Practical Browser Forensics Checklist

Firefox

Profiles
- Identify a number of profiles in the browser
- Identify deleted profiles
- Identify the path of default profiles
  - **Windows** — C:\Users\Chintan\AppData\Roaming\Mozilla\Firefox\profiles.ini
    - Identify profile names which will be in this format:
      - b79i29xe.default-1585553309609
      - x9umjrk2g.default
    - Visit in each folder which contains multiple files such as:
      - Crashes
      - Datareporting
      - Minidumps
      - Storage
      - Security_state
      - etc...
    - If these folders exist, it means you are in the correct folder to start your investigation.
  - **Linux (kali)** — /home/chintan/.mozilla/firefox/profiles.ini
    - Identify profile names which will be in this format:
      - b79i29xe.default-1585553309609
      - x9umjrk2g.default
    - Visit in each folder which contains multiple files such as:
      - Crashes
      - Datareporting
      - Minidumps
      - Storage
      - Security_state
      - etc...
    - If these folders exist, it means you are in the correct folder to start your investigation.

Cookies
- Identify the number of cookies that are present in the browser
- Identify cookies that are going to expire soon (Cookies validity)
  - Search for cookies.sqlite to investigate all the cookies in the profile folder
    - Run the following command to view all the cookies:
      - `SELECT * from 'moz_cookies';`
      - Result

Login Information
- All the login information is saved in logins.json file.
  - On windows, this file can be found at the C:\Users\Chintan\AppData\Roaming\Mozilla\Firefox\Profiles\{PROFILE_NAME-FOLDER}
  - On Linux, it can be found at the /home/chintan/.mozilla/firefox/{PROFILE_NAME_FOLDER}
- Raw Output (cat logins.json)
- Information provided in this file may not provide good visibility in terms of the investigation. We can copy entire this data and put it in the https://jsonformatter.curiousconcept.com for further processing. This is designed to solve the problem by formatting and beautifying the JSON data so that it is easy to read and debug by human beings.
- Formatted Output
- Identify anomalies such as:
  - When the user changed the website's password?
  - Is that change password event occurred during/on/post-breach incident? Analyse and compare the incident timeline and event timeline (that includes date and time).
  - What are all email ids being used by a user on all websites or a specific website?
  - Identify any pattern of a password which can allow access to other similar websites a user is using.
  - How many times the did user change the password for a particular website?

Other information to look for your investigation
- Identify all email addresses used by a user
  - Search for the formhistory.sqlite file to start your investigation
  - Run the following query to find all results with email ids:
    - `SELECT * from 'moz_formhistory' where value LIKE '%@%';`
  - Result
  - This can be helpful to identify:
    - Which all websites user used to register company email id
    - What all email ids are being used by a user
- Identify other malicious applications/URLs supported by the firefox browser
  - Search for the handlers.json file to start your investigation
  - Handlers.json file stores the actions of applications and that is also helpful which links of applications are being supported by the firefox such as IRC, torrent (magnet), etc. This is to identify what all applications or their URLs are being supported by the firefox other than HTTP and HTTPs.
  - This is the result of a user using such applications before by adding additional handlers (like addon/extensions or supported utilities of the browser)
  - In this case, the IRC links can be supported by Firefox for this URL.
  - Copy the content of the handlers.json file and use the jsonformatter.curiousconcept.com to format it.
- Identify addons used by a user
  - Search for the addons.json file to start your investigation. This file contains all the details of every addon within their URLS, descriptions, addon name and a couple of other property info.
  - This is useful to identify whether:
    - User is using any privacy addon
    - User is using any proxy/VPN addon
    - User is using any file-sharing addon
    - User is using any pentest/hacking tool browser addon
  - In this case, the user is using HackBar addon, which is typically used for the hacking and pentesting purpose. If each behaviour is identified in the corporate laptop, it indicates suspicious behaviour.

Chrome

Location
- **Windows** — C:\Users\Chintan\AppData\Local\Google\Chrome\User Data\Default\
- **Linux** — ~/.config/google-chrome/default

History
- File: History.sqlite — Command: SELECT * from 'urls'
  - Identify what all URLs were visited by a user
  - Find suspicious/malicious URLs
  - Result
- File: History.sqlite — Command: SELECT * from 'downloads'
  - Find suspicious files were downloaded by a user
  - Find suspicious/malicious files downloaded
  - Find suspicious/malicious scripting file downloaded for further attack
  - Find any VPN/Proxy file downloaded
  - Find any executable downloaded on a system
  - Find any files which do not have a known filetypes.
  - History / Result
- File: History.sqlite — Command: SELECT * from 'keyword_search_terms'
  - Search for keyword searches
  - This is extremely helpful to identify user's behaviour pattern and his/her end goal of either stealing data/compromising data or any other malicious motive they carry.
  - Result
- File: Network Action Predictor.sqlite — Command: SELECT * from 'network_action_predictor'
  - Network Action Predictor: This event tracks user input keyword search character by character.
  - Result

Cookies
- File: Cookies.sqlite — Command: SELECT * from 'cookies'
  - Identify all cookies stored in the Chrome to investigate:
    - What all websites were being authenticated by a user?
    - When does the cookie expire for XYZ websites soon?
  - Result

Bookmarks
- File: Bookmarks — Command: Open the file in the notepad++ to view all the saved bookmarks in the Chrome
  - Identify all bookmarks saved in the Chrome
  - Bookmarks

Bookmarks (Firefox)
- Identify URLs from all bookmarks
  - Find suspicious/malicious URLs bookmarked
  - Identify date and time for suspicious URLs bookmarked
- Search for places.sqlite file to investigate all the bookmarks information
  - Run sqlite3 commands to view all bookmarks
    1. sqlite3 places.sqlite
    2. .tables
    3. select * from moz_bookmarks. It will display the result in this format
  - If you are using sqlite3 addon in your browser, then the result will look identical.
  - If you are using Linux terminal, then the result would look identical to this.

Convert epoch to human-readable date and vice versa
- For both types of result, there is a column - dateAdded.
- Sample value - 13055331226070000
- The value of this column is in the epoch format, and we need to convert this using online tool.
- Now we have the converted date and time in a human-readable format

History (Firefox)
- Find suspicious URL visit for:
  - Cloud file sharing websites (Dropbox, Megaupload, etc.)
  - Remote administration tool download website
  - Proxy, VPN websites
  - Identify torrent usage
  - Identify downloaded files through the firefox
  - etc.
  - A variety of categories can be involved in this search to identify malicious/suspicious activities of a user.
- Similarly to the bookmarks, this can be achieved by running the following query on the places.sqlite file. `select * from moz_places.`
  - Result
- Identify downloaded files through places.sqlite. `SELECT * from moz_annos.`
  - Result

Additional Information
- Important areas for investigation:
  - History
  - Cache
  - Cookies
  - Typed URLs
  - Sessions
  - Most visited sites
  - Screenshots
  - Financial info
  - Form values (Searches, Autofill)
  - Downloaded files (Downloads)
  - Favorites
  - Physical dumps of mobile devices
  - File systems of mobile devices
  - Backup of mobile devices
  - Data, which can be extracted from Clouds
  - Hard drives
  - Images of hard drives
  - Memory dumps
  - Hibernation and page files
- Find the Web Browsers artifacts

Tools
- **Hindsight** — https://github.com/obsidianforensics/hindsight
  - Hindsight is a free tool for analyzing web artifacts. It started with the browsing history of the Google Chrome web browser and has expanded to support other Chromium-based applications (with more to come!). Hindsight can parse a number of different types of web artifacts, including URLs, download history, cache records, bookmarks, autofill records, saved passwords, preferences, browser extensions, HTTP cookies, and Local Storage records (HTML5 cookies).
  - Every activity mentioned in this guide can be performed using single click on this software
- **RS Browser Forensics** — https://www.rsinsoft.net/web_browser_tools.html
- **Various tools by Nirsoft**

References
- https://www.digitalforensics.com/blog/an-overview-of-web-browser-forensics/
- https://www.hackers-arise.com/post/2016/11/17/digital-forensics-part-7-browser-forensics