## CIT 430/530: Forensic Activity #1.8 – Web Browser Reconstruction

A large percentage of investigations involve checking a suspect’s browser to for malicious activity. A suspect may use their browser to accomplish some unauthorized task. A computer forensics investigator can check browser activity for evidence or use what’s found as a source for creating a timeline of activity. In this activity you will reconstruct and investigate the Firefox browsing activities of a user.

**References**

* Dumpzilla - <https://www.dumpzilla.org/Manual_dumpzilla_en.txt>
* Webopedia - <https://www.webopedia.com/DidYouKnow/Internet/all_about_cookies.asp>
* w3schools.com - <https://www.w3schools.com/jsref/prop_doc_cookie.asp>

### Part 1 – Tool Setup

In this lab you’ll use the open source forensic browser tool Dumpzilla, which can be used to collect and extract relevant files and information such as Cookies from any user account on a system. Download and configure Dumpzilla, then update your python library.

**# wget** [**https://www.dumpzilla.org/dumpzilla.py**](https://www.dumpzilla.org/dumpzilla.py)

**# sudo apt-get install python3**

To complete the configuration, find the path for python to the dumpyzill.py file as shown below, then add the location to dumpzilla.py. *Note: When using the text editor nano, you only have to start typing at the appropriate location. Use ctrl+o to write and save your edits, then ctrl+x to exit back to the prompt*.

**# which python**

**# nano dumpzilla.py**



### Part 2 – Working with Dumpzilla

Dumpzilla works by sifting through data saved to the files under a user’s Mozilla browser profile (i.e. Firefox, Iceweasel and Seamonkey).

* Find the name of the user profile used by Firefox in your Kali vm, which ends in **.default** as shown in the image below.

**# cd ~/.mozilla/firefox**

**# ls -l**



Next, change back to the sansforensics home directory and issue the python3 command provided below.

**# cd ~**

**# python3 dumpzilla.py --Help**

The information returned is a list of commands and options available for dumpzilla. Run dumpzilla with the user profile name found in the **./mozilla/firefox** directory

**# python3 dumpzilla.py ~/.mozilla/firefox/######.default --All**

The resulting output should be a list of actions performed using Firefox, which may not return anything useful because of the limited usage of Firefox in this course.

### Part 3 – Web Browser Reconstruction with Dumpzilla

In Canvas, download the Suspect2.default.tar firefox user profile extracted from a different system. From the terminal, change to Downloads and issue the following command to uncompress the file.

**# cd Downloads**

**# tar -xvf Suspect2-2.default.tar**

In the following steps you will attempt to find data and create a timeline of activity performed by a user. Take note of the output returned from each command as this information should be included in your brief write-up at the end.

### Cookies

When a user visits a website, cookies are used to “identify users and possibly prepare customized webpages or save site login information.” Cookies are not intended to be harmful to a system, but can be an annoyance for a user (e.g. tracking of activity for targeted ads). For a forensics investigation, cookies can be helpful.

Using the commands below to see what cookies where collected by dumpzilla. Be sure to review all the output.

**# cd ~**

**# python3 dumpzilla.py ~/Downloads/Suspect2.default --Cookies**

**# python3 dumpzilla.py ~/Downloads/Suspect2.default --Cookies -showdom**

* **Total number cookies found**
  + **752**
* **Total number of DOM data found**
  + **225**
* **The significance of DOM data in relation to Cookies**
  + **Document object model which is part of the website request process. Dom data is what collects the code and holds the visited website information, compared to cookies which collects user preferences.**
* **How can DOM data and/or Cookies impact a forensic investigation?**
  + **Overall, it is about the correlation and the connection of the two that make up an internet activity of a user.**

### Bookmarks

Find a dumpzilla command from the --Help to return any Bookmark information for Suspect2.default profile.

* **Include the command used**
  + **python3 dumpzilla.py ~/Downloads/Suspect2.default -–Bookmarks**
* **What are the most recent Bookmarks? Be sure to note the timestamp information and sites returned in the output.** 
  + **Text

    Description automatically generated**

### Thumbnail Images

Find a dumpzilla command to return thumbnails.

* **Number of images found**
  + **0**

### Downloads

Find a dumpzilla command to view any downloaded data.

* **Include the command used**
  + **python3 dumpzilla.py ~/Downloads/Suspect2.default -–Downloads**
* **Number of downloads found**
  + **4**
* **Timestamp information, location of the downloads and the original source of the downloaded files.** 
  + **Root/Downlaods**
  + **Text

    Description automatically generated**

### User Input

Forms are commonly used in webpage development. They allow a user to enter information on a website which is then collected by a web server. Although the info is transmitted securely, it’s still captured on the user systems.

Find the appropriate option to return data entered by the suspect when they completed a form for making purchases.

* **Include the command used**
  + **python3 dumpzilla.py ~/Downloads/Suspect2.default -–Forms**
* **A timeline of activity that could be pertinent to an investigation, for example, attempting to use a credit card.** 
  + **89 forms**
  + **Luke Skywalker**
  + Text

    Description automatically generated
  + Trying to buy Kentucky wildcat seats
  + *Notes:*
    - *You may want to pipe the command output to less to ensure nothing is missed*
    - *The data listed in the ‘Value’ and ‘Name’ fields is where you should focus. Using the timestamp information, you can put together exactly when and what the user did.*

### Browser History

Use the ‘History’ option to view a complete list and timeline of activity performed by the suspect. **Visually search the entries for attempts to access any email accounts**. Piping output to the less command may be useful.

* **The number and type of emails accounts found**
  + **Total urls is 79**
  + **Gmail account, school email, amazon account**
* **Whether the user was able to successfully login to their email. If so, what mailbox was accessed and the activities performed**.
  + Text

    Description automatically generated

## Submission

Submit a completed copy of this document to Canvas by the due date. Screenshots are acceptable where appropriate.