

Terminal--Yule Log (Part 2)

Solution for the EventIDs

The command to get the terminal to convert the .evtx file to XML was:

```
python evtx_dump.py ho-ho-no.evtx
```

```
elf@80e2a1c50a9b:~$ ls -l
total 6896
-rw-r--r-- 1 elf elf 1353 Dec 14 16:13 evtx_dump.py
-rw-r--r-- 1 elf elf 1118208 Dec 14 16:13 ho-ho-no.evtx
-rwxr-xr-x 1 elf elf 5936968 Dec 14 16:13 runtoanswer
elf@80e2a1c50a9b:~$ python evtx_dump.py ho-ho-no.evtx
```

My solution to find the EventIDs was this:

```
elf@e5ab6bbe8baf:~$ python evtx_dump.py ho-ho-no.evtx | grep
EventID | sort | uniq -c | sort -n
```

```
1 <EventID Qualifiers="">4608</EventID> win start
1 <EventID Qualifiers="">4647</EventID> log off
1 <EventID Qualifiers="">4826</EventID> Boot config db change
1 <EventID Qualifiers="">4902</EventID> Change to audit policy
1 <EventID Qualifiers="">5024</EventID> Firewall started
1 <EventID Qualifiers="">5033</EventID> Firewall start
2 <EventID Qualifiers="">4724</EventID> reset password
2 <EventID Qualifiers="">4738</EventID> user account changed
2 <EventID Qualifiers="">4904</EventID> Register sec event source
2 <EventID Qualifiers="">5059</EventID> Key Storage Provider import or export
10 <EventID Qualifiers="">4688</EventID> new process
34 <EventID Qualifiers="">4799</EventID> Enumerate group
45 <EventID Qualifiers="">4768</EventID> TGT req
108 <EventID Qualifiers="">4776</EventID> AD success logon*****
109 <EventID Qualifiers="">4769</EventID> TGT req
212 <EventID Qualifiers="">4625</EventID> fail log on*****
756 <EventID Qualifiers="">4624</EventID> success log on*****
```

I manually added the name of the event to each line. The events of interest are 4776, 4624, and 4625. There were many failed logins, much more than usual.

One problem with this data is that a single event takes multiple lines. The `grep` command works best when all the data you seek is in a single line. You can compensate for this by using the `-A 25` option, where `grep` will show you 25 lines after the match, but I found it easier to write a simple Python script that grabbed the information I needed. Finding a line containing EventID is easy. If the variable holding the contents of one line of XML is `line`, then this will work.

```
if 'EventID' in line:
```

How do we grab the EventID number out of the line? Here's a line with an EventID. We want 4625.

```
<EventID Qualifiers="">4625</EventID>
```

One way to grab the number is to use a Regular Expression.

This will match the last bit of text to the left of 4625:

>

This will match the right, and make sure we catch EventID

</EventID

This will catch the number in the middle

(.*)

An added benefit of the parentheses in that match that catches the number is that it makes the number a group and makes it easy to recover. **Note:** If you want a regular expression for something that includes a quote (like TargetUserName) you must “escape” it (somestuff\">(and so on.)

The regular expression grab the number from an EventID is then

>(.*)</EventID

If a regular expression is being used repeatedly, you can save time by compiling the expression before it will be used and saving it in a variable. The lines to do this in Python for our regular expression would be:

```
import re
getevtid = re.compile(r">(.*)</EventID")
```

```
import re
getevtid = re.compile(r">(.*)</EventID")
```

Note that the re.compile method requires that the value starts with the letter “r” and encloses the expression in quotes.

A simple Python script that grabs fields of interest from the XML file follows. It just checks each line for the fields we want and stores the values. The </EventID> tag signifies the end of an event. If we see that, we print our values and reset them for the next event. Note that I named the file that contains the XML data pasted from the terminal, “hh.xml”.

```
import re
# compile the regexs
getevtid = re.compile(r">(.*)</EventID")
# regexs for other fields go here

# clear our variables
evtid = = = = ''

#open the file
with open('hh.xml') as f:
#read the file line by line
    for line in f:
        if 'EventID' in line:
            evtid = (getevtid.findall(line))[0]
        elif #other fields go here
        elif #the look almost like the EventID lines
        elif
#we hit the end of an event, so print and clear variables
        elif '</Event>' in line:
            print(evtid, , , )
            evtid = = = = ''
```

Hand In

Pick some other elements to extract and add them to the script. TargetUserName should be one of them.

- 1) Turn in your script and the file that it created.