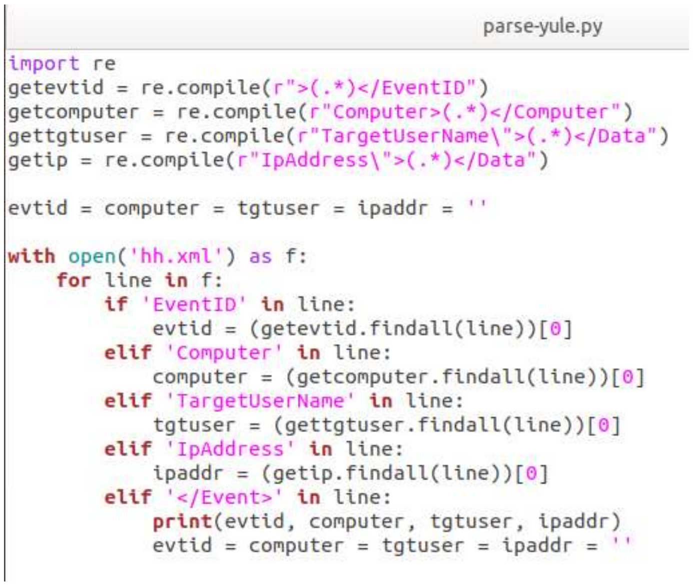
Terminal--Yule Log (Part 3)

Solution for the Python script

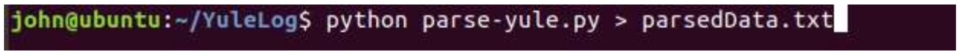
This is the Python script that I used. Yours may be different.


parse-yule.py

```
import re
getevtid = re.compile(r">(.*)</EventID")
getcomputer = re.compile(r"Computer>(.*)</Computer")
gettgtuser = re.compile(r"TargetUserName\">(.*)</Data")
getip = re.compile(r"IpAddress\">(.*)</Data")

evtid = computer = tgtuser = ipaddr = ''

with open('hh.xml') as f:
    for line in f:
        if 'EventID' in line:
            evtid = (getevtid.findall(line))[0]
        elif 'Computer' in line:
            computer = (getcomputer.findall(line))[0]
        elif 'TargetUserName' in line:
            tgtuser = (gettgtuser.findall(line))[0]
        elif 'IpAddress' in line:
            ipaddr = (getip.findall(line))[0]
        elif '</Event>' in line:
            print(evtid, computer, tgtuser, ipaddr)
            evtid = computer = tgtuser = ipaddr = ''
```

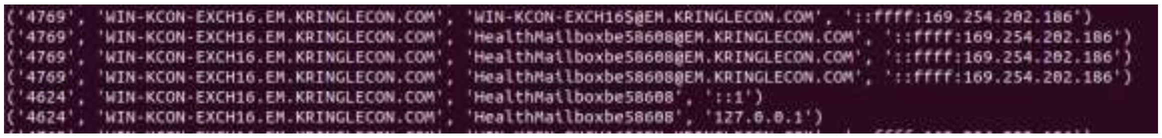
```
import re
getevtid = re.compile(r">(.*)</EventID")
getcomputer = re.compile(r"Computer>(.*)</Computer")
gettgtuser = re.compile(r"TargetUserName\">(.*)</Data")
getip = re.compile(r"IpAddress\">(.*)</Data")

evtid = computer = tgtuser = ipaddr = ''

with open('hh.xml') as f:
    for line in f:
        if 'EventID' in line:
            evtid = (getevtid.findall(line))[0]
        elif 'Computer' in line:
            computer = (getcomputer.findall(line))[0]
        elif 'TargetUserName' in line:
            tgtuser = (gettgtuser.findall(line))[0]
        elif 'IpAddress' in line:
            ipaddr = (getip.findall(line))[0]
        elif '</Event>' in line:
            print(evtid, computer, tgtuser, ipaddr)
            evtid = computer = tgtuser = ipaddr = ''
```

The command below runs the script. Again, note that the XML data from the server is stored in hh.xml. I didn't bother to use an argument to read the file from the command line.



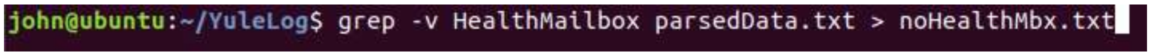
```
john@ubuntu:~/YuleLog$ python parse-yule.py > parsedData.txt
```

As I looked through the log with `less`, I found that there were large numbers of entries for "HealthMailbox". Let's get rid of those.



```
('4769', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'WIN-KCON-EXCH16$@EM.KRINGLECON.COM', '::ffff:169.254.202.186')
('4769', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'HealthMailboxbe58608@EM.KRINGLECON.COM', '::ffff:169.254.202.186')
('4769', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'HealthMailboxbe58608@EM.KRINGLECON.COM', '::ffff:169.254.202.186')
('4769', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'HealthMailboxbe58608@EM.KRINGLECON.COM', '::ffff:169.254.202.186')
('4624', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'HealthMailboxbe58608', '::1')
('4624', 'WIN-KCON-EXCH16.EM.KRINGLECON.COM', 'HealthMailboxbe58608', '127.0.0.1')
```

The -v option in grep tells it to omit lines that match, instead of selecting them.



```
john@ubuntu:~/YuleLog$ grep -v HealthMailbox parsedData.txt > noHealthMbx.txt
```

Password Spraying vs. Brute Forcing

In a brute force password attack, different passwords from a list are tried against one account until the account is locked or the attack is successful. A successful attack will appear in the logs as a series of unsuccessful attempts against an account followed by a successful login.

In password spraying, one password is tried against a series of accounts and then another round starts with a new password. Each account with have either one failure or one success for each round. So, looking for a failed attempt on Alabaster's followed by a successful login to Alabaster's account will not help.

Hand In

Scan through the file you generated (mine was noHealthMbx.txt) and see if you can spot a suspicious login.

1) Who fell victim to the password spraying attack?