

Terminal--Yule Log (Part 1)

What you can learn from this

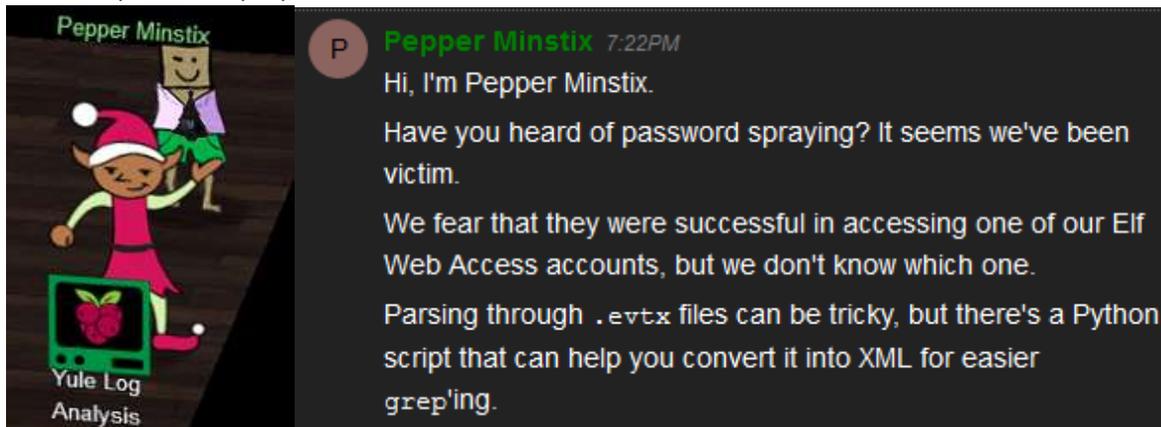
It is important that security professionals (and any IT administrators) be able to parse large log files. Linux has many tools that are helpful, among them `grep` and `awk`. In addition, my method of solving this challenge involved Python and [regular expressions](#). Others solved it more simply, but this helped me to understand the problem better. Using `grep`, writing a short Python script, and using regular expressions will be good practice.

Required Watching

In this challenge you are required to find an account that was compromised with a password spraying attack. To do that, you need to understand what password spraying is, and how it is different from password brute force attacks. Watch this presentation by Beau Bullock, [Everything You've Ever Wanted to Know About Password Spraying](#). Be sure that you understand the difference between a brute force password attack and a password spraying attack.

Getting Started

Pepper Minstix and her Yule Log terminal are on the right side of the second floor, past Tangle Coalbox and the Speaker Unpreparedness room.



Run the Python script on the Yule Log terminal to translate the `.evtx` file into XML. Allow it to display to the screen so you can copy it and paste it to your local machine for analysis. After the file scrolls through your terminal, you'll be at the bottom of the file. Click there, scroll up to the command you entered and shift-click. This should select the entire text of the file. Copy the file with Control-C (right-click copy will not work) and paste it into a file on your machine.

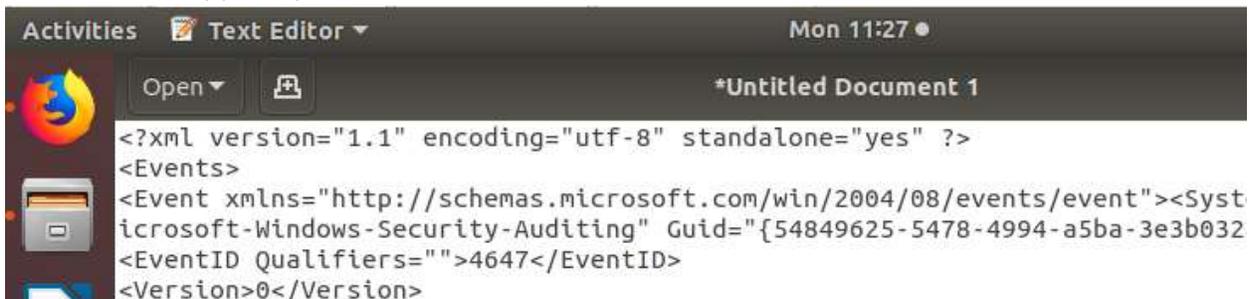
Select some text at the bottom.

```
<Data Name="RestrictedAdminMode"></Data>
<Data Name="TargetOutboundUserName"></Data>
<Data Name="TargetOutboundDomainName"></Data>
<Data Name="VirtualAccount">%1843</Data>
<Data Name="TargetLinkedLogonId">0x0000000000000000</Data>
<Data Name="ElevatedToken">%1842</Data>
</EventData>
</Event>
</Events>
elif@ed2f70330610:~$
```

Scroll to the top and shift-click (I've blacked out part of the command.)

```
elif@ed2f70330610:~$ python
<?xml version="1.1" encoding="utf-8" standalone="yes" ?>
<Events>
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event"><System><Provider Name="M
icrosoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}"></Provider>
<EventID Qualifiers="">4647</EventID>
<Version>0</Version>
<Level>0</Level>
<Task>12545</Task>
```

Control-C and copy into your own document (this is Gedit in Linux.)



One method for solving this challenge would be to import the file you've created into Python (or some other language, even PowerShell) as XML and analyze it from there. I found that to be less than straightforward, so I'll leave it as an exercise. Some notes: It works better in PowerShell if you change the XML version of the file to 1.0 instead of 1.1.

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Events>
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event"><System><Provider Name="M
icrosoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}"></Provider>
<EventID Qualifiers="">4647</EventID>
<Version>0</Version>
<Level>0</Level>
```

Also, be sure the file has an </Events> tag at the end to close out the XML.

```
<Data Name="ElevatedToken">%%1842</Data>
</EventData>
</Event>
</Events>
Plain Text Tab Width: 8 Ln 50895, Col 1 INS
```

Examine the file

First, it is good to understand the format of the file. The XML root for the entire file starts with <Events> and ends with </Events>. That is important if we were to use an XML editor. For our purposes the important part is that each individual event starts with <Event> and ends with </Event>. The primary elements we are interested in are EventID (what happened) and

TargetUserName (who it happened to.) Additional fields of interest could be the time, the users' SID, the computer, IP address, etc.

This is a sample of one event, in XML.

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event"><
Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a
Provider>
<EventID Qualifiers="">4624</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>12544</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2018-09-10 12:19:20.695601"></TimeCreated>
<EventRecordID>231726</EventRecordID>
<Correlation ActivityID="" RelatedActivityID=""></Correlation>
<Execution ProcessID="664" ThreadID="668"></Execution>
<Channel>Security</Channel>
<Computer>WIN-KCON-EXCH16.EM.KRINGLECON.COM</Computer>
<Security UserID=""></Security>
</System>
<EventData><Data Name="SubjectUserSid">S-1-0-0</Data>
<Data Name="SubjectUserName">-</Data>
<Data Name="SubjectDomainName">-</Data>
<Data Name="SubjectLogonId">0x0000000000000000</Data>
<Data Name="TargetUserSid">S-1-5-18</Data>
<Data Name="TargetUserName">SYSTEM</Data>
<Data Name="TargetDomainName">NT AUTHORITY</Data>
<Data Name="TargetLogonId">0x000000000000003e7</Data>
<Data Name="LogonType">0</Data>
<Data Name="LogonProcessName">-</Data>
<Data Name="AuthenticationPackageName">-</Data>
<Data Name="WorkstationName">-</Data>
<Data Name="LogonGuid">{00000000-0000-0000-0000-000000000000}</Data>
<Data Name="TransmittedServices">-</Data>
```

What EventIDs are in the file, and which ones should we look for?

It will be good to know what we are looking for. What EventIDs represent failed logins, successful logins (there may be more than one,) and are they present in the file. Write a simple one-line grep command that will grab all the lines that contain the string EventID, sort them, find unique EventIDs, and count them. Then look up the EventIDs that are present and see what they mean.

Hand In

- 1) What was your command to convert the .evtx file to XML?
- 2) What is your command grep for EventIDs, sort them, and count unique events?
- 3) What EventIDs are present and what do they mean?