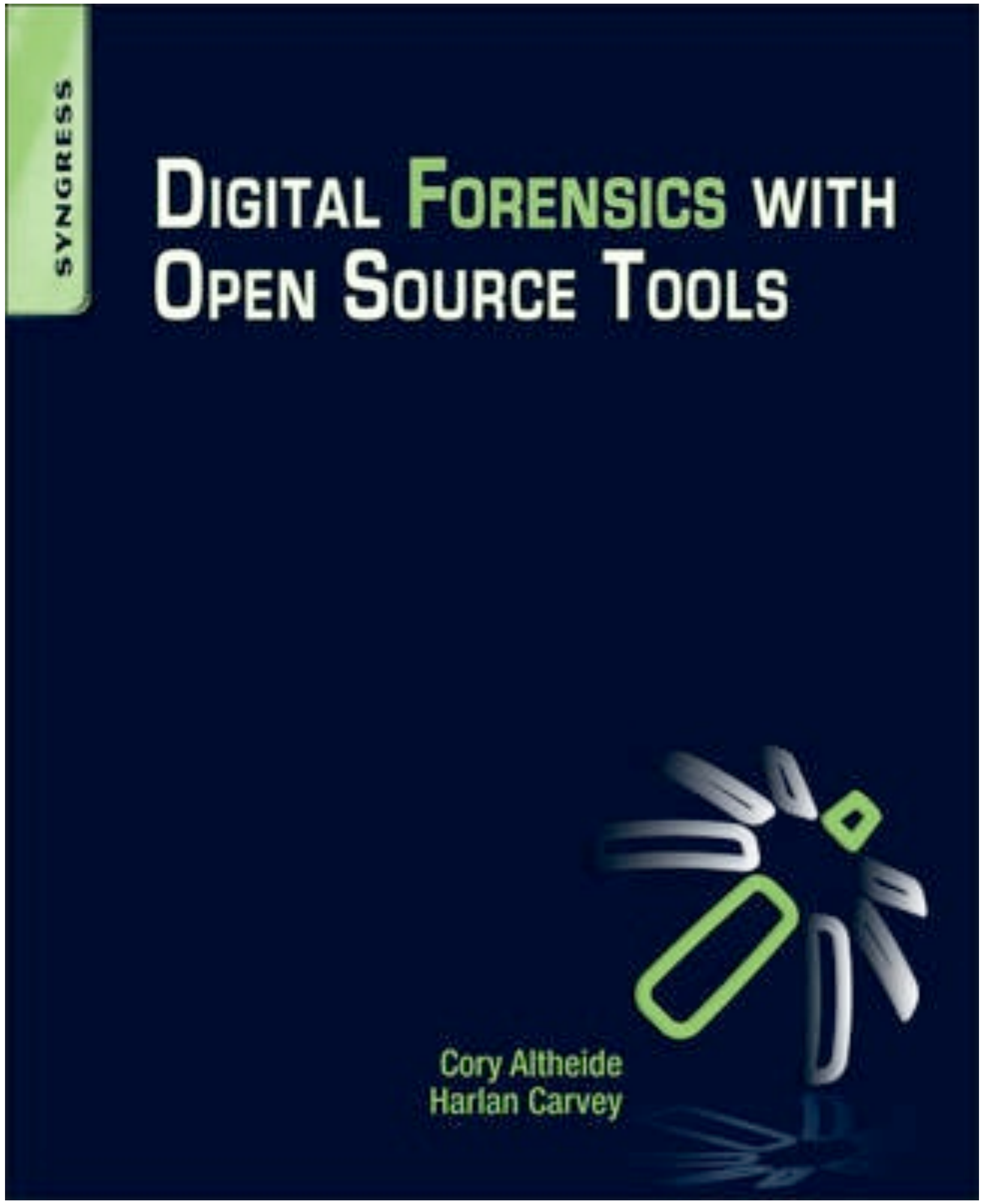
Making It Rain:
Examining Cloud Artifacts

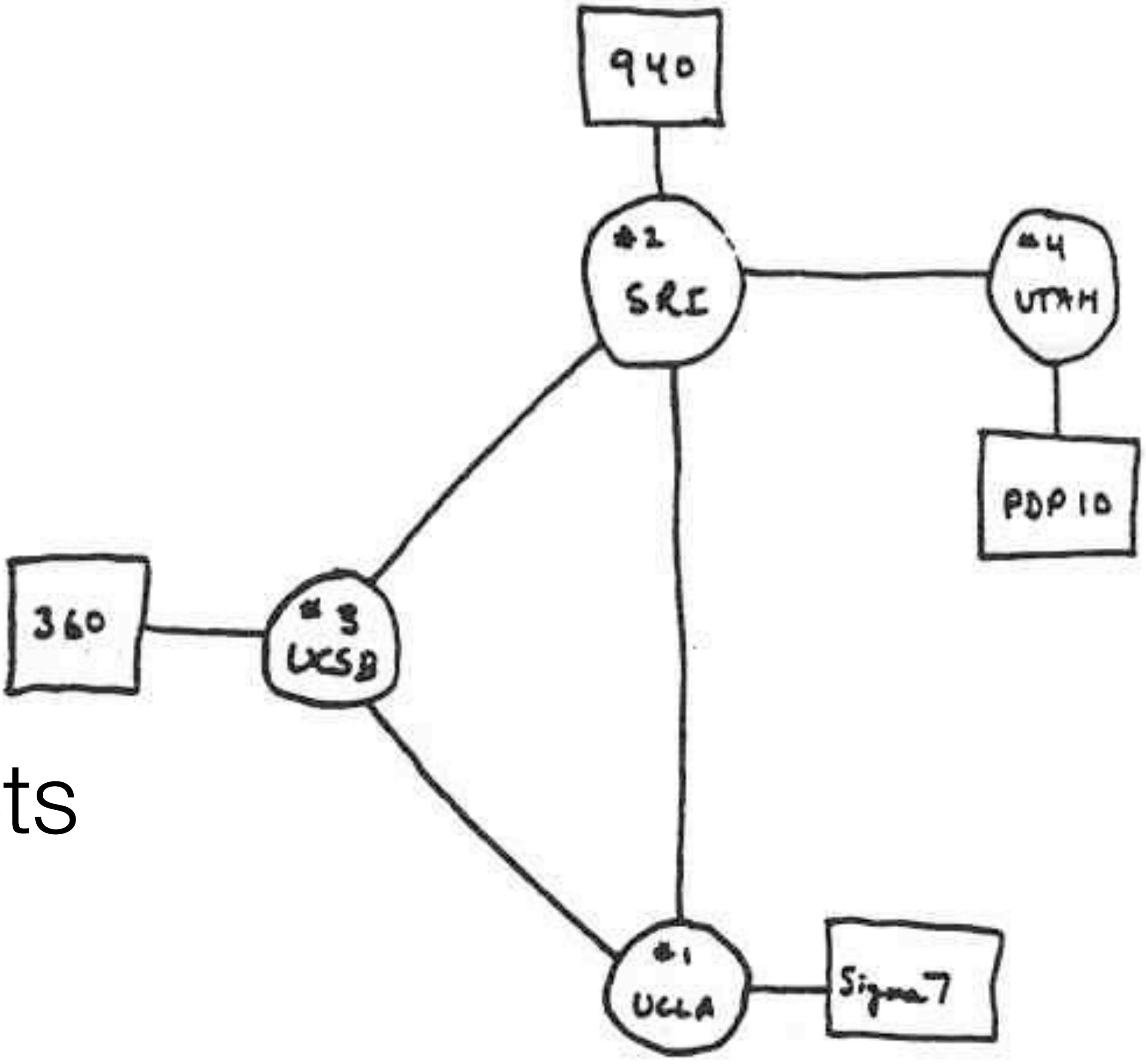
Cory Altheide

Who I Am

- Security Engineer @ Google, focused on Incident Response & Forensic Analysis
- Previously w/ MANDIANT, IBM XForce Emergency Response, US National Nuclear Security Administration
- Author of *Digital Forensics With Open Source Tools*.
- Winner of Honorary Forensic 4Cast Awards: "Nicest Beard, Most Self-Nominations"



Prehistoric Web Artifacts

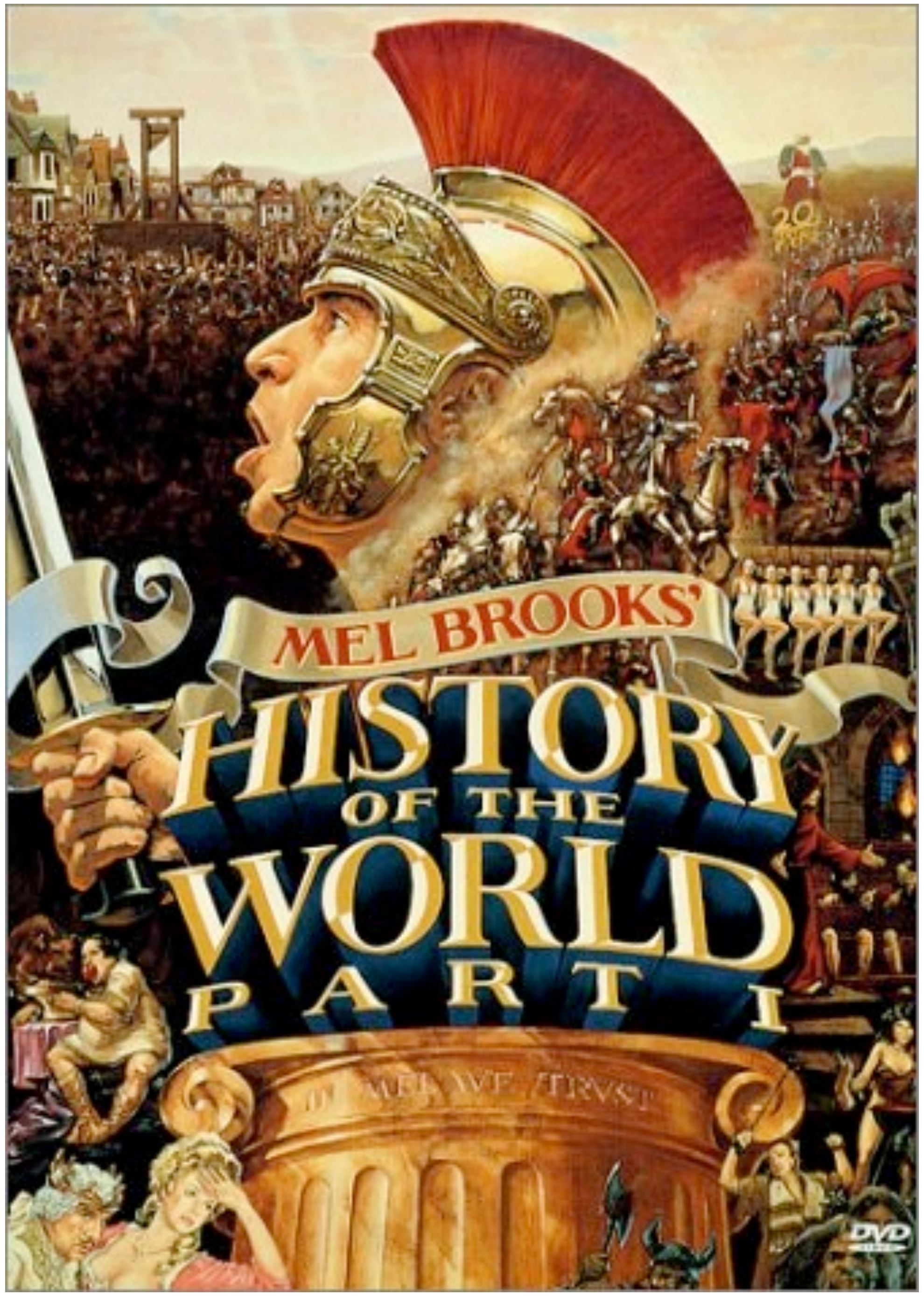


THE ARPA NETWORK

DEC 1969

4 NODES

History



- Who went where, when
- Basic universal information includes URL, time & date of visit
- Additional info includes visit counts, server status information, and more.
- This is, often, enough information.

Bookmarks

- User-generated shortcuts for a specific URL
- Basic information is... a URL.
- Can also include:
  - Page Title
  - Limited time information

** Image source courtesy Lee Whitfield's personal archive*

Cookies

- Delicious delicacies
- Also, small text files stored locally for:
    - maintaining state
    - authentication
    - other name:value pairs
- Contain information on domain issued for, c-time, and expiration

Modern Web Artifacts

HTML5

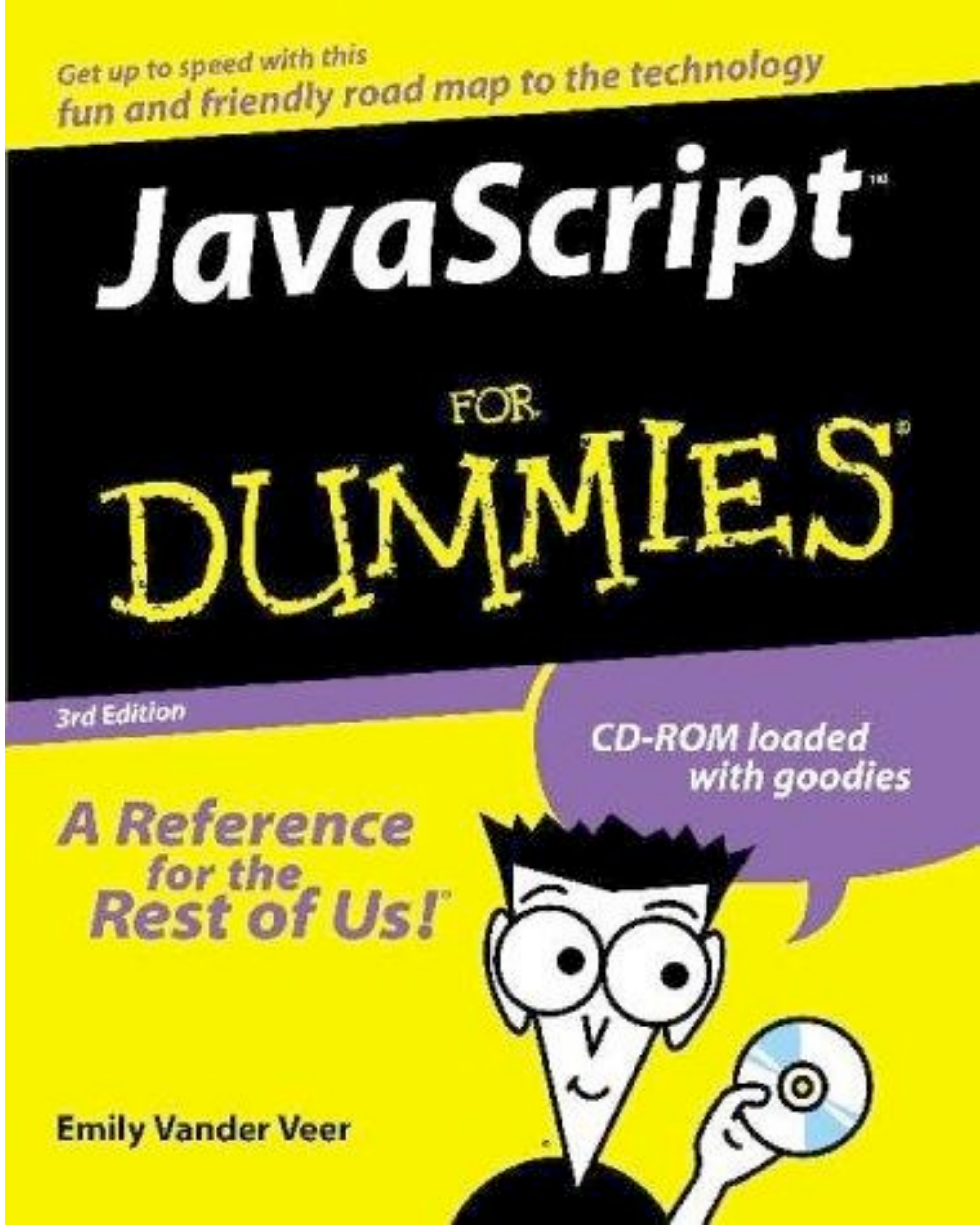
HTML5 Web Storage

- Rich Web Applications have increased local storage demands
- “Local Storage” for data that persists across sessions
- “Session Storage” for temporary data that is cached for one session
- Spec isn’t solid yet - different browsers handle this in different ways
- This will develop into an interesting data source

Javascript & JSON

Javascript

- *Lingua franca* of the modern web
- Heavily used in client-side applications
- Increasing use in server applications (*Node.js*)
- Some understanding of JavaScript is important
    - Client-side intrusions/exploits
    - Malware drive-by-downloads
    - Rich web application artifacts


Get up to speed with this fun and friendly road map to the technology
JavaScript™
FOR DUMMIES®
3rd Edition
CD-ROM loaded with goodies
A Reference for the Rest of Us!®
Emily Vander Veer

JSON

- “JavaScript Object Notation”
- Structured data interchange format used by many web applications
- Like XML-lite or “Human Readable Markup Readable By Real Humans”
- Local ephemeral artifacts for browsers and web applications will often be JSON objects
- Can view in text editor or in dedicated JSON parser/presenter:
    - *edit_json* in Ruby-JSON package, *jsonpipe* Python tool.

JSON

- “JavaScript Object Notation”
- Structured data interchange format used by many web applications
- Like XML-lite or “Human Readable Markup Readable By Real Humans”
- Local ephemeral artifacts for browsers and web applications will often be JSON objects
- Can view in text editor or in dedicated JSON parser/presenter:
    - *edit_json* in Ruby-JSON package, *jsonpipe* Python tool.

SQLite

SQLite in a Nutshell

- Simple, light, database-in-a-file
- Limited subset of SQL syntax
- Used in heavily in Webkit Browsers & Firefox for history data.
- Can process with *sqlite3* (CLI) or *sqliteman* (GUI)

SQLite slack

- SQLite databases grow, basically unbounded
- Removed rows/records will remain present, but unallocated, until overwritten
- VACUUM command compacts database, eliminating free space
- Until database is vacuum'd, old data may persist, can be carved.

Browsers

Microsoft Internet Explorer

Internet Explorer (8)

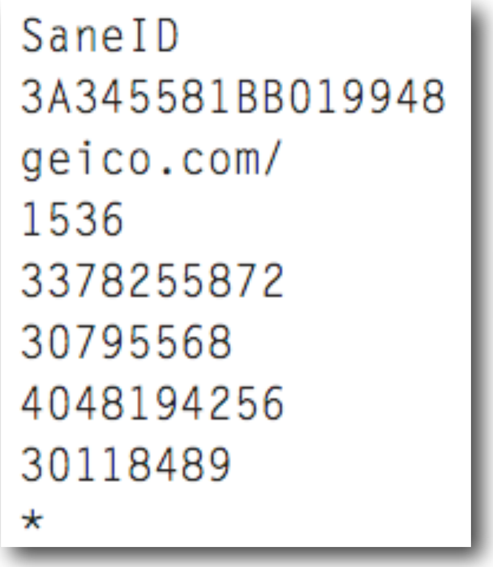
- Default browser on fresh Windows Install
- Primary use is for downloading Chrome (or Firefox, I guess).
- Shockingly enough, all local artifacts stored in goofy proprietary formats - “Microsoft Internet Explorer Cache File” (MSIECF)
- *{User}*\AppData\Local\Microsoft\Windows\Temporary Internet Files \Content.IE5*
- Parse these with Joachim Metz’s wonderful *libmsiecf*.
- Remember, you can’t spell “AAAIIIEEEEEEEEE!!!!” without “IE”

History - IE Cache Files

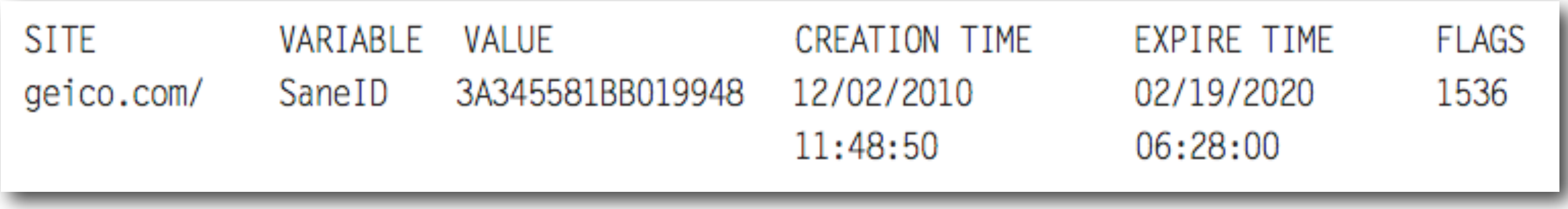
- Two different ‘index.dats’
  - Daily (UTC & Local Timestamps) & Weekly (Local only)
- 4 Record types:
  - URL: Contain URL, modified & accessed time, expiry, and response code
  - REDR: Indicate browser redirect
  - HASH: *<it is a mystery>*
  - LEAK: Attempted deletion while corresponding file is locked open.

Cookies

- AppData\Roaming\Microsoft\Windows\Cookies
- Discrete, plain text files per issuing host
- Date/times can (still) be parsed with *galleta*.



```
SaneID
3A345581BB019948
geico.com/
1536
3378255872
30795568
4048194256
30118489
*
```



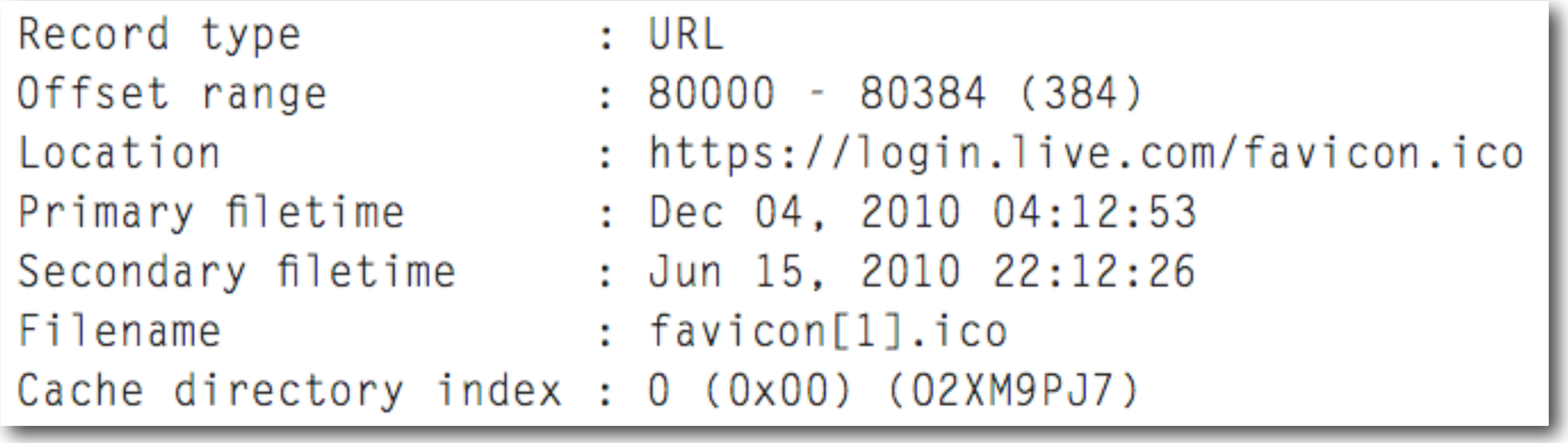
SITE	VARIABLE	VALUE	CREATION TIME	EXPIRE TIME	FLAGS
geico.com/	SaneID	3A345581BB019948	12/02/2010 11:48:50	02/19/2020 06:28:00	1536

Session Restore

- AppData/Local/Microsoft/Internet Explorer/Recovery/
    - Active/ - Last/Current Browsing Session
    - Last Active/ - Previous Browsing Session
    - RecoverStore.{*GUID*}.dat & {*GUID*}.dat
    - OLE Compound File Format (same as binary Office docs)
- ***This is an area of open research!***

Cache

- 4 randomly-named subdirectories of Content.IE5
- MSIECF “index.dat” file in Content.IE5 holds pointers to cached files
- Subdirectories contain cached files



```
Record type           : URL
Offset range          : 80000 - 80384 (384)
Location              : https://login.live.com/favicon.ico
Primary filetime      : Dec 04, 2010 04:12:53
Secondary filetime    : Jun 15, 2010 22:12:26
Filename              : favicon[1].ico
Cache directory index : 0 (0x00) (02XM9PJ7)
```

Mozilla Firefox

Firefox (4)

- Second most popular browser overall
- SQLite databases for nearly all relevant artifacts
- User profile location:
  - Win7: AppData\Roaming\Mozilla\Firefox\Profiles
  - Linux: .mozilla/firefox/Profiles
  - OS X: Library/Application Support/Firefox/Profiles
- {8 Random Characters}.default/

CLINT EASTWOOD
FIREFOX
The most devastating killing machine ever built. His job: steal it!

History - places.sqlite

- Most relevant tables:
    - moz_places: URL, page title, count
    - moz_historyvisits: “from_visit,” date, time, “visit_type”
        - Link, Typed, Bookmark, Embed, Redirect (Perm or Temp), Download
- Dates in “PRTime” - 64-bit microseconds since Unix Epoch

http://www.forensicswiki.org/wiki/Mozilla_Firefox_3_History_File_Format

Additional SQLite Artifacts

- formhistory.sqlite: saved form submission data
- downloads.sqlite: exactly what it sounds like
- webappstore.sqlite: HTML5 local database
- cookies.sqlite: ...cookies...

Bookmarks

- Stored in places.sqlite in three tables:
  - moz_bookmarks
  - moz_places
  - moz_items_annos
- Backups stored in “bookmarks-backups” directory as JSON objects

Cache

- One _CACHE_MAP_ & 3 cache files (*_CACHE_OO1_-_CACHE_OO3_*).

- 16 Subdirectories (0-F), with a number of additional subdirectories

- Randomly numbered files - local file copies.

- _CACHE_MAP_ & _CACHE_###_ files contain mappings between URLs & local cache files.

- Currently no open source tools to process these (**HINT**), but freeware Windows-only tools are available.

Session Restore

- *sessionrestore.js* is used to restore browsing session after crash
- Stored as series of JSON objects
- Items of note:
  - Closed tabs & windows
  - Saved form data
  - Temporary cookies

Google Chrome

Google Chrome (11)

- The best browser, basically.
- SQLite databases for nearly all relevant artifacts
- User profile location:
  - Win7: AppData\Local\Google\Chrome\default
  - Linux: .config/google-chrome/Default
  - OS X: Library/Application Support/Google/Chrome/Default

History

- Three main tables of interest:
  - downloads: downloaded files
  - urls: all visited URLs
  - visits: type of visit & time of visit

- ‘urls’ & ‘visits’ combine to generate most of our “history” data.

History - visits (transition row) - partial

- LINK: Clicked a link
- TYPED: Typed in URL bar.
- AUTO_BOOKMARK: Through UI suggestion
- AUTO_SUBFRAME: Content automatically loaded in a non-toplevel frame.
- MANUAL_SUBFRAME: Subframe explicitly requested by user
- FORM_SUBMIT: User filled out values in a form and submitted
- RELOAD: User reloaded the page

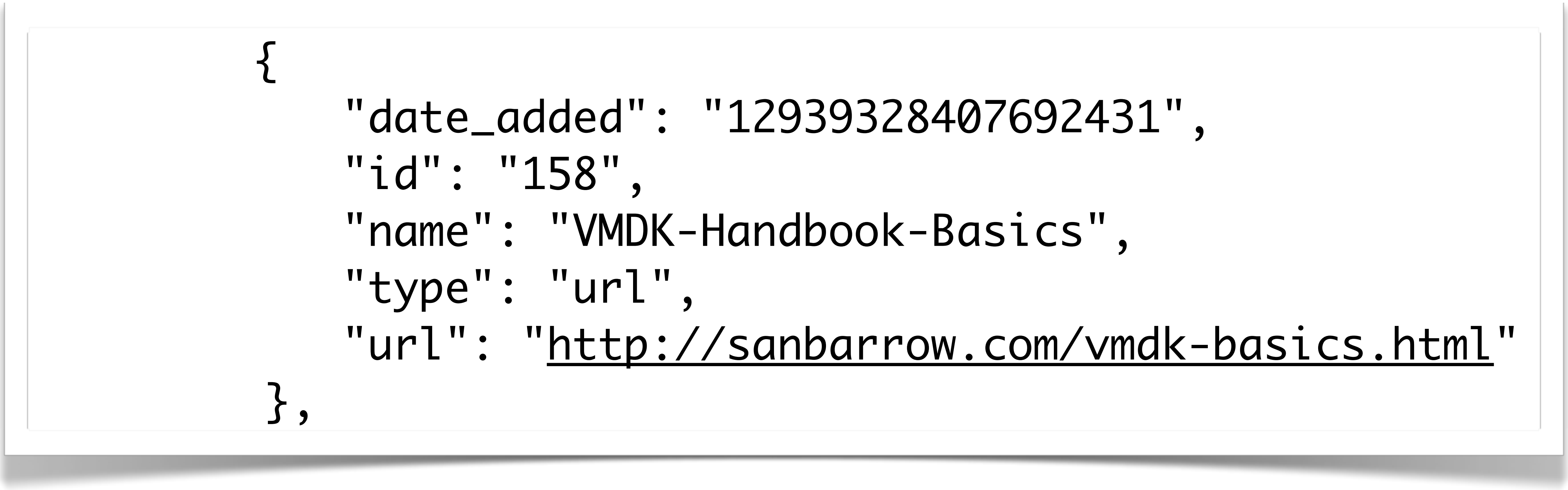
http://computer-forensics.sans.org/blog/2010/01/21/google-chrome-forensics/

Other SQLite Artifacts

- History Index *{YEAR-MO}*: Archived History files, (*indefinitely*)?
- Web Data: Saved form auto-fill data
- Thumbnails: stored thumbnail images of visited pages
- Cookies: contains... cookies

Bookmarks

- Sequence of JSON objects
- Entries contain data added



```
        {
            "date_added": "12939328407692431",
            "id": "158",
            "name": "VMDK-Handbook-Basics",
            "type": "url",
            "url": "http://sanbarrow.com/vmdk-basics.html"
         },
```

Local State

- Chrome’s browsing session restore mechanism
- Series of JSON objects
- Can include form data, closed tabs & windows

Apple Safari

Safari (5)

- Default browser on OS X
- Nobody else uses this
- User profile location:
  - Win7: AppData\Roaming\Apple Computer\Safari
  - OS X: Library/Safari
- Most data stored in (drum roll) binary property lists
- Use *plutil* (on OS X), or *plutil.pl*, or *Safari Forensics Tools* (jafat.sf.net)

Property Lists

- A quick note about property lists:
  - Two flavors: plain text (XML) and binary XML (blech)
  - Heavily used for OS X configuration
  - Also used in Safari on Windows

History.plist

- Main Safari “History” file.

- Stores URL, Last Visit Date/Time, Number of Visits, Page Title.

- Raw time is “CFAbsoluteTime” - number of seconds since Jan 1 2001

Everything Else

- Downloads.plist: holds information on downloaded files - URL, size, and path
- Bookmarks.plist: Bookmarks, just title and URL
- *Cookies.plist: Domain, time, key:value*
- TopSites.plist: User’s Safari home screen hot list
- Webpage Previews/: large image captures of pages visited
- LocalStorage/: directory containing SQLite databases for HTML5 localstorage
- *HistoryIndex.sk: no one knows. Prime research opportunity!*

Cache

- Stored in “Cache.db” SQLite database
- Two main tables:
  - cfurl_cache_response: URL & Request metadata
  - cfurl_cache_blob_data: cached data
- Even when “emptied,” database is not “vacuumed” - entries and cached data remain

LastSession.plist

- Used to restore browser state
- URLs & page titles can be recovered
- No form data
- No closed windows/tabs (as far as I have seen)

Safari Forensics Tools

- *safari_hist*: History.plist
- *safari_download*: Download.plist
- *safari_cookies*: Cookies.plist
- *safari_bm*: Bookmarks.plist
- *pref_parser*: any other binary plist.

the end

cory@google.com