Aaron’s final program. A formal approach to the analysis of software archives for the social and historical sciences.

Camille Akmut
January 14, 2020

Abstract
A working version of keepgrabbing.py was reconstructed from archives.
Software as archives

In writing Aaron Swartz’s JSTOR code we did not merely attempt a “technical” presentation, but sought to give a full “sociological, legal and historical” account – as part of our efforts to extend both the domain of one and the other science. These sciences meet now halfway, or they are the same.

These debates are far from settled.

Written over a period of a few weeks, though a long time in the coming¹, in multiple iterations, we drew primarily from 3 documents:

— The source code of the program (entitled keepgrabbing.py).
— USA v. Swartz (hundreds if not thousands of pages of collected legal and technical prose²).
— The Report to the President of MIT, ordered by the latter institution and written by computer scientist Hal Abelson inside of a larger team³.

Our efforts took us so far that we ended up reconstructing a working version;

From a careful analysis – as good as we could, anyway – followed chaotic consequences.

But, crucially, this reconstruction, while of strong scientific interest, was not able to accomplish anything with the inputs we gave to it, exempli gratia, the titles of two texts by Marc Bloch and Yitang Zhang.

Let alone that the infrastructure at JSTOR must have considerably changed since the last decade.

In any case, the moneyed interests of this institution cannot trump those of research, and the quest for knowledge and truth.

The main difficulty consisted in identifying the inputs passed to the lambda function (encapsulated in line) in the absence of the redacted passage of code.

But, this could be reversed engineered thanks to even summary knowledge of the Python standard library – as long as paired with a knowledge of functional programming.

The .split() function was key.

```python
>>> ['regions', 'landau'].split()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'list' object has no attribute 'split'
```

It only accepts certain types of objects. From there the rest could be derived.

¹This research had for antecedent the relevant sections in “A User-Defined Web” (2019).
²We have made use of one document in particular: the so-called “Superseding indictment” from Sep. 2012. Others are encouraged to go through the rest – these inhumanely numerous pages are the new archives, of this new history and sociology.
³We presume it was Abelson who wrote, or made the biggest contributions to the more technical “Part 1” and connected apparatus (e.g. inventory of terms found at the end).
The redacted passages point to a website with the ending /grab, of which we have presumed so far it was created by Aaron. But, as we argue, even with no knowledge of this portion of the code, a black box, a correct or even partial understanding of the anonymous function \texttt{lambda x} was sufficient.

To be clear, \texttt{keepgrabbing.py} was not Aaron Swartz’s last program\textsuperscript{4}. But, it was the one that had the most effect in defining the next three years of his life, from 2010 to January 2013.

Aaron Swartz had lived in Cambridge, MA\textsuperscript{5}, but moved away presumably as a result of the events, and hanged himself in Brooklyn, NY\textsuperscript{6}.

We simplified the program, but did not lessen its functionalities:
Instead of receiving its inputs from the Internet/Web (via the Python module \texttt{urllib}), here it does so from a list, the same list it would have received, or close enough.

\begin{verbatim}
def getblocks():
    r = "regions_and_social_groups On_the_Landau-Siegel_Zeros_Conjecture"

    Simplying and experimenting, we cannot stress this enough, remain some of the most powerful tools in our equivalent of the archaeologist’s toolkit.

A diagram to complete this discussion:
\end{verbatim}

\begin{verbatim}
http://--------/grab
r = "article1 article2 article3 ..."

| r.split() |
| ['article1', 'article2', 'article3'] |
| getblocks() |
blocks = getblocks() |
| line(block) |
| line('article1')
\end{verbatim}

\textsuperscript{4}See our history of his time spent on Github.
\textsuperscript{5}See Abelson et al., beginning of part 2 of their report.
\textsuperscript{6}His partner has given an account of their last days.
| lambda 'article1'
| $ curl www.jstor.org/.../article1.pdf
| repeat with 'article2' etc.

What follow are the last fragments of these experiments.
Annex

– 1 : raw code (as found about in archives i.e. the legal documents)
– 2 : working version (the redacted part is bypassed as well as the use of proxy, we assume the simplest scenario) do not execute!
– 3 : this version is run successfully for the first time (after multiple tries and modifications resulting in 2)
– 4 : this version is run a second time by accident (inside the Python shell)
– 5 : a second version for local testing is created
– 6 : output of the second version : the x to the lambda function is as intended and the rest too
#!/usr/bin/python
import subprocess, urllib, random
class NoBlocks(Exception): pass
def getblocks():
    r = urllib.urlopen("http://--------/grab").read()
    if '<html' in r.lower(): raise NoBlocks
    return r.split()

import sys
if len(sys.argv) > 1:
    prefix = ['--socks5', sys.argv[1]]
else:
    prefix = ['#'-interface','eth0:1']
line = lambda x: ['curl'] + prefix + ['-H', "Cookie: TENACIOUS=" +
str(random.random())[3:], '-o', 'pdfs/' + str(x) + '.pdf', "http://www.jstor.org/
stable/pdfplus/" + str(x) + ".pdf?acceptTC=true"]

while 1:
    blocks = getblocks()
    for block in blocks:
        print block
        subprocess.Popen(line(block)).wait()
import subprocess, random #urllib,
#class NoBlocks(Exception): pass
def getblocks():
    r = "regions_and_social_groups On the Landau-Siegel_Zeros Conjecture"
    urllib.urlopen("http://--------/grab").read()
    if '<html' in r.lower(): raise NoBlocks
    return r.split()

# import sys
# if len(sys.argv) > 1:
#     prefix = ['--socks5', sys.argv[1]]
# else:
#     prefix = []
#     'interface','eth0:1'
line = lambda x: ['curl'] + prefix + ['-H', 'Cookie: TENACIOUS=' +
    str(random.random())[3:], '-o', 'pdfs/' + str(x) + '.pdf',
    "http://www.jstor.org/stable/pdfplus/" + str(x) + "pdf?acceptTC=true"

while 1:
    blocks = getblocks()
    for block in blocks:
        print block
        subprocess.Popen(line(block)).wait()
$ torsocks python python.py
regions_and_social_groups

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
     0      0     0      0        0     0     0     0 --:--:-- --:--:-- --:--:--
     0      0     0      0     0:00:01 --:--:--

Traceback (most recent call last):
  File "python.py", line 21, in <module>
    subprocess.Popen(line(block)).wait()
  File "/usr/lib/python2.7/subprocess.py", line 1099, in wait
    pid, sts = _eintr_retry_call(os.waitpid, self.pid, 0)
File "/usr/lib/python2.7/subprocess.py", line 125, in _eintr_retry_call
    return func(*args)
KeyboardInterrupt
import python.py
regions_and_social_groups

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/regions_and_social_groups.pdf: No such file or directory
100 117 100 117 0 0 319 0 -->:--:--:--:--:--:--:-- 318

curl: (23) Failed writing body (0 != 117)

On_the_Landau-Siegel_Zeros_Conjecture

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/On_the_Landau-Siegel_Zeros_Conjecture.pdf: No such file or directory
100 129 100 129 0 0 1535 0 -->:--:--:--:--:--:--:-- 1535

curl: (23) Failed writing body (0 != 129)

regions_and_social_groups

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/regions_and_social_groups.pdf: No such file or directory
100 117 100 117 0 0 1300 0 -->:--:--:--:--:--:--:-- 1300

curl: (23) Failed writing body (0 != 117)

On_the_Landau-Siegel_Zeros_Conjecture

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/On_the_Landau-Siegel_Zeros_Conjecture.pdf: No such file or directory
100 129 100 129 0 0 1402 0 -->:--:--:--:--:--:--:-- 1402

curl: (23) Failed writing body (0 != 129)

regions_and_social_groups

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/regions_and_social_groups.pdf: No such file or directory
100 117 100 117 0 0 1026 0 -->:--:--:--:--:--:--:-- 1026

curl: (23) Failed writing body (0 != 117)

On_the_Landau-Siegel_Zeros_Conjecture

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/On_the_Landau-Siegel_Zeros_Conjecture.pdf: No such file or directory
100 129 100 129 0 0 1343 0 -->:--:--:--:--:--:--:-- 1343

curl: (23) Failed writing body (0 != 129)

regions_and_social_groups

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--

Warning: Failed to create the file pdfs/regions_and_social_groups.pdf: No such file or directory
100 117 100 117 0 0 1218 0 -->:--:--:--:--:--:--:-- 1218

curl: (23) Failed writing body (0 != 117)

On_the_Landau-Siegel_Zeros_Conjecture

% Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
          Download Upload  Total   Spent    Left  Speed
0 0 0 0 0 0 0 -->:--:--:--:--:--:--:--
Warning: Failed to create the file
Warning: pdfs/On_the_Landau-Siegel_Zeros_Conjecture.pdf: No such file or
directory
100 129 100 129 0 0 1290 0 --:--:-- --:--:-- --:--:-- 1290
curl: (23) Failed writing body (0 != 129)
regions_and_social_groups
% Total  % Received % Xferd Average Speed Time Time Time Current
Download Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:--
0^CTraceback (most recent call last):
  File "<stdin>"", line 1, in <module>
  File "python.py", line 21, in <module>
    subprocess.Popen(line(block)).wait()
  File "/usr/lib/python2.7/subprocess.py", line 1099, in wait
    pid, sts = _eintr_retry_call(os.waitpid, self.pid, 0)
  File "/usr/lib/python2.7/subprocess.py", line 125, in _eintr_retry_call
    return func(*args)
KeyboardInterrupt
>>>
import subprocess, random #urllib,
#class NoBlocks(Exception): pass

def getblocks():
    r = "regions_and_social_groups On_the_Landau-
Siegel_Zeros_Conjecture"#urllib.urlopen("http://--------/grab").read()
    #if '<html' in r.lower(): raise NoBlocks
    return r.split()

#import sys
#if len(sys.argv) > 1:
    # prefix = ['--socks5', sys.argv[1]]
#else:
    prefix = []
    line = lambda x: ['curl'] + prefix + ['-H', "Cookie: TENACIOUS="]
    str(random.random())[3:], '-o', 'pdfs/' + str(x) + '.pdf', "http://www.jstor.org/" + str(x) + "?acceptTC=true"

#while 1:
blocks = getblocks()
for block in blocks:
    print block
for block in blocks:
    print(line(block))
    #subprocess.Popen(line(block)).wait()
$ python python2.py
regions_and_social_groups
On_the_Landau-Siegel_Zeros_Conjecture
['curl', '-H', 'Cookie: TENACIOUS=39548224339', '-o', 'pdfs/
regions_and_social_groups.pdf', 'http://www.jstor.org/stable/pdfplus/
regions_and_social_groups.pdf?acceptTC=true']
['curl', '-H', 'Cookie: TENACIOUS=7764535213', '-o', 'pdfs/On_the_Landau-
Siegel_Zeros_Conjecture.pdf?acceptTC=true']