Browser Forensics Collection Tool – Final Report

Grant Gollinger & Dylan Voss
IT 360 – Defensive Security
Illinois State University
Fall 2025

---

1. Introduction

The Browser Forensics Collection Tool was developed to automate the collection of browser artifacts from Linux systems for digital forensic investigations. Automating a daily task that forensic investigators conduct will help save time and prevent the possibility of user error. Browser activity, including history, cookies, cache files, and session data, is collected and hashed. These artifacts are often critical when reconstructing user behavior, identifying unauthorized access, and analyzing malicious activity.

Manual extraction of these artifacts is time-consuming and carries a high risk of accidental modification or omission. This tool provides a repeatable, standardized, and forensically sound method for collecting browser artifacts from both Firefox (Snap) and Chromium-based browsers. Our tool prioritized using Firefox, but any browser can be used with proper additions to the script.

By automating extraction and organizing output in a consistent format, the tool improves the speed and reliability of browser artifact collection while preserving evidence integrity for DFIR workflows.

---

2. Technical Implementation

The project is built completely in Bash and relies on standard Linux tools, along with SQLite utilities for reading browser databases. The script is broken into clean, separate sections so each part handles one job. Finding profiles, copying data, parsing databases, converting timestamps, creating hashes, and generating organized output.

2.1 Firefox (Snap) Artifact Extraction

Firefox installed through Snap stores profile data under:
`~/snap/firefox/common/.mozilla/firefox/`

The tool automatically detects the active profile directory (for example, `xxxxxx.default-release`) under this path before copying artifacts.

Once a profile is identified, the tool extracts:

- Browsing history (`history.sqlite`)
- Cookies (`cookies.sqlite`)
- Session recovery data (`sessionstore.jsonlz4`)
- Form history (`formhistory.sqlite`)
- Full Firefox `cache2/` directory

SQLite artifacts are exported to CSV using a read-only query, for example:

```
sqlite3 -readonly file.sqlite 'SELECT * FROM some_table;' > output.csv
```

This allows investigators to analyze the data without proprietary forensic software.

2.2 Chromium Artifact Extraction

For Chrome, Chromium, and Chromium-based browsers, the tool collects:

- Browsing history
- Cookies
- Top Sites
- Favicon database
- Network Action Predictor data

All SQLite-based files are parsed in read-only mode and exported to CSV for structured analysis.

2.3 Timestamp Normalization

Different browsers use different timestamp formats:

- **Firefox:** Microseconds since Unix epoch
- **Chromium:** WebKit timestamps (microseconds since January 1, 1601)

The tool converts both formats into standard readable timestamps:

```
echo "$timestamp / 1000000" | awk '{print strftime("%Y-%m-%d %H:%M:%S", $1)}'
```

Readable time values significantly simplify timeline reconstruction.

2.4 Evidence Integrity (SHA-256 Hashing)

To ensure forensic soundness, every raw and parsed file is hashed:

```
sha256sum <file> >> integrity_report.txt
```

This produces a verifiable record proving the integrity of each extracted artifact.

2.5 Organized Output Structure

Each run produces a timestamped evidence directory containing:

- Firefox artifacts
- Chromium artifacts
- CSV exports
- Integrity hash reports
- A summary folder for quick review

This structure supports clean reporting and standardized DFIR documentation.

2.6 Forensic Safety Controls

To avoid evidence contamination, the tool:

- Only **copies** files, never modifies originals
- Uses SQLite in strict read-only mode
- Warns the user if browsers are open
- Prevents overwrites of prior extractions
- Stores all output outside profile paths

These safeguards align with best practices in digital forensics.

---

3. Results

The Browser Forensics Collection Tool successfully extracted and processed browser artifacts from Linux systems running Firefox Snap and Chromium.

3.1 Firefox Results

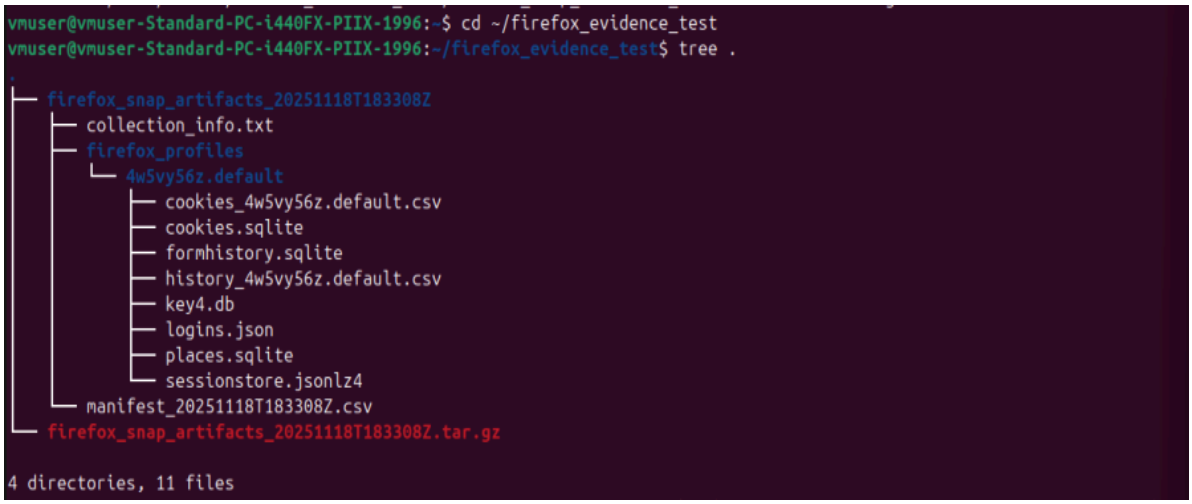
The tool produced:

- Complete browsing history in CSV format
- Parsed cookies (over 900 entries in testing)
- Full Firefox `cache2/` directory copied without corruption
- Recovered session data (`sessionstore.jsonlz4`)
- Form history in CSV
- SHA-256 integrity logs for all artifacts

All databases were parsed correctly, and WebKit timestamps were converted into proper human-readable formats.

3.3 Sample Output

- **Figure 1.** Firefox Evidence Tree



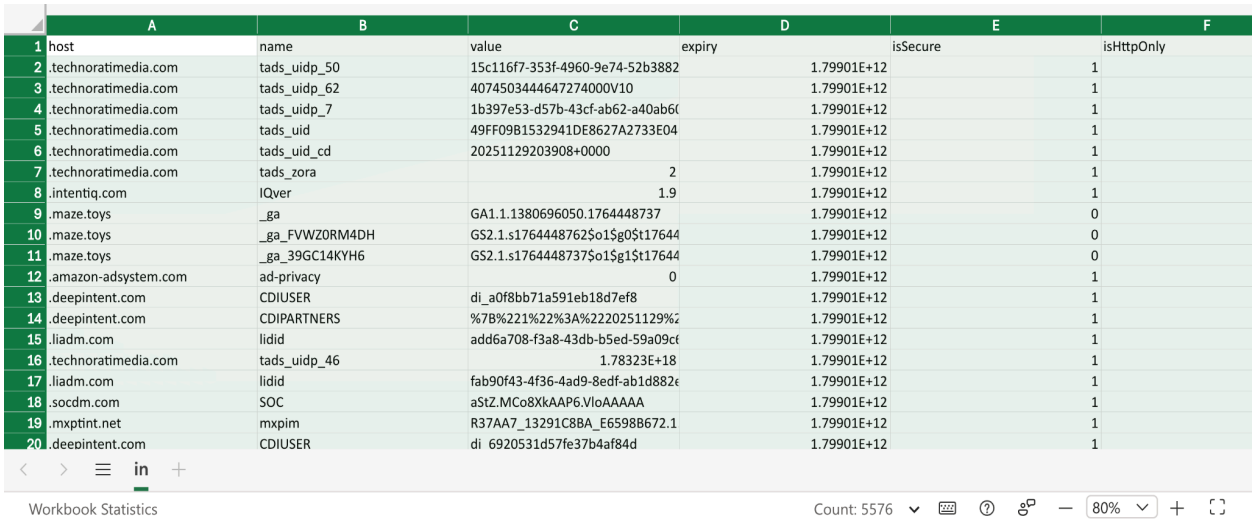
- **Figure 2.** Firefox cookies CSV

```
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/firefox_evidence_test$ head firefox_snap_artifacts*/firefox_profiles/*/cook
ies_*.csv
host,name,value,expiry,isSecure,isHttpOnly
.youtube.com,PREF,f6=40000000&tz=America.Chicago,1798050699283,1,0
.amazon-adsystem.com,ad-privacy,0,1798050693507,1,1
.amazon.com,session-id,145-2021336-9632957,1798050689094,1,0
.amazon.com,session-id-time,2082787201l,1798050689094,1,0
.amazon.com,session-token,FMdgwFxcJBLaFL5ifFWczJOFBrYGmHF5O5liuyiyr+B8E97LStsPyX7ouiGjXigoXg8mu7nNm80X2xDA6W7FFoX/Ju6oEl
39fSccSqjEPyQul+6PV6fh7roBlwLc6r/1B+SDJeZo7iW+pPCFKEF+itjZz9MsAfOqqSeJyivastZGrXYwDjeVFXi6/X5P7F0I3XPm4+G7Ou9HOtSYjSdIPb
xxxg3A8Wad9cyxYTvzb/vRJHlH2d0yD2POy8XimexSWBdRKaNeoZvJlbU4S7nBM+Cx6zToIuQ4g2Gf/8CvxIqv6ReFLRRZcKfwAUrgSc6zwYthW5NXddjYfE
hLQkZKo2RhGF3pVBH2,1798050689094,1,1
.reddit.com,loid,00000000228rgi07ri.2.1763483751943.Z0FBQUFBQnBIS0JuajZUT3hKSVFIdWVMb0szdGpaYWROT0w1NkxQcDVEcWtpM3MtNmY5
dWxFX1ZOOGFoYkFHV0Q5cTR6LWRsXzhvWnBSaXNyemxrX1FONVlaVGtqYWxaaDFsSEpabXlEOFhmSTE5eW9fRGVKbklBeTRQczdobjRtUkhZTVNKLU83WXQ,
1798043751989,1,0
.reddit.com,csv,2,1798043751989,1,0
.reddit.com,edgebucket,g90sKzHH5KIDz9gb5c,1798043751989,1,0
accounts.google.com,__Host-GAPS,1:sY3EaPZgoXJZyVYHqMGK1SIR6pkfoA:jJedo-TSTyGTIiND,1798043744443,1,1
```

- **Figure 3.** Integrity verification report

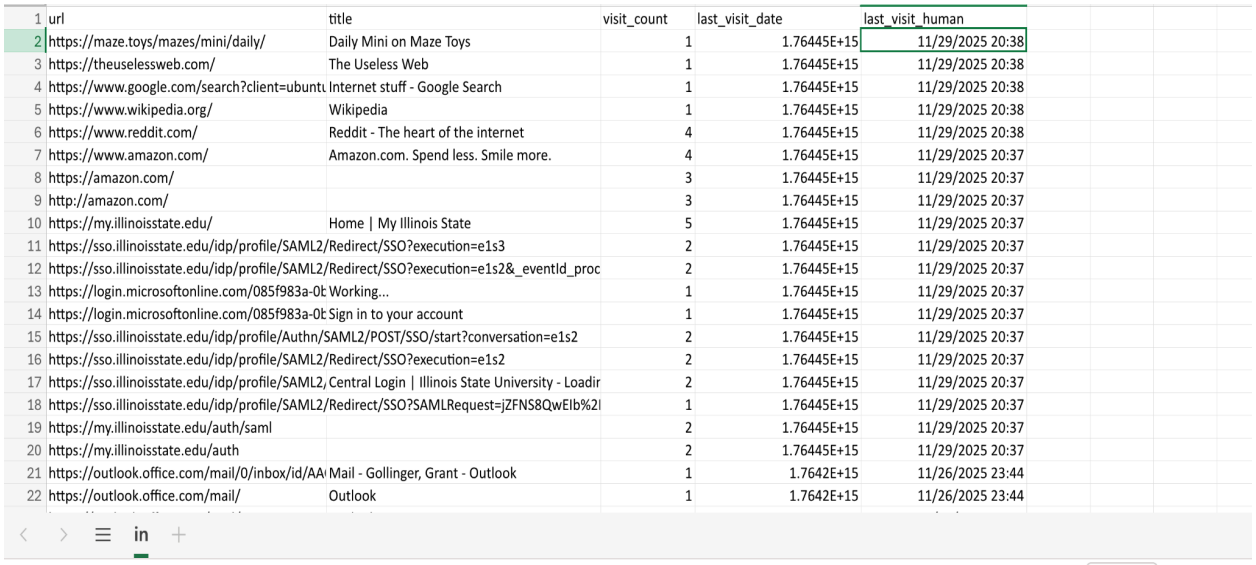
```
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/browser_evidence/firefox_snap_artifacts_20251129T204131Z$ ls
collection_info.txt  firefox_profiles  integrity_report.txt  manifest_20251129T204131Z.csv
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/browser_evidence/firefox_snap_artifacts_20251129T204131Z$ head integrity_re
port.txt
f081d00202c87e87195c06badf63513c70fdbc168258c53a284c4d67dae12187  ./collection_info.txt
5a7c2570d3f05396abf50055f10b7ee02e19180eb042a56ff19e0d37cc0010c7  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1006
66660
eb481f1f6189d69b2e256227e75fbe3ea0131d2c2b63b4d0324b59e0781ba00a  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1017
702641
0934949b5078db5776c394894bb1bc2f81f7fc0169ff9b533644c4ea623d5648  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1018
549586
03017258c0ab7489e93bdaa4b4f11d6c17a24c6c026d11716da7b501b14a6cec  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1040
395629
20c22aa6ac0a91a354d4c6971ea6e06534a37d2a24214f12e0244458372b53a3  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1061
513389
35f410b5a4bf71d6f4d2785d0195694e13e4efc92c34ba74290e5728357075bb  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1061
898831
e0f6bd120fa359e5ccf6472cefa20425304592a83e5e186b57827fcd00118b61  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1070
063498
a68832d9212e664ebdd354ab5dce5a590b2a878715be54ce2d59fcd136011395  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1091
129781
4e026b9b1fcf4b430783e660caaa18e5caf85a8855de3c497ad2d17c4ec52072  ./firefox_profiles/4w5vy56z.default/cache2/doomed/1103
652255
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/browser_evidence/firefox_snap_artifacts_20251129T204131Z$ find . -type f |
wc -l
20577
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/browser_evidence/firefox_snap_artifacts_20251129T204131Z$ wc -l integrity_r
eport.txt
20577 integrity_report.txt
vmuser@vmuser-Standard-PC-i440FX-PIIX-1996:~/browser_evidence/firefox_snap_artifacts_20251129T204131Z$
```

- **Figure 4.** Excel spreadsheet of raw cookies



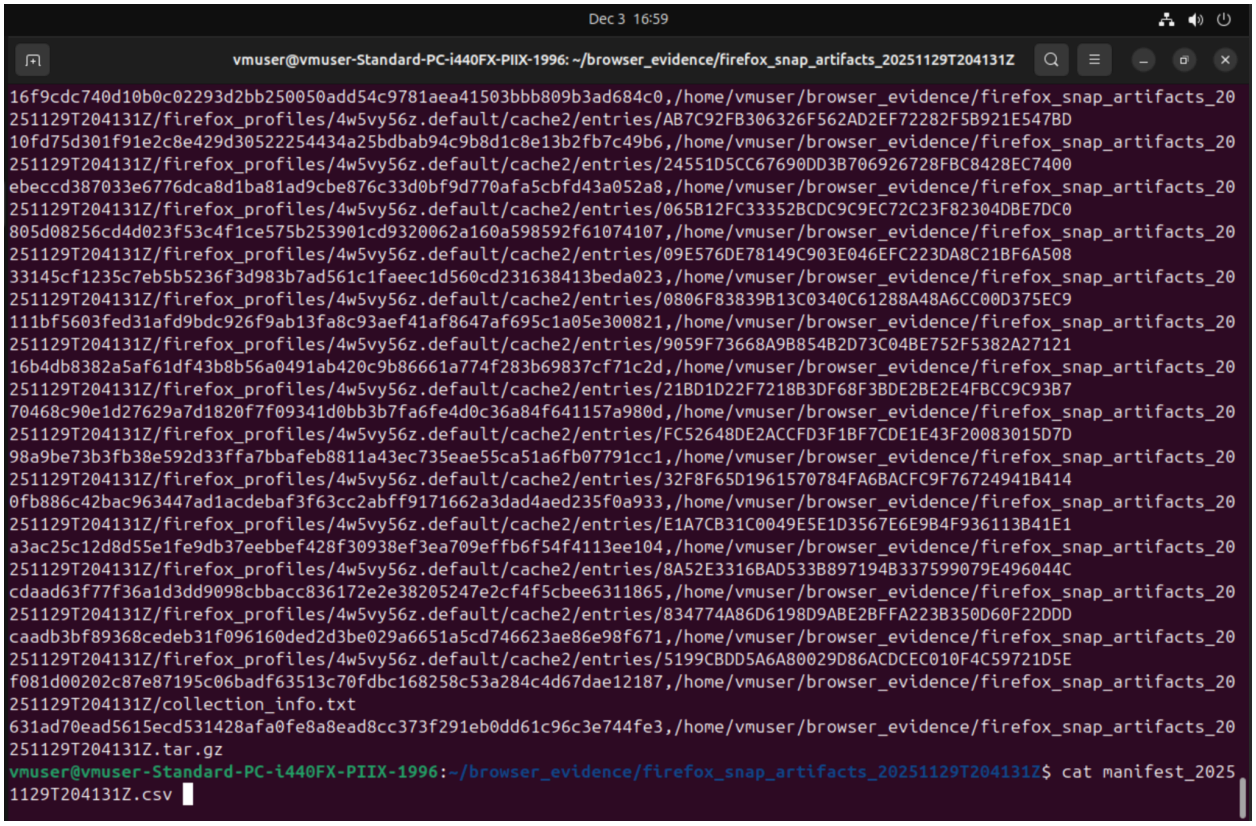
	A	B	C	D	E	F
1	host	name	value	expiry	isSecure	isHttpOnly
2	.technoratimedia.com	tads_uidp_50	15c116f7-353f-4960-9e74-52b3882	1.79901E+12	1	
3	.technoratimedia.com	tads_uidp_62	4074503444647274000V10	1.79901E+12	1	
4	.technoratimedia.com	tads_uidp_7	1b397e53-d57b-43cf-ab62-a40ab6(	1.79901E+12	1	
5	.technoratimedia.com	tads_uid	49FF09B1532941DE8627A2733E04	1.79901E+12	1	
6	.technoratimedia.com	tads_uid_cd	20251129203908+0000	1.79901E+12	1	
7	.technoratimedia.com	tads_zora	2	1.79901E+12	1	
8	.intentiq.com	IQver	1.9	1.79901E+12	1	
9	.maze.toys	_ga	GA1.1.1380696050.1764448737	1.79901E+12	0	
10	.maze.toys	_ga_FVWZ0RM4DH	GS2.1.s1764448762$o1$g0$t17644	1.79901E+12	0	
11	.maze.toys	_ga_39GC14KYH6	GS2.1.s1764448737$o1$g1$t17644	1.79901E+12	0	
12	.amazon-adsystem.com	ad-privacy	0	1.79901E+12	1	
13	.deepintent.com	CDIUSER	di_a0f8bb71a591eb18d7ef8	1.79901E+12	1	
14	.deepintent.com	CDIPARTNERS	%7B%221%22%3A%2220251129%2	1.79901E+12	1	
15	.liadm.com	lidid	add6a708-f3a8-43db-b5ed-59a09c(	1.79901E+12	1	
16	.technoratimedia.com	tads_uidp_46	1.78323E+18	1.79901E+12	1	
17	.liadm.com	lidid	fab90f43-4f36-4ad9-8edf-ab1d882e	1.79901E+12	1	
18	.socdm.com	SOC	aStZ.MCo8XkAAP6.VloAAAAA	1.79901E+12	1	
19	.mxptint.net	mxpim	R37AA7_13291C8BA_E6598B672.1	1.79901E+12	1	
20	.deepintent.com	CDIUSER	di_6920531d57fe37b4af84d	1.79901E+12	1	

- **Figure 5.** Excel spreadsheet of raw history



	url	title	visit_count	last_visit_date	last_visit_human
2	https://maze.toys/mazes/mini/daily/	Daily Mini on Maze Toys	1	1.76445E+15	11/29/2025 20:38
3	https://theuselessweb.com/	The Useless Web	1	1.76445E+15	11/29/2025 20:38
4	https://www.google.com/search?client=ubuntu	Internet stuff - Google Search	1	1.76445E+15	11/29/2025 20:38
5	https://www.wikipedia.org/	Wikipedia	1	1.76445E+15	11/29/2025 20:38
6	https://www.reddit.com/	Reddit - The heart of the internet	4	1.76445E+15	11/29/2025 20:38
7	https://www.amazon.com/	Amazon.com. Spend less. Smile more.	4	1.76445E+15	11/29/2025 20:37
8	https://amazon.com/		3	1.76445E+15	11/29/2025 20:37
9	http://amazon.com/		3	1.76445E+15	11/29/2025 20:37
10	https://my.illinoisstate.edu/	Home | My Illinois State	5	1.76445E+15	11/29/2025 20:37
11	https://sso.illinoisstate.edu/idp/profile/SAML2/	Redirect/SSO?execution=e1s3	2	1.76445E+15	11/29/2025 20:37
12	https://sso.illinoisstate.edu/idp/profile/SAML2/	Redirect/SSO?execution=e1s2&_eventId_proc	2	1.76445E+15	11/29/2025 20:37
13	https://login.microsoftonline.com/085f983a-0t	Working...	1	1.76445E+15	11/29/2025 20:37
14	https://login.microsoftonline.com/085f983a-0t	Sign in to your account	1	1.76445E+15	11/29/2025 20:37
15	https://sso.illinoisstate.edu/idp/profile/Authn/	SAML2/POST/SSO/start?conversation=e1s2	2	1.76445E+15	11/29/2025 20:37
16	https://sso.illinoisstate.edu/idp/profile/SAML2/	Redirect/SSO?execution=e1s2	2	1.76445E+15	11/29/2025 20:37
17	https://sso.illinoisstate.edu/idp/profile/SAML2/	Central Login | Illinois State University - Loadir	2	1.76445E+15	11/29/2025 20:37
18	https://sso.illinoisstate.edu/idp/profile/SAML2/	Redirect/SSO?SAMLRequest=jZFNS8QwElb%2I	1	1.76445E+15	11/29/2025 20:37
19	https://my.illinoisstate.edu/auth/saml		2	1.76445E+15	11/29/2025 20:37
20	https://my.illinoisstate.edu/auth		2	1.76445E+15	11/29/2025 20:37
21	https://outlook.office.com/mail/0/inbox/id/AA	Mail - Gollinger, Grant - Outlook	1	1.7642E+15	11/26/2025 23:44
22	https://outlook.office.com/mail/	Outlook	1	1.7642E+15	11/26/2025 23:44

- **Figure 6.** Manifest file that holds hashed values



3.4 Validation & Reliability

Testing confirmed:

- Correct handling of SQLite files
- Accurate timestamp conversion
- Stable performance with large datasets

- All evidence preserved with verified SHA-256 hashes
- No accidental modification of browser profiles

The tool performed reliably across multiple systems and test profiles.

---

4. Lessons Learned & Conclusion

4.1 What Worked Well

- Modular script design made the tool easy to update and troubleshoot.
- SQLite extraction and CSV conversion worked consistently.
- Timestamp normalization significantly improved analysis clarity.
- Integrity hashing strengthened the forensic soundness of the tool.
- Automatic detection of Firefox Snap profiles minimized user error.

4.2 Challenges Encountered

- WebKit timestamp conversion required research and custom logic.
- Snap’s sandboxed Firefox directory required specialized path handling.
- Large cookie exports required additional formatting to remain readable.
- Browser processes had to be closed to avoid SQLite locking issues.

4.3 Future Improvements

Potential enhancements include:

- A GUI interface

- Support for additional browsers (Brave, Edge, Tor)
- Domain classification and risk scoring

4.4 Conclusion

The Browser Forensics Collection Tool successfully provides a reliable, repeatable, and forensically sound method for extracting browser artifacts from Linux systems. By automating the collection of history, cookies, session data, and cache files—and applying integrity hashing—the tool aligns with modern digital forensics practices and significantly reduces human error.

This project demonstrates a strong understanding of DFIR principles, scripting, digital evidence handling, and artifact analysis. It provides a solid foundation for future enhancements and real-world forensic applications.