	11/14/2025 5:5...	11/14/2025 5:5...
www.google.c...	/	DV	E6eIRSxm_IZEM4d...	3/29/2039 9:44...	No		5	11/14/2025 5:5...	11/14/2025 5:5...

Part II: Viewing, Monitoring, and Analyzing Windows Events

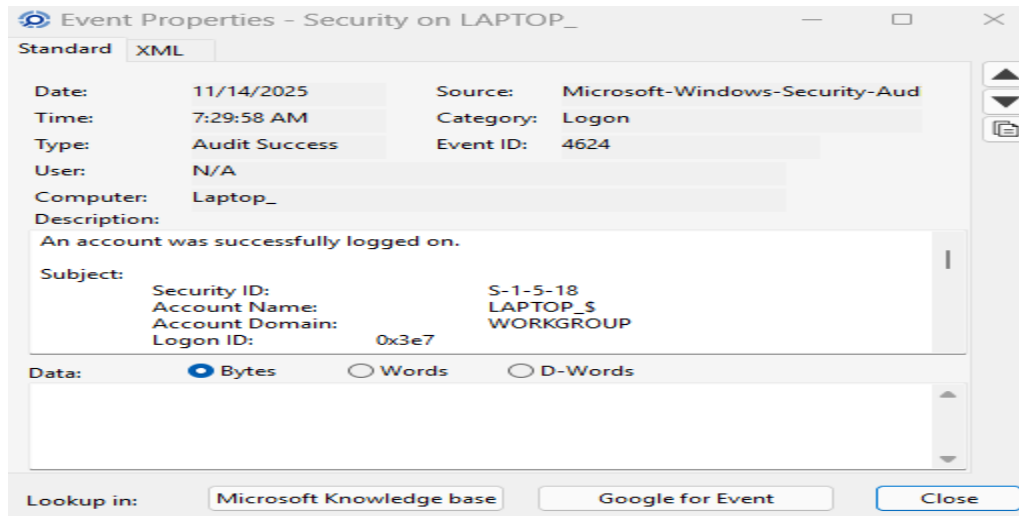
This next part had me downloading a program used to view Windows events called Event Log

Explorer. Although I am not sure how much different this program is from the default event viewer already on my windows system. Plus I had viewed and recorded logs for windows in a previous lab so ill keep this part brief for the most part.



This image shows the chronological master log of security events on my computer and for the next step I'll be taking a closer look at event 4624 "An account was successfully logged on."

From the screen shot of the details for the log of event 4624 within the subject section I can see who signed in(Security ID: S-1-5-18 Account Name: LAPTOP_\$ Account Domain: WORKGROUP) the information is essentially saying that the computer itself authenticating to perform a background task or start a service at 7:29:58 and that it was successful.



Part IV: Extracting Windows Security Account Manager (SAM)

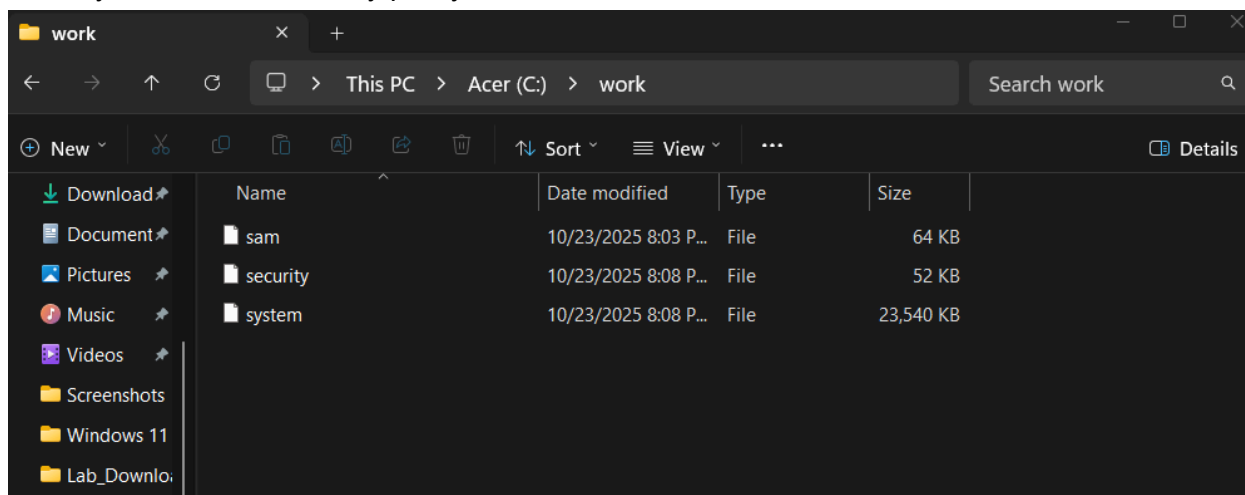
The following part has me extracting the file that stores all the hashed passwords for the local users on my machine. The SAM file cannot be copied and pasted as windows locks it while it's running. To get the files, I have to use a special registry command. Opening the command prompt as the administrator and running these commands `mkdir c:\work`, `reg save hklm\sam c:\work\sam`, `reg save hklm\security c:\work\security`, and `reg save hklm\system c:\work\system` saves the sam, security, and file will be saved into a new folder named work.

What the files do exactly are as follows

The sam file contains the usernames and the hashed passwords

System file contains the key(SYSKEY) that is used to encrypt the sam file

Security file contains security policy data



(From the lab pdf showing a linux VM)

With the command `samdump2` an analyst(hacker) could now take these three files, boot into a

separate machine and use a tool to dump the hashes. The command uses the system key to unlock the sam file, revealing the password hashes for all users someone could then try to crack these hashes to find the original password

Part V: Handling Windows Registry using Python

The next part of the lab has me running provided code(in python) in order to enumerate profiles from the registry. The winreg library in the code is used to connect to a top level spot in the windows registry called Hkey_USERS. The spot is a container that holds the personal profile settings for every user that has ever logged on to my machine. In the output of the code subkeys are listed that operate like sub-folders inside HKEY_USERS. The numbers S-1-5-18, S-1-5-19, S-1-5-20 are known SIDs(Security Identifiers) built for system accounts(LocalSystem, LocalService, NetworkService). The numbers that look similar with a much longer string of numbers in the end is the SID of a human user on my machine (S-1-5-21-3818357903-3657931246-2350636298-1001) An important note about SID is that even if a user changes their username the SID stays the same. SID can be used to trace someone's activity everywhere in the OS.

```
REGISTRY.py x Registry2.py
1 import winreg
2
3 reg = winreg.ConnectRegistry( computer_name: None, winreg.HKEY_USERS)
4
5 key = winreg.OpenKey(reg, sub_key: None)
6 lst_sids = []
7
8 for n in range(10):
9
10     try:
11         x = winreg.EnumKey(key, n)
12         lst_sids.append(x)
13
14         print("{:d}: {:s}".format( *args: n, x))
15     except:
16         break
17
18
```

Run REGISTRY x

```
C:\Users\cris\anaconda3\envs\school-env\python.exe C:\Users\cris\PycharmProjects\PythonProject\REGISTRY.py
0: .DEFAULT
1: S-1-5-19
2: S-1-5-20
3: S-1-5-21-3818357903-3657931246-2350636298-1001
4: S-1-5-21-3818357903-3657931246-2350636298-1001_Classes
5: S-1-5-18
Process finished with exit code 0
```

The next step had me combining code from the previous step and a new snippet of code to find evidence of a specific software installed by a specific user. The code uses `convtolocaltime(ts)` function which is supposed to translate windows time into human readable time. Unfortunately I ran into many errors trying to implement the code so I will provide some detail about what the code was supposed to do and pictures of my code.

`subkey = "S-1-5-21-3818357903-3657931246-2350636298-1001"` hardcodes the SID of the user(me)
`subkeyfield1 = subkey+r"\SOFTWARE\MICROSOFT\Office"` Was trying to create the full registry path to my personal microsoft office settings
`key = winreg.OpenKey(reg, subkeyfield1)` code is trying to open the office key but fails because the office is not installed?

And for `n in range(500):` `x = winreg.EnumKey(key, n)` was supposed to loop thru the subkeys inside the office key and get the name of the subkey

```
REGISTRY.py × Registry2.py
1 import winreg
2
3 reg = winreg.ConnectRegistry( computer_name: None, winreg.HKEY_USERS)
4
5 key = winreg.OpenKey(reg, sub_key: None)
6 lst_sids = []
7
8 for n in range(10):
9
10     try:
11         x = winreg.EnumKey(key, n)
12         lst_sids.append(x)
13
14         print("{:d}: {:s}".format( *args: n, x))
15     except:
16         break
17
18 from datetime import datetime, timedelta
19 import pytz
20 from dateutil.tz import tzlocal
21
22 def convtolocaltime(ts): 1 usage
23     ds = datetime( year: 1601, month: 1, day: 1) + timedelta(microseconds=ts // 10)
24     ds = ds.replace(tzinfo=pytz.UTC)
```

```
REGISTRY.py × Registry2.py
22 def convtolocaltime(ts): 1 usage
    ds = ds.astimezone(tzlocal())
26
27     return ds
28
29     subkey = "S-1-5-21-3818357903-3657931246-23500636298-1001"
30
31     subkeyfield1 = subkey+r"\SOFTWARE\MICROSOFT\Office"
32     key = winreg.OpenKey(reg, subkeyfield1)
33
34     for n in range(500):
35         try:
36             x = winreg.EnumKey(key, n)
37             subkeyfieldi = subkeyfield1 + "\\ " + x
38             subkeyi = winreg.OpenKey(reg, subkeyfieldi)
39
40             ts = winreg.QueryInfoKey(subkeyi)
41
42             winreg.CloseKey(subkeyi)
43
44             localtime = convtolocaltime(ts[2])
45
46             print(x, ":", localtime)
47         except:
48             break

Run REGISTRY ×
4: S-1-5-21-3818357903-3657931246-23500636298-1001_Classes
5: S-1-5-18
Traceback (most recent call last):
  File "C:\Users\cris\PycharmProjects\PythonProject\REGISTRY.py", line 32, in <module>
    key = winreg.OpenKey(reg, subkeyfield1)
FileNotFoundError: [WinError 2] The system cannot find the file specified

> Process finished with exit code 1
```

I could not fix the file not found error maybe its because I dont have Office installed? Either way using a different pathway still results in the same error the change is located on like 31

```
26
27     return ds
28
29     subkey = "S-1-5-21-3818357903-3657931246-23500636298-1001"
30
31     subkeyfield1 = subkey+r"\SOFTWARE\Microsoft\Windows\CurrentVersion"
32     key = winreg.OpenKey(reg, subkeyfield1)
33
34     for n in range(500):
35         try:
36             x = winreg.EnumKey(key, n)
37             subkeyfieldi = subkeyfield1 + "\\ " + x
38             subkeyi = winreg.OpenKey(reg, subkeyfieldi)
39
40             ts = winreg.QueryInfoKey(subkeyi)
41
42             winreg.CloseKey(subkeyi)
43
44             localtime = convtocaltime(ts[2])
45
46             print(x, ":", localtime)
47         except:
48             break
```

Run REGISTRY x

```
4: S-1-5-21-3618357903-3657931246-23500636298-1001_CLASSES
5: S-1-5-18
Traceback (most recent call last):
  File "C:\Users\crist\PycharmProjects\PythonProject\REGISTRY.py", line 32, in <module>
    key = winreg.OpenKey(reg, subkeyfield1)
FileNotFoundError: [WinError 2] The system cannot find the file specified

Process finished with exit code 1
```

Part VI: Handling Windows Recycle Bin using Python

In this next part I run some more code provided to me but this time the code will handle things inside the recycling bin on my windows system. Pictured below is the code running as intended but I had to make some minor modifications from the provided code to get it running. Nothing too extreme as it was just swapping line 13 to direct to a file that exists on my system. The code lists the recycle bin contents, prints the SID, creates test.txt in my designated folder, deletes the .txt file and then undeletes it.

```

REGISTRY.py  Registry2.py  Recycle.py x
1  import winshell
2  import re
3
4  r = list(winshell.recycle_bin())
5
6  for x in r:
7      print(x.original_filename(), x.recycle_date(), sep='\t')
8
9  f1 = r[0].filename()
10 y = re.search(pattern=r"S.*\d{4}", f1)
11 print(y.group(0))
12
13 path = r'C:\Users\cris\u\Documents\test.txt'
14 with open(path, 'w') as file:
15     file.write('This is a test file')
16
17 winshell.delete_file(path)
18 winshell.undelete(path)

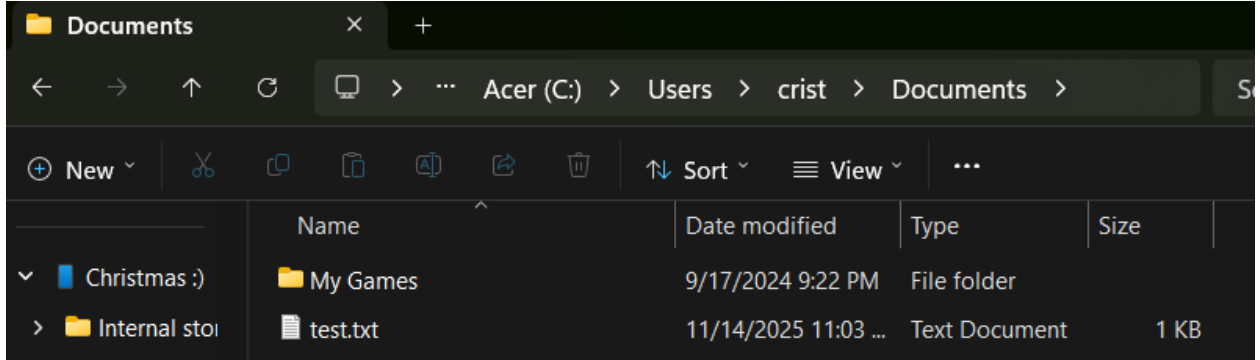
Run  Recycle x

C:\Users\cris\anaconda3\envs\school-env\python.exe C:\Users\cris\PycharmProjects\PythonProject\Recycle.py
C:\Users\cris\AppData\Roaming\discord\GPUCache 2025-09-18 23:14:11+00:00
C:\Users\cris\AppData\Roaming\discord\Code Cache 2025-09-18 23:14:06+00:00
C:\Users\cris\OneDrive\Desktop\ubuntu-24.04.3-desktop-amd64.iso 2025-09-12 07:44:19+00:00
C:\Users\cris\Downloads\SteamSetup.exe 2025-09-22 07:59:32+00:00
C:\Users\cris\Downloads\Core-Temp-setup-v1.18.1.0.exe 2025-09-22 07:59:22+00:00
C:\Users\cris\Downloads\hwi64_810.exe 2025-09-22 07:59:20+00:00
C:\Users\cris\OneDrive\Desktop\kali-linux-2025.2-installer-amd64.iso 2025-09-12 07:44:19+00:00
C:\Users\cris\Downloads\dog-8781844_1280.png 2025-09-22 23:10:21+00:00
C:\Users\cris\AppData\Roaming\discord\Cache 2025-09-18 23:14:03+00:00
C:\Users\cris\OneDrive\Desktop\Windows.iso 2025-09-12 07:44:19+00:00
S-1-5-21-3818357903-3657931246-2350636298-1001

Process finished with exit code 0

```

Test.txt showing up in the file designated



Part VI: Reading Browser History, Cookies, and Cache using Python

The code provided and to be replicated needed a couple of changes. First the file path needed to be updated to where the "default release file is actually located on my system but additionally I had to swap from "Local " to "Roaming " to get the output I desired, not doing so results in nothing being printed on the console. Important .sqlite files include places.sqlite which contains all the user browsing history, bookmarks, downloads, and favorite sites. Cookies.sqlite which stores all cookies from websites and formhistory.sqlite which stores data that was typed into web forms. This script functions similarly to the tools I used earlier in the lab.

```
1 import os
2
3 path = r'C:\Users\crist\AppData\Roaming\Mozilla\Firefox\Profiles\wm44me1q.default-release'
4
5 files = os.listdir(path)
6
7 for file in files:
8     if file.endswith(".sqlite") or file.endswith(".db"):
9         print(file)
```

Run MZFiles x

```
C:\Users\crist\anaconda3\envs\school-env\python.exe C:\Users\crist\PycharmProjects\PythonProject\MZFiles.py
bounce-tracking-protection.sqlite
cert9.db
content-prefs.sqlite
cookies.sqlite
domain_to_categories.sqlite
favicons.sqlite
formhistory.sqlite
key4.db
logins.db
permissions.sqlite
places.sqlite
protections.sqlite
storage.sqlite
suggest.sqlite
webappsstore.sqlite

Process finished with exit code 0
```

The final part of the lab has me running one final script that retrieves browsing history from my mozilla browser. Some important lines include import sqlite3 which is python's built-in library for reading and writing SQLite databases needed for unlocking the file. statement = 'SELECT url, visit_count FROM moz_places;' is an SQL query, SELECT url, visit_count makes it so we only get the URL and visit count. FROM moz_places makes it so the script looks in a table named moz_places where firefox stores the browsing data. The lab script was incomplete so I had to find a way to print out the results into the console and close history.connect the final results and code are in the photos below.

```

1  import os
2  import sqlite3
3
4  path = r'C:\Users\crist\AppData\Roaming\Mozilla\Firefox\Profiles\wm44me1q.default-release'
5
6  files = os.listdir(path)
7
8  for file in files:
9      if file.endswith(".sqlite") or file.endswith(".db"):
10         print(file)
11
12     history = os.path.join(path, 'places.sqlite')
13     history_connect = sqlite3.connect(history)
14     history_cursor = history_connect.cursor()
15
16     statement = 'SELECT url, visit_count FROM moz_places;'
17     history_cursor.execute(statement)
18     results = history_cursor.fetchall()
19
20     print("\nFirefox Browsing History")
21     for row in results:
22         print(f"URL: {row[0]}, Visits: {row[1]}")
23
24     history_connect.close()

```

```

C:\Users\crist\anaconda3\envs\school-env\python.exe C:\Users\crist\PycharmProjects\PythonProject\MZFiles2.py
bounce-tracking-protection.sqlite
cert9.db
content-prefs.sqlite
cookies.sqlite
domain_to_categories.sqlite
favicons.sqlite
formhistory.sqlite
key4.db
logins.db
permissions.sqlite
places.sqlite
protections.sqlite
storage.sqlite
suggest.sqlite
webappsstore.sqlite

Firefox Browsing History
URL: https://support.mozilla.org/products/firefox, Visits: 0
URL: https://support.mozilla.org/kb/customize-firefox-controls-buttons-and-toolbars?utm\_source=firefox-browser&utm\_medium=default-bookmarks&utm\_campaign=
URL: https://www.mozilla.org/contribute/, Visits: 0
URL: https://www.mozilla.org/about/, Visits: 0
URL: https://www.google.com/search?client=firefox-b-1-d&q=ublock+origin+firefox, Visits: 1
URL: https://www.google.com/search?client=firefox-b-1-d&q=ublock+origin+firefox&sei=dRcXadytFv3n0PEPn4zcqAU, Visits: 1
URL: https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/, Visits: 1

Process finished with exit code 0

```

Part 3-Works Cited

PDF File Lab 9