Intro slide:

Hello everyone, today I will be presenting on a python script used to monitor Windows event logs.

Slide 2:

Windows Event logs are special files that record significant events on your computer, such as when a user logs on or when an application encounters an error. Each event will generate their own event logs and the information contained within the log depends on the category of event.

Within the logs more information is contained related to the type of Windows event, security level, and time generated just to name a few.

Slide 3:

Since this is such a large repository of information, having a good log monitoring policy is a simple but effective way to protect oneself throughout the cybersecurity cycle. It can provide information to set baselines of “normal” activity, help to identify and prevent attacks, as well as perform post breach forensics. Evidence of security breaches have been found in event logs, but they often go unnoticed because of the large volume of data collected. A large corporation may log thousands of events per second and without a log monitoring system in place this can be an overwhelming task to tackle. A 2012 Verizon Data Breach report found that even though 85% of breaches took several weeks to be discovered, 84% of victims had evidence of the breach in their event logs.

Slide 4:

Hackers try to hide their presence, so knowing what to look for is important.

A problem we may all run into is an application error which seems harmless, but frequent application crashes could actually be the work of an attacker trying to cover their tracks from inserting malware or a virus onto your computer. Performing a search of the application logs looking at Event ID 1000 (application error) can provide more information and

A popular attack is Pass the Hash which allows hackers to gain access to an account without needing to know the password. This was used as recently as April 2022 by a Ransomware as a Service platform called Hive. This attack leveraged the hash technique to target a large number of Microsoft’s Exchange Server customers. Monitoring the Security logs for event IDs associated with logins, such as 4624 (account successfully logged on), 4625 (account failed to log on), or 4648 (login attempted with special credentials) could reveal more information and help detect suspicious levels of activity

These are just two examples of how an attacker tries to cover their tracks and ways you can use the logs for monitoring. Taking this knowledge I developed a python script to search by log type and event ID to print out some quick information that can be used to further investigate if a problem is suspected or detected.

Slide 5:

Python doesn’t come preinstalled on most windows computers, so I had to install it on my machine.

In order for python to access the Windows event log you also need to install pywin32 module and it is done using the following command:

Slide 6:

Once you have done that you can open python, making sure to run as administrator and you will have the following python shell to work from.

Using python tutorials and repositories I was able to develop a script to monitor the windows event logs

Slide 7:

Demo

Starting from the Python shell that I ran as administrator

I will open my python script

Which will show in a new window:

Here is my script that I developed with the help of tutorial videos on python scripting as well as repositories on python scripts for Windows event logs

My script will ask for an event id input and then search the win32 event logs for that event id. It will print out the time generated, source name, computer name and event type.

First we will run the python script from the python shell by clicking run and then run module.

In this window it asks for the log type: Application and then event ID and we will type in 1000

It will then print out each log entry with the information we requested on one line. For my computer there aren’t too many logs and I can use the time generated field to see that there are several days between events, so this may not be the result of suspicious activity.

Next from the command line we can run this script by typing python and the script name. We will search the Security Logs for Event ID 4648 (login attempted with special credentials). This prints out significantly more entries and using the information provided I may want to do further investigating which could be done by looking into those other event IDs associated with logins.

Slide 8:

To wrap up my demonstration I will touch on the application of this script.

The benefit of this script comes when you apply it outside of using it on an individual computer because Windows does have an event viewer which you can use to interact with these logs. However the script allows for scalability when applying to a large organization and you may need to search for events across hundreds or even thousands of machines. Don’t have to go do individually can run this once and get the information needed

Lastly, If you have a Siem that is not configured to consume Windows logs, this script would be a way to access that. This is an issue the MN Air National Guard defensive cyber operations team was running into and was the inspiration for my project. This script was actually recently used by them to conduct host based forensic analysis on a particular Event ID they were monitoring.

This concludes my presentation. I have a slide here on the references I used. Thank you for your time.