Casey Garrett, Research notes and references, CS 445 project Social media post

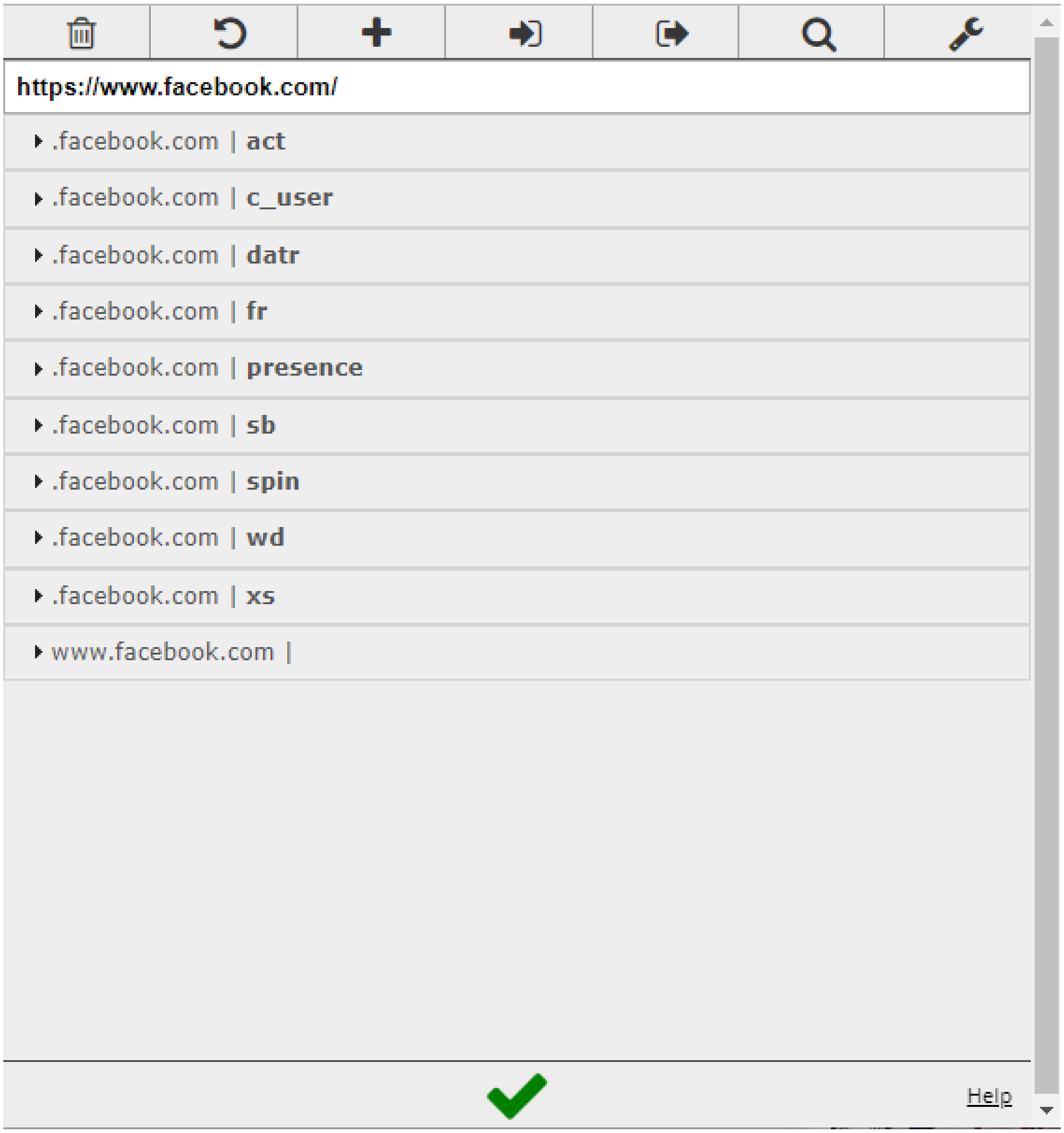
Date: 4/5/2019

Task: What social media has the 'easiest' cookies to understand and possible recreate

I found some google extensions which can allow me to look at the cookies of a website. Of the basic social media platforms, the heavy hitters are Facebook, twitter, and Instagram. Instagram while it does have an online web service it mostly exists on mobile devices which are far too complicated for me to attempt.

First a look at Facebook cookies:

It has 10 different cookie values that the extension has found: act, c_user, datr, fr, presence, sb, spin, wd, xs, and a blank value field.

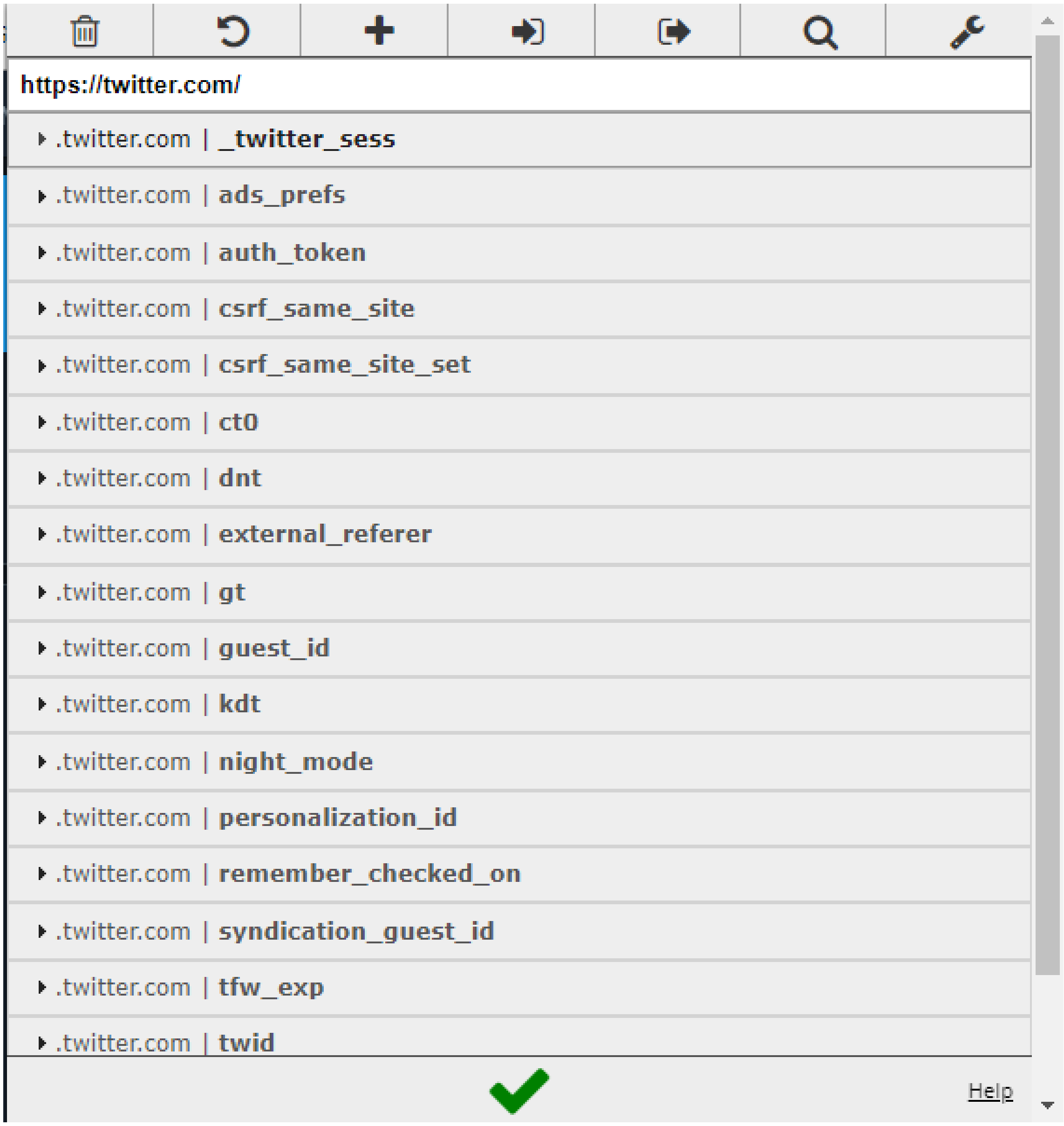


https://www.facebook.com/

- .facebook.com | act
- .facebook.com | c_user
- .facebook.com | datr
- .facebook.com | fr
- .facebook.com | presence
- .facebook.com | sb
- .facebook.com | spin
- .facebook.com | wd
- .facebook.com | xs
- www.facebook.com |

Next a look at twitter's cookies:

Twitter has a lot more cookies and since I have the screencap no point in listing them out



https://twitter.com/

.twitter.com | _twitter_sess

.twitter.com | ads_prefs

.twitter.com | auth_token

.twitter.com | csrf_same_site

.twitter.com | csrf_same_site_set

.twitter.com | ct0

.twitter.com | dnt

.twitter.com | external_referer

.twitter.com | gt

.twitter.com | guest_id

.twitter.com | kdt

.twitter.com | night_mode

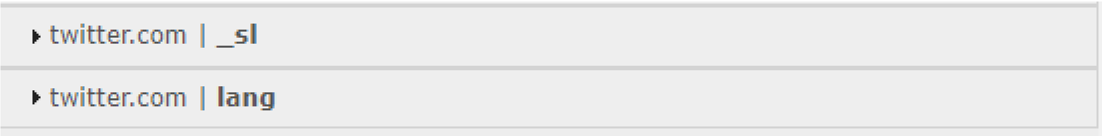
.twitter.com | personalization_id

.twitter.com | remember_checked_on

.twitter.com | syndication_guest_id

.twitter.com | tfw_exp

.twitter.com | twid



twitter.com | _sl

twitter.com | lang

It seems that Facebook is the way to go but I did ask another classmate who has already made more progress on the project and he recommend twitter complete. So for now I'll deal with twitter.

Date: 4/14/2019

Task: Where are cookies stored in Linux for Firefox.

Since I already have two Kali machine I'm going to be using them for the rest of the project, unless I run into any problems with assignments and this project conflicting.

Found that the cookies are stored in an SQLite file called cookies located:

cd ~/.mozilla/firefox/<profile path>/

For me since I do not have a Firefox account the profile path was an odd name: f0y9xdf4.default

In the folder there are 3 cookies files:

Cookies.sqlite & cookies.sqlite-shm & cookies.sqlite-wal

Source for information:

https://unix.stackexchange.com/questions/82597/where-does-firefox-store-its-cookies-on-linux

Installed sqlite on my kali machine. However, if I simply cat the cookies file its non-human readable. I searched for quite a few days to find something that can help me print/parse/and read the cookies file to no available. Most likely cause I'm not the best with SQlite/SQL database things.

Date: 4/23/2019

Task: Grab the data in my linux cookies file and use this to login to twitter

Running out of time I cannot find a simple way that works with my knowledge base of SQlite, and the encrypted data that is held in the cookies file. Next best thing would be to try and get as much information as I can and show how I am able to get 'some' (hopefully) of the cookie data and show progress to constructing my own cookie or http request to auto login to twitter with no authentication.

Looks like the best bet for seeing the cookie data without having to decrypt the data (somehow) is using Python and do a http request using my login information and see the returned data.

Data: 5/1/2019

Task: Python code to get cookies form twitter

Below is the code I committed to my repo for requesting login access to https://twitter.com/login

```
from requests import session

f = open("header.txt", "a")


payload = {
	'action': 'login',
	'username': '7753504700',
	'password': 'This_Woould_be_my_real_pass_but_That's mine'
}

with session() as c:
	c.post('https://twitter.com/login', data=payload)
	response = c.get('https://twitter.com/login')
	f.write(repr(response.headers))
	f = open("response.txt", "a")
	f.write(repr(response.text))

f.close()
```

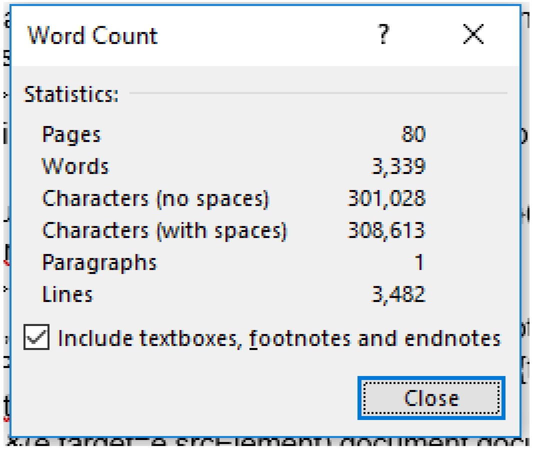
This code gathers the response header and response text the header is filled with this information:

First the Header File:

```
{'status': '200 OK', 'x-response-time': '30', 'content-length': '49480', 'strict-transport-security':
'max-age=631138519', 'x-twitter-response-tags': 'BouncerCompliant', 'x-transaction':
'0005258500eb3819', 'x-content-type-options': 'nosniff', 'content-encoding': 'gzip', 'set-cookie':
'fm=0; Max-Age=0; Expires=Sat, 1 May 2019 19:00:11 GMT; Path=/; Domain=.twitter.com;
Secure; HTTPOnly,
_twitter_sess=BAh7CSIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6OkZsYXNo
%250ASGFzaHsABjoKQHVzZWR7ADoPY3JlYXRlZF9hdGwrCLtVRKhqAToMY3NyZl9p%250A
ZCIlYTk0NmEyMGQ0OGM2MWNiYmZiN2U5YjYxZjMxM2ZjMzM6B2lkIiVhMjhl%250AN2JjZW
NmMDM4YWM5MmQ3NWQ1MjRkMjMzNzA2ZQ%253D%253D--
031cb83d250b2ad11b72eef6317a4215ef07fd65; Path=/; Domain=.twitter.com; Secure;
HTTPOnly', 'expires': 'Tue, 31 Mar 1981 05:00:00 GMT', 'server': 'tsa_a', 'last-modified': 'Sat, 1
May 2019 19:00:11 GMT', 'x-xss-protection': '1; mode=block;
report=https://twitter.com/i/xss_report', 'x-connection-hash':
'93b8addb835d3ac7db397f18a663457e', 'x-ua-compatible': 'IE=edge,chrome=1', 'pragma': 'no-
cache', 'cache-control': 'no-cache, no-store, must-revalidate, pre-check=0, post-check=0', 'date':
'Sat, 1 May 2019 19:00:11 GMT', 'x-frame-options': 'DENY', 'content-type': 'text/html;charset=utf-
8'}
```

The Response file is massive and would be pointless to include it, so here is a screenshot of the word cout in MS Word, and the first page of information:

Casey Garrett, Research notes and references, CS 445 project Social media post



Statistics:

Pages	80
Words	3,339
Characters (no spaces)	301,028
Characters (with spaces)	308,613
Paragraphs	1
Lines	3,482

☑ Include textboxes, footnotes and endnotes

```
u'<!DOCTYPE html>\n<html lang="en" data-scribe-reduced-action-queue="true">\n  <head>\n
\n    \n    \n    \n    \n    \n    \n    <meta charset="utf-8">\n      <script
nonce="0n0qk49P4ENAmzTI2yUsGQ==">\n
!function(){window.initErrorstack||(window.initErrorstack=[]),window.onerror=function(r,i,n,o,t){r.i
ndexOf("Script error.")>-
1||window.initErrorstack.push({errorMsg:r,url:i,lineNumber:n,column:o,errorObj:t})}}();\n
</script>\n    \n    \n  \n  <script id="bouncer_terminate_iframe"
nonce="0n0qk49P4ENAmzTI2yUsGQ==">\n    if (window.top != window) {\n
window.top.postMessage({\'bouncer\': true, \'event\': \'complete\'}, \'*\');\n}\n  </script>\n  <script
id="swift_action_queue" nonce="0n0qk49P4ENAmzTI2yUsGQ==">\n    !function(){function
e(e){if(e||(e=window.event),!e)return!1;if(e.timestamp=(new
Date).getTime(),!e.target&&e.srcElement&&(e.target=e.srcElement),document.documentEleme
nt.getAttribute("data-scribe-reduced-action-queue"))for(var
t=e.target;t&&t!=document.body;){if("A"==t.tagName)return;t=t.parentNode}return
i("all",o(e)),a(e)?(document.addEventListener||(e=o(e)),e.preventDefault=e.stopPropagation=e.s
topImmediatePropagation=function(){},y?(v.push(e),i("captured",e)):i("ignored",e),!1):(i("direct",e
),!0)}function t(e){n();for(var t,r=0;t=v[r];r++){var
a=e(t.target),i=a.closest("a")[0];if("click"==t.type&&i){var
o=e.data(i,"events"),u=o&&o.click,c=!i.hostname.match(g)||!i.href.match(/#$/);if(!u&&c){window.l
ocation=i.href;continue}}a.trigger(e.event.fix(t))}window.swiftActionQueue.wasFlushed=!0}functi
on r(){for(var e in b)if("all"!=e)for(var
t=b[e],r=0;r<t.length;r++)console.log("actionQueue",c(t[r]))}function n(){clearTimeout(w);for(var
e,t=0;e=h[t];t++)document["on"+e]=null}function a(e){if(!e.target)return!1;var
t=e.target,r=(t.tagName||"").toLowerCase();if(e.metaKey)return!1;if(e.shiftKey&&"a"==r)return!1;i
f(t.hostname&&!t.hostname.match(g))return!1;if(e.type.match(p)&&s(t))return!1;if("label"==r){var
n=t.getAttribute("for");if(n){var a=document.getElementById(n);if(a&&f(a))return!1}else for(var
i,o=0;i=t.childNodes[o];o++)if(f(i))return!1}return!0}function i(e,t){t.bucket=e,b[e].push(t)}function
o(e){var t={};for(var r in e)t[r]=e[r];return t}function
u(e){for(;e&&e!=document.body;){if("A"==e.tagName)return e;e=e.parentNode}}function c(e){var
t=[];e.bucket&&t.push("["+e.bucket+"]"),t.push(e.type);var
r,n,a=e.target,i=u(a),o="",c=e.timestamp&&e.timestamp-
d;return"click"===e.type&&i?(r=i.className.trim().replace(/\\s+/g,"."),n=i.id.trim(),o=/[^#]$/.test(i.
href)?" ("+i.href+")":"",a=\'"\'+i.innerText.replace(/\\n+/g,"
").trim()+\'"\'):(r=a.className.trim().replace(/\\s+/g,"."),n=a.id.trim(),a=a.tagName.toLowerCase()
,e.keyCode&&(a=String.fromCharCode(e.keyCode)+" :
"+a)),t.push(a+o+(n&&"#"+n)+(!n&&r?"."+r:"")),c&&t.push(c),t.join(" ")}function f(e){var
t=(e.tagName||"").toLowerCase();return"input"==t&&"checkbox"==e.getAttribute("type")}function
s(e){var
t=(e.tagName||"").toLowerCase();return"textarea"==t||"input"==t&&"text"==e.getAttribute("type")||
"true"==e.getAttribute("contenteditable")}for(var m,d=(new
Date).getTime(),l=1e4,g=/^([^\\.]+\\.)*twitter\\.com$/,p=/^key/,h=["click","keydown","keypress","ke
yup"],v=[],w=null,y=!0,b={captured:[],ignored:[],direct:[],all:[]},k=0;m=h[k];k++)document["on"+m]
=e;w=setTimeout(function(){y=!1},l),window.swiftActionQueue={buckets:b,flush:t,logActions:r,wa
sFlushed:!1}}();\n  </script>\n  <script id="composition_state"
```

Data: 5/11/2019

Task: Final thoughts.

So overall this project ended up being vast majority research and a micro code which did put me in the right direction, but due to time constraint and overall road blocks for the project this is were I call it. Also, Iman sorry for the crappy single commit I thought there was no real reason for me to commit a document like this hence the dates and overall journal theme to this document.