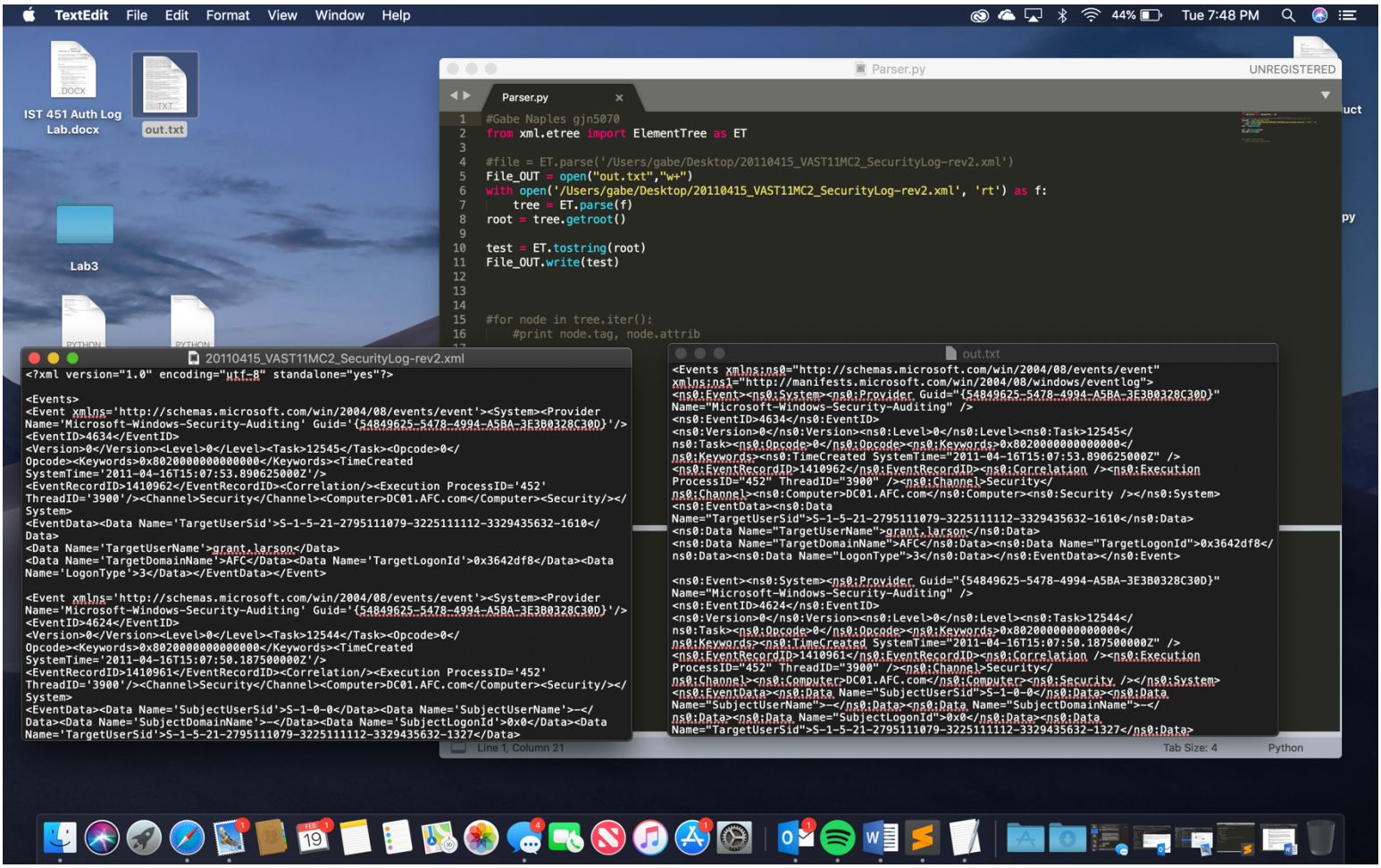
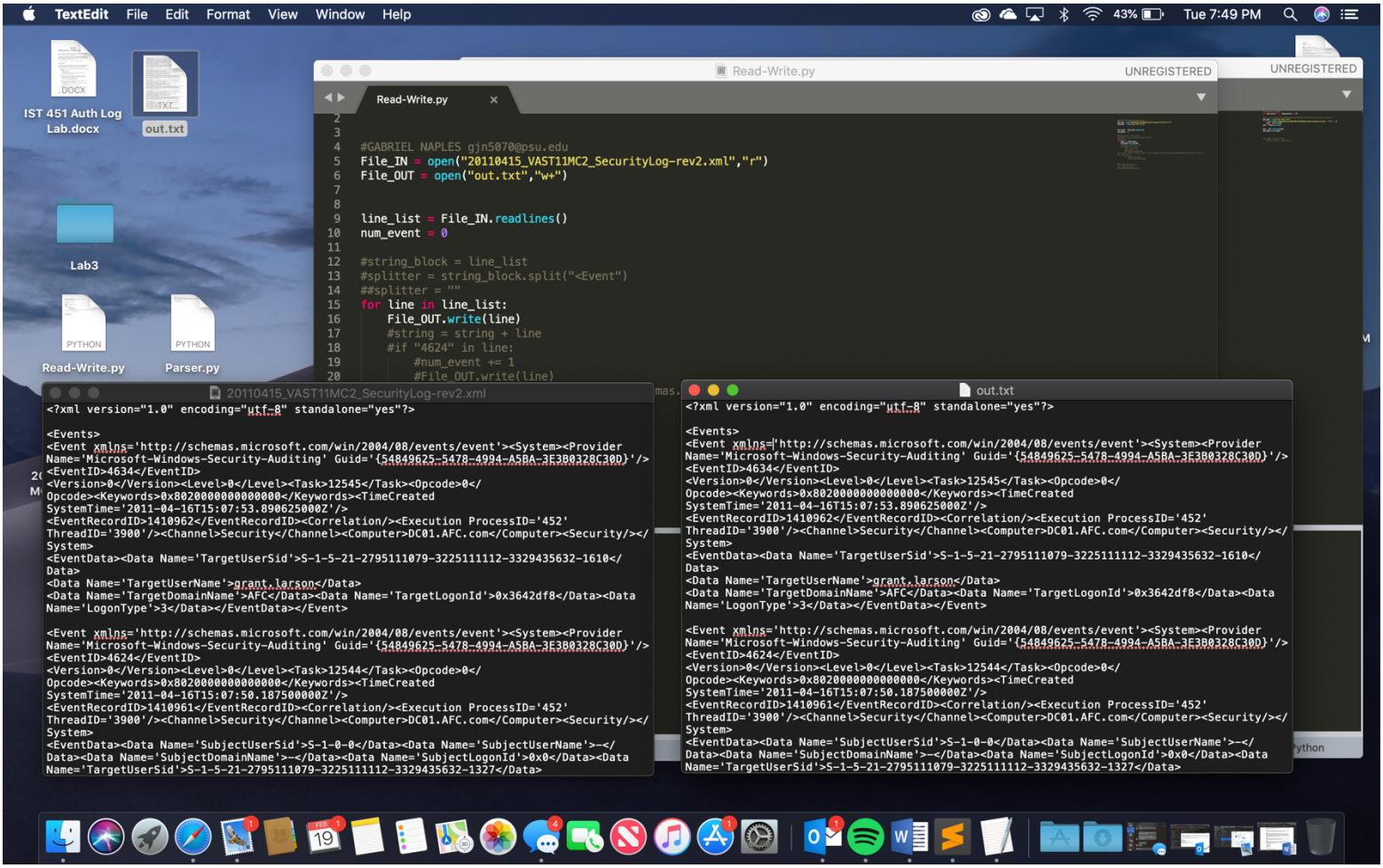
Gabriel Naples
2/20/19
Lab 3

Part A

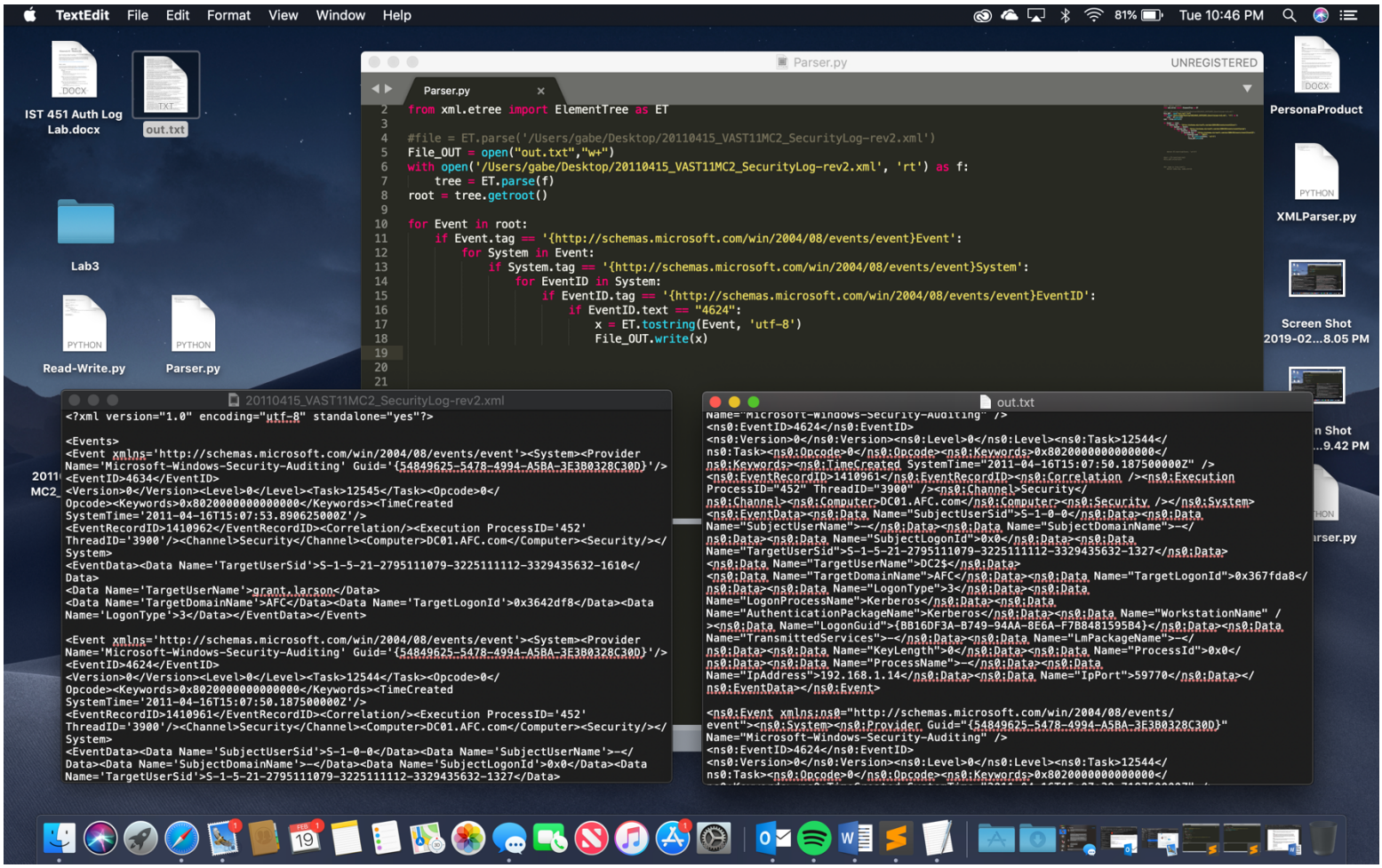


This is the code that uses the “ElementTree” library to print out the information taken from the xml document. The original document is opened on the left and the out.txt (the new file the output put information in) is on the right. Within the output file there is the addition of ‘ns0’ before the events which is associated with a namespace from the xml document. I have also done a similar copy function without using the ElementTree library which copies the file line by line and puts it in “out.txt.” The code for that is below.



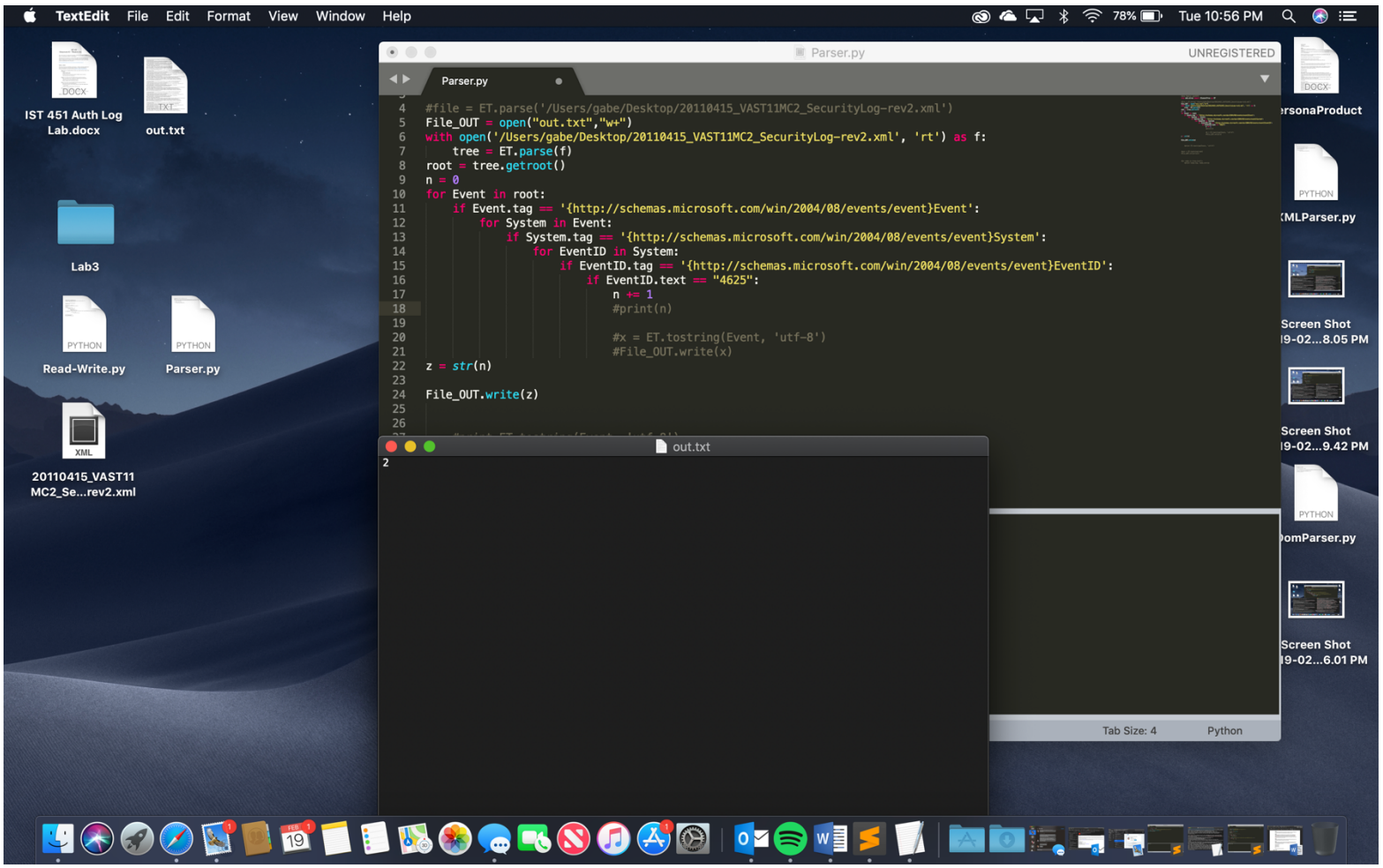
Part B

The below code shows the program parsing through the tree of the XML file. When first it gets within the over “Events” and then into each “Event.” From within each “Event” it scans for the “EventID.” Then it compares the “EventID.text (which shows the text between the EventID tags)”to see if it equals “4624.” If it does, then it outputs the entire event into the file. The output is showed to the right.



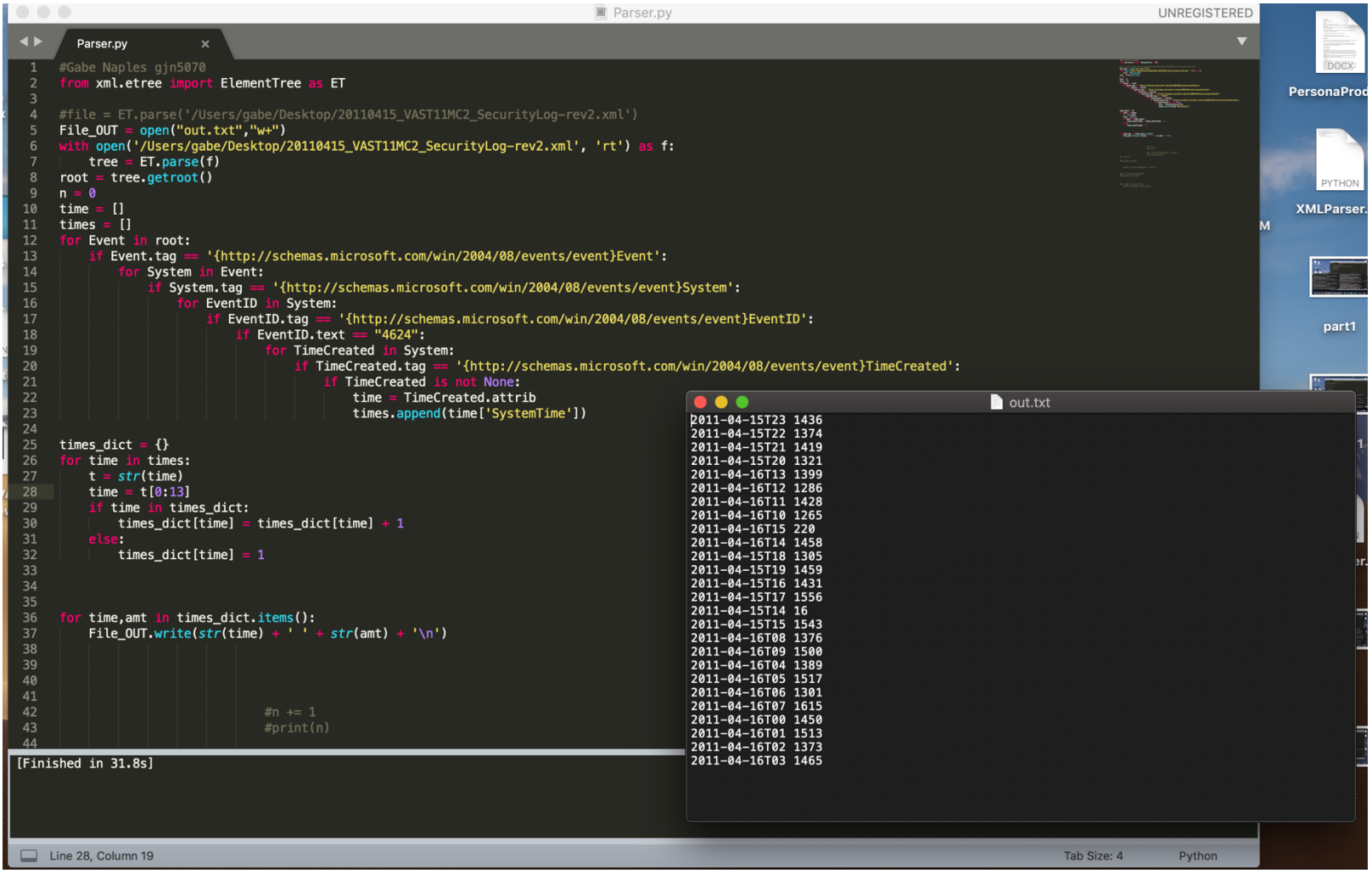
Part C

In the below code it scans to see if the text within “EventID” in the tag equals “4625.” If it does then it iterates the counter to add one and then run again, which creates a running count of all of the occurrences of that EventID.



Part D

In the below screenshot it shows the code that takes the system times for event 4624 and displays them in the out.txt file in a way that shows the names of the occurrences and how many times each one occurs under the EventID of “4624.” This program takes the full SystemTime which includes a longer string than just the date. The program takes the whole SystemTime and then converts it to a string and shortens it to include only the first 12-character spaces which is the date. It then takes that value and adds it to a dictionary. When adding these values to the dictionary it iterates through and compares values. If the value is not present, then it adds it and adds the value of the occurrence to ‘1.’ If it is already in the dictionary, then it simply increases the value of occurrence by ‘1.’ Once that is done it outputs the date followed by the amount of times it occurs in the “out.txt” file for each dictionary entry.


Parser.py
UNREGISTERED
Parser.py
#Gabe Naples gjn5070
from xml.etree import ElementTree as ET
#file = ET.parse('/Users/gabe/Desktop/20110415_VAST11MC2_SecurityLog-rev2.xml')
File_OUT = open("out.txt","w+")
with open('/Users/gabe/Desktop/20110415_VAST11MC2_SecurityLog-rev2.xml', 'rt') as f:
tree = ET.parse(f)
root = tree.getroot()
n = 0
time = []
times = []
for Event in root:
if Event.tag == '{http://schemas.microsoft.com/win/2004/08/events/event}Event':
for System in Event:
if System.tag == '{http://schemas.microsoft.com/win/2004/08/events/event}System':
for EventID in System:
if EventID.tag == '{http://schemas.microsoft.com/win/2004/08/events/event}EventID':
if EventID.text == "4624":
for TimeCreated in System:
if TimeCreated.tag == '{http://schemas.microsoft.com/win/2004/08/events/event}TimeCreated':
if TimeCreated is not None:
time = TimeCreated.attrib
times.append(time['SystemTime'])
times_dict = {}
for time in times:
t = str(time)
time = t[0:13]
if time in times_dict:
times_dict[time] = times_dict[time] + 1
else:
times_dict[time] = 1
for time,amt in times_dict.items():
File_OUT.write(str(time) + ' ' + str(amt) + '\n')
#n += 1
#print(n)
[Finished in 31.8s]
Line 28, Column 19
Tab Size: 4
Python
out.txt
2011-04-15T23 1436
2011-04-15T22 1374
2011-04-15T21 1419
2011-04-15T20 1321
2011-04-16T13 1399
2011-04-16T12 1286
2011-04-16T11 1428
2011-04-16T10 1265
2011-04-16T15 220
2011-04-16T14 1458
2011-04-15T18 1305
2011-04-15T19 1459
2011-04-15T16 1431
2011-04-15T17 1556
2011-04-15T14 16
2011-04-15T15 1543
2011-04-16T08 1376
2011-04-16T09 1500
2011-04-16T04 1389
2011-04-16T05 1517
2011-04-16T06 1301
2011-04-16T07 1615
2011-04-16T00 1450
2011-04-16T01 1513
2011-04-16T02 1373
2011-04-16T03 1465
DOCX
PersonaProd
PYTHON
XMLParser.
part1