11/8/2021

Lab 3 – IS.6303.001

Log File Analysis

Joseph O. Mendez
QXO307

Joseph Mendez
Qxo307
IS-6303-001

After our first two labs dealing with topics such as testing out and analyzing operating systems as well as cracking password files from Windows and Linux machines; we shift our gaze to another important field in cyber security: that of computer log analysis. In our third lab of the semester, we will look at various aspects of logs and logging. We will start by defining what logs are, how they function on different OSs, and how to access them. Between these, we'll also cover how to read them, what tools can be used to analyze them. In these sections, we will give slightly more focus to the Window's side of things as that's what our final section will be about. Speaking of the final section, our last report topic will discuss an example analysis of Windows OS logs.

Now to kick things off, we'll start by asking "what is a computer log"? Logs are a certain type of file created by computers to track various forms of information, including "… usage patterns, activities, and operations…" (What Is a Log File?). With these logs, users can then formulate reports on a plethora of analytics. In our case, however, computer logs are mainly used to monitor computers in real time and focus on a computer's health and software usage. The Window's logs are rather detailed and, as such, allow users to diagnose problems and issues while also being able to predict any future issues that happen. The elements that come with an individual entry include:

- ***Date***
- ***Time***
- ***User***
- ***Computer***
- ***Event ID***
- ***Source***
- ***Type***

Most of these elements are straight forward and do not require any explanation. The *Event ID* is the ID number that tells the user what the event type is for that entry. The *Source* attribute informs what application, file, or component cause the event. And lastly, the *Type* attribute tells the user what type of event this is (such as information, error, security success audit, security failure audit, or warning).

These Windows logs are in the *C:\WINDOWS\system32\config* directory. However, Windows does come with GUI in which to view these. The *Windows Event Viewer*, which will be discussed later, divides events into five main categories. The first of these is the *Application Events* which refers to events that arise from any application that is in the computer. If an application crashes, there is where the event will be logged. Next, the *Security Events* section contains entries that deals with audit polices. A good example includes entries that log verification for account credentials. *Setup Events* focus on enterprise events that relate "… to the control of domains, such as the location of logs after a disk configuration". On the other hand, System Events deal with logs for the Window's system. Here is where issues and the status for

device drivers will be displayed. Finally, the *Forwarded Events* area contains logs for other machines on the same network. This section is mainly used for when admins want to gather logs from other systems.

As mentioned before, the *Windows Event Viewer* application allows users to view the computer logs that are on the computer. The Event Viewer is accessed by clicking on the Windows button and searching for the Windows Event Viewer. From here, users can then choose which of the specific log types they'd like to see by selecting it on the left-hand plane. Windows then categorizes events with security levels. These are then ordered from information, to warning, error, and critical. The log type with the most entries in general would have to be the *Information Events*. This is due to them not being tied to any event that causes issues or problems. *Warning Events* are those that attempt to inform users of any potential risk. These risks, however, aren't as drastic or important in which they'd need immediate attention. *Error Events* are those in which a component of the system failed or is not operating as expected. The last of these is the *Critical Events*. These represent the most serious problems present on a system (Gillis).

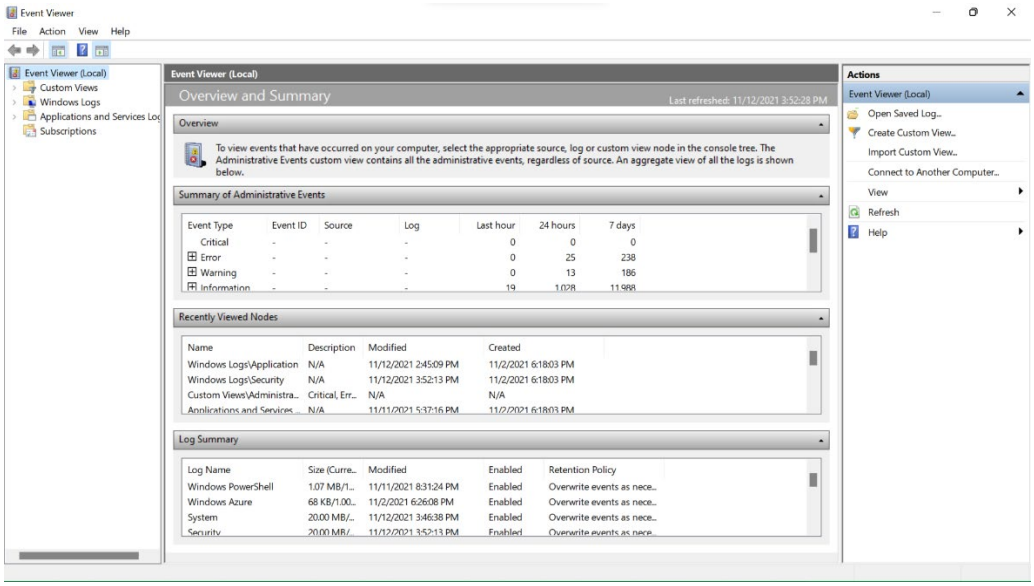

Figure 1 – Windows Event Viewer

To now shift our gears a little over to the Linux side, Linux systems store their logs in the */var/log* directory. Here, users can find logs varying from logs containing information on the OS to those that contain information on its myriad of applications. However, from its large list of logs, a few stand out as being of more importance. The */var/log/syslog* and */var/log/messages* store a large overview of system activity information. For this information, Debian-based systems use */var/log/syslog* while Red Hat-based systems use */var/log/messages*. */var/log/auth.log* and */var/log/secure* contain security events. These mainly include login information, root-user actions, and pluggable authentication module (PAM) outputs. As with the two previously stated

logs, these two differ based on what Linux distribution is being used. *…/auth.log* is used by Ubuntu and Debian while *…/secure* is used by CentOS. */var/log/kern.log* stores events that deal with the kernel, any errors, and warning logs; and is universal among all distributions of Linux. Lastly, */var/log/cron* contains information about cron jobs (scheduled tasks). It is also universally used in all Linux distributions.

As for their structure, a basic *Syslog* is comprised of a header containing a timestamp, the application that produced the log entry, location on system of where the entry came from, and its priority. A nifty feature of these logs is that a user can add new fields to their logs. An example from SolarWinds' Loggly site is converting the basic log structure of:

- `Jun 4 22:14:15 server1 sshd[41458] : Failed password for root from 10.0.2.2 port 22 ssh2`

into a the following log by using the *<%pri%>, %protocol-version%, and %timestamp::date-rfc3339%* entries:

- `<%pri%>%protocol-version% %timestamp:::date-rfc3339% %HOSTNAME% %app-name% %procid% %msgid% %msg%n`
- `<34>1 2019-06-05T22:14:15.003Z server1 sshd - - pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.2.2`

As shown with the entries above, there are certain fields that are of a little more importance than most. These are as follows:

- ***Timestamp*** – shows the date and time for when the entry was generated.
- ***Hostname*** – shows the name of the system or host that created the message.
- ***App-Name*** – shows the name of the application that created the message.
- ***Priority*** – shows how urgent the event is. It is composed of two main numerical fields, the facility and severity values. The facility value represents the type of process that generated the event. The severity value ranges from 0 to 7 and represents the severity of the event (Linux Logging Basics - The Ultimate Guide to Logging).

When it comes to Linux, its log files are all registered as text files. As such, any program that can read/display text files can be used to read these Linux computer logs. These logs can be viewed using commands such as *dmesg, tail, grep, head, cat, more, or multitail* (Wallen). For the most part, Linux users are known for their preference of the *terminal*, and as such the methods to view the logs may work for them. However, others may want something akin to Window's Event Viewer in which they'll be able to view them or analyze them easier. For these, users can choose to use log analyzing freeware or paid applications.

Some of these include tools such as *Sematext Logs*; a log management system working from the cloud that allows users to analyze their logs in real time. *Sematext Logs* contains various features for analysis such as text searches, filtering, and log tagging. Users can also set up alerts for certain situations and view errors as they're logged. This software is available for users at both a free and paid plan. Another useful tool is *SolarWind's Loggly*, another cloud-based log

management tool. This tool allows for collection of server or client-side logs via *Syslogs* or *HTTP* and contains potent search tools and indicators. The last tool we'll mention is a rather popular one in the cybersecurity field. *Splunk* is a mostly free log management system that uses a multi-line method to collect logs of any type, including both "structured, unstructured, and sophisticated application logs'' (Kuć). Before moving on to the final section of the report, it is important to note that these applications, while mentioned in our Linux section of the report, are not OS-exclusive and can also be used on Windows if one would prefer them over what Windows natively provides.

Lastly, we move to the last section of our report: the log analysis. For this section we will cover a certain situation in which a Windows user, in this case me, encountered an issue during their daily computer use. In doing so we'll attempt to establish their reports as follows:

- issue noticed by user
- identifying its event type
- cause of event log
- event log analysis

Let's focus on the following scenario. At around 4:00 PM on Tuesday 9th, 2021 a user was using their windows machine casually with nothing more than their browser for music and *Word* open. Suddenly, the music cuts, and attempting to minimize Word reveals that the process is rather slow. The user also reported some errors with their file explorer. The browser takes so long to respond, that it almost seems like it had frozen or crashed. When it does open, we see that the music video is still playing, despite the lack of sound. Checking the volume, it would seem that the driver may have crashed. After a couple of minutes things go back to normal. With that situation in mind, let's check out the logs to see if we can make sense of what happened.

We open *Event Viewer* and head over to the *System* tab on the left-handed pane. Afterwards, we can use the *Filter Current Log* option to filter the logs to show things from around 4 PM on Tuesday. In this case we'll filter to show logs from Nov. 9th on 3:54 PM to 5:54 PM on the same day. In doing so, we immediately see that there's an error listed.

Joseph Mendez
Qxo307
IS-6303-001

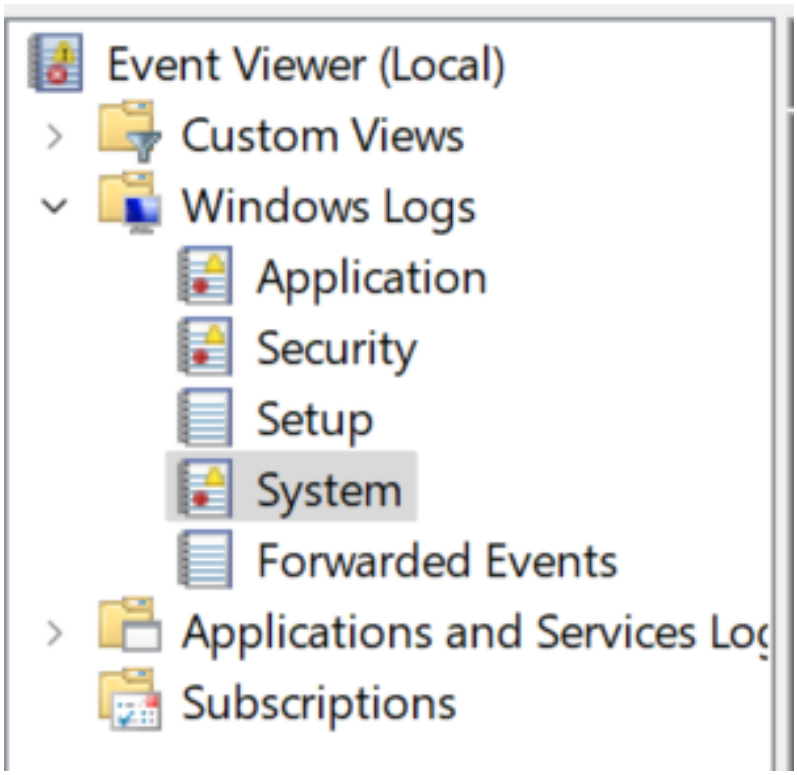

Figure 2 – Left-hand pane showing the various types of logs in Windows Event Viewer

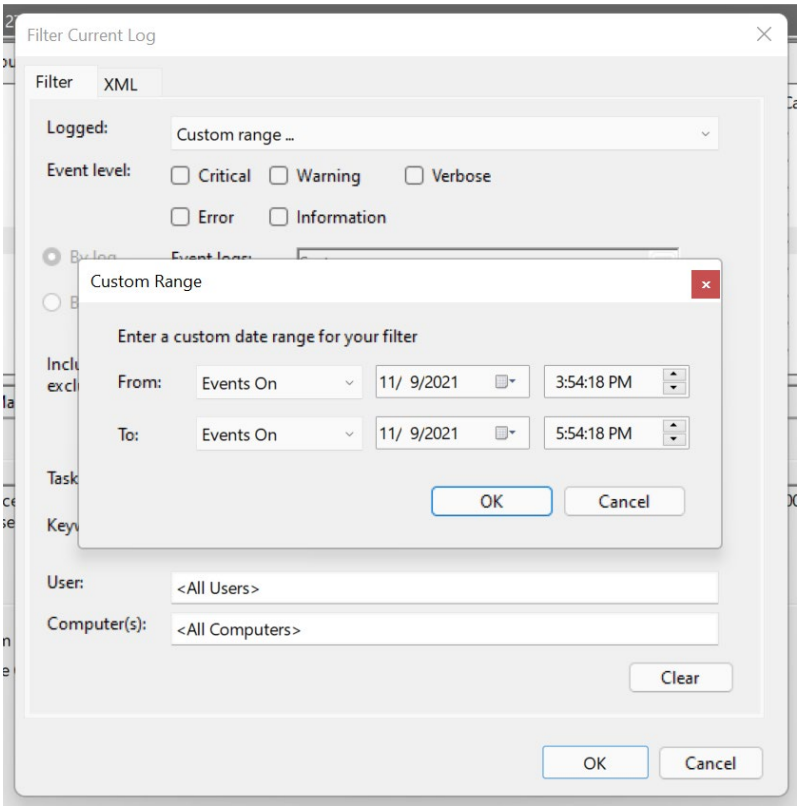

Figure 3 – "Filter Current Log" menu adding a range for a log's timestamp

At 11/9/2021 4:13:33 PM, an issue happened with the computer that was bad enough to have caused an error-level log. This error is listed as having come from the *Service Control Manager* and has an event ID of 7031. Thanks to this, we know that the event is tied to a service crash (Critical Windows Events). We can then take look at the *General* area to see that the specific service that crashed was the *Windows Audio Service*. This crash of the *Audio Service* explains what why the audio went out. However, this does not really explain what happened to the applications open and why they were slow and lagging. For this, we will have to venture out and look at different logs to piece together what caused the rest to happen.

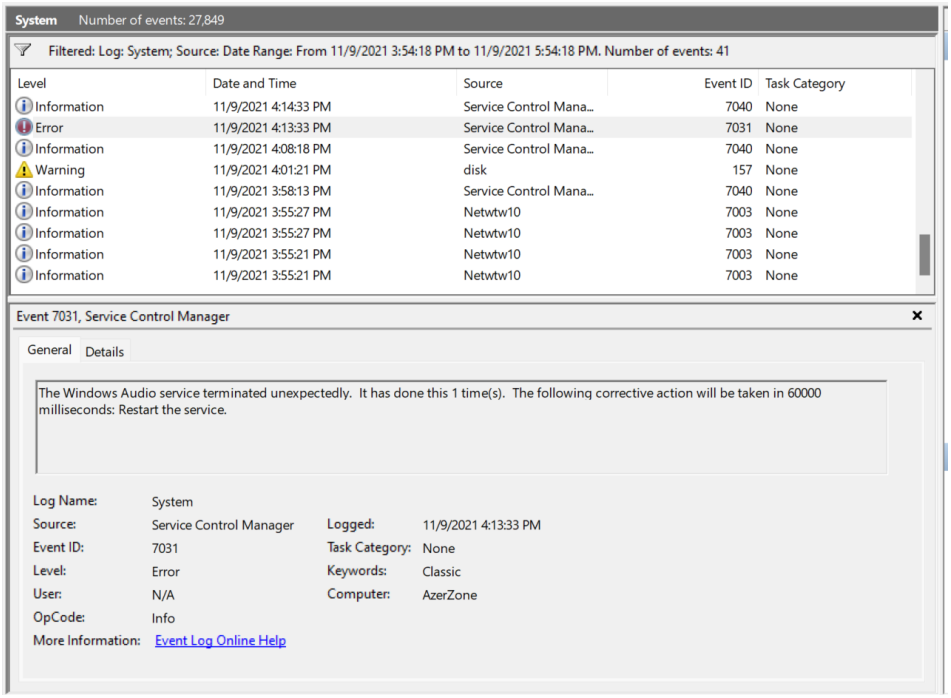

Figure 4 – Example of a 7031 Event

As the error with the browser and *Word* pertain to applications on the computer, we will once again go to the left-hand pane and select the *Application* option. We once again set our filter to the same as the one used for the *System* logs and scroll to around 4:13 PM to check out what we find. Once set, we see that there were a whopping 5 errors all in the span of around 30 seconds. The first of these errors, at 4:13:03 PM, has an event ID of 1002 and signifies an application hang. The specific application in questions is *explorer.exe*, which is the file explorer. The log says that *explorer.exe* was unable to connect to Windows and was subsequently closed. This event gives us an answer as to what happened to the file explorer. The next error occurs at 4:13:14 PM and is yet another application hang. However, this time it's for a crash of *svchost.exe*. There are three more errors after this. Two of them are just crashes for *explorer.exe* and *svchost.exe*, while the other is an application error (event ID of 1000) for a *bad_module_info*. The issues with both *svchost.exe* and *bad_module_info* is interesting as these two could be what caused the lagging. This is because various other Information-type logs listed *svchost* as being

connected to various audio processes, hence a connection to the audio crash log from earlier. On the other hand, the *bad_module_info* event is evident of an error with the system's memory timing. This, in turn, would have led to the lagging of the overall system as well as the crash of *explorer.exe*. Thankfully, this is an issue that should be relatively easy to fix as all that should be done is reboot the system more often. Tools such as *memtest86* can also be run to test the memory of the system for any abnormalities (tyclonebauer2410).

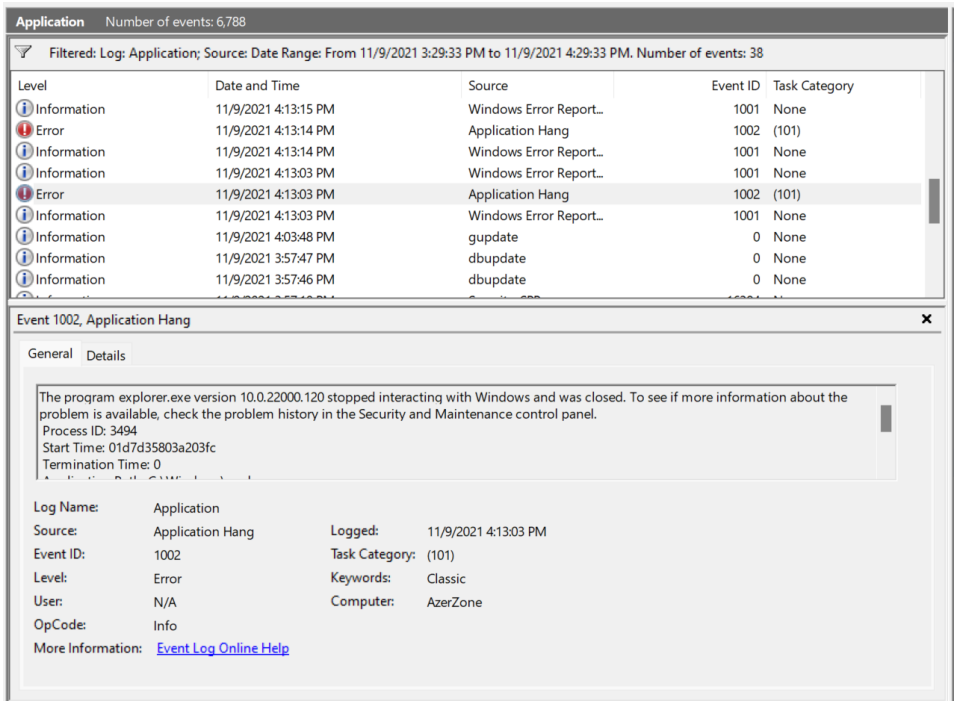

Figure 5 – Example of a 1002 Event for explorer.exe

Joseph Mendez
Qxo307
IS-6303-001

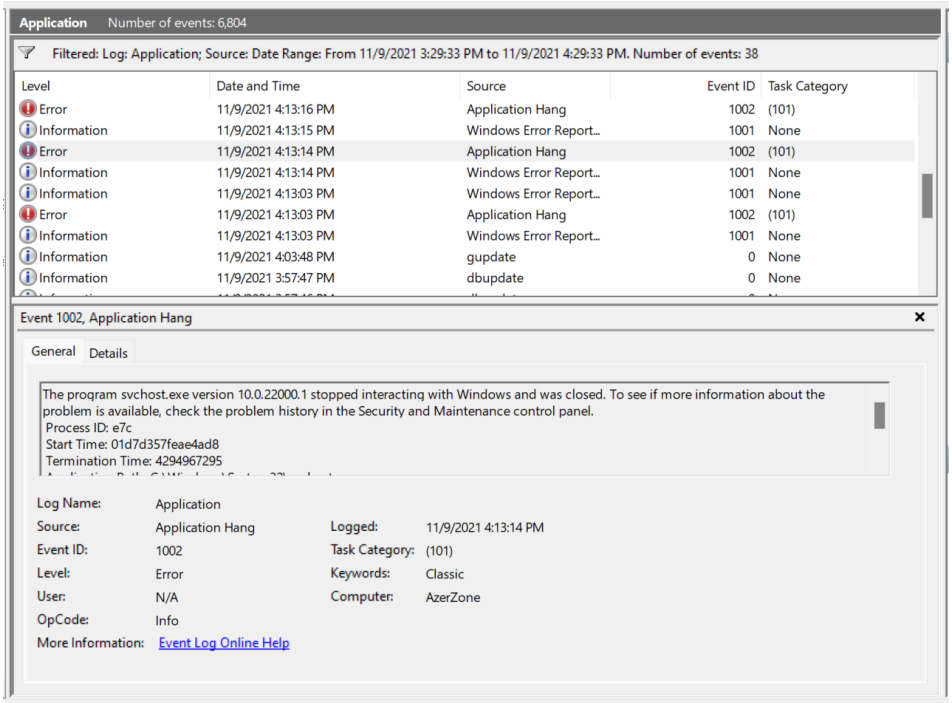

Figure 6 - Example of a 1002 Event for svchost.exe

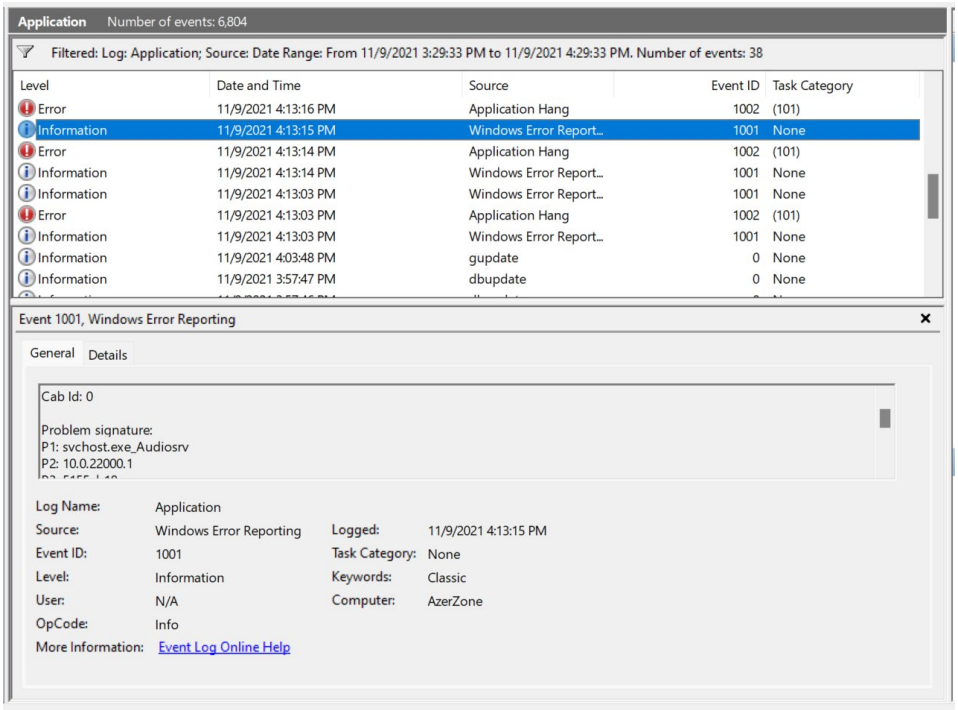

Figure 7 - Example of a 1001 Event where svchost.exe is connected to an audio service

Joseph Mendez
Qxo307
IS-6303-001

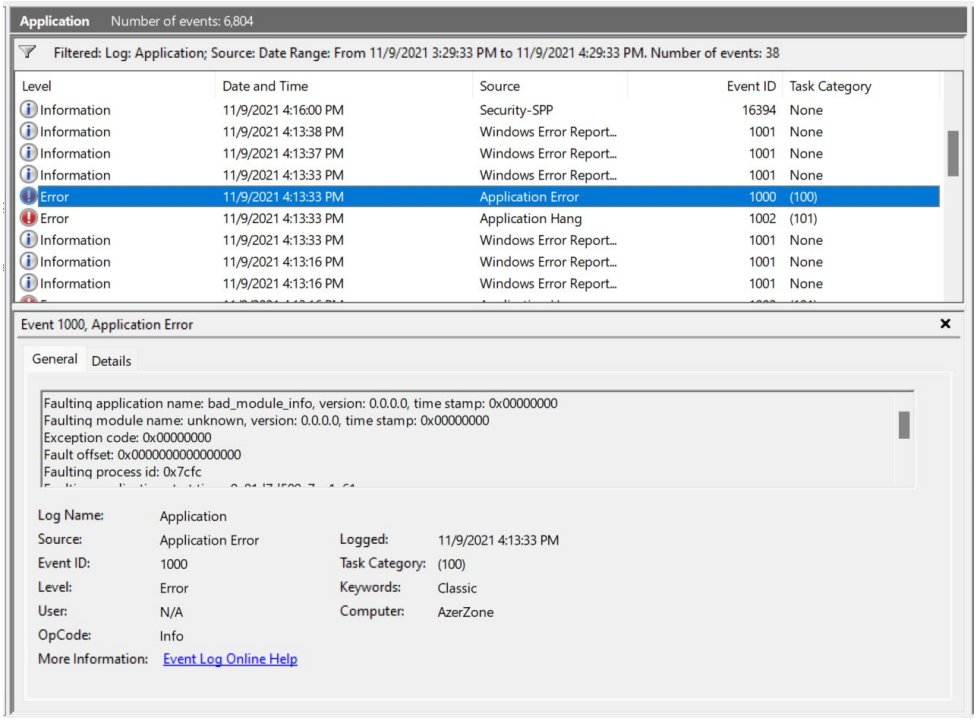

Figure 8 - Example of a 1000 Event for bad_module_info

To lastly wrap-up our report on logs, let's briefly discuss how logs tie into the grander scope of cyber security. How will we do this, one might ask? Well, computer logs can be used as a source of evidence when investigating. These can then, by proxy, be presented in a court of law as evidence. Their applicability comes from how much data logs can hold when it comes to incidents; similarly, to the incident we presented above. Logs can, for example, be used to construct timelines to help investigators narrow down when incidents occurred. These logs can also be used in tandem with an intrusion detection system (IDS) to back up certain events and accusations in court. As with most legal items there are some issues that can arise, however. Computer logs, while not volatile when exported, can still be seen by some as insufficient evidence due to innate vulnerabilities in how they function. Some of the possible vulnerabilities include the ability to disable logging on Windows, important data being modifiable, logs being transferrable, and inaccurate timestamps affecting logs. These can be protected against by using proper forensic and investigative procedures to protect digital artifacts. These procedures include restriction of physical access to the archived evidence, daily backup of logs, use of a single time stamping device, and use of a PKI server to authenticate the users as to protect from injection of fake events (Lucideus).

Joseph Mendez
Qxo307
IS-6303-001

Bibliography

"What Is a Log File?" *Sumo Logic*, https://www.sumologic.com/glossary/log-file/.

Gillis, Alexander S. "What Is Windows Event Log? - Definition from Whatis.com." *SearchWindowsServer*, TechTarget, 31 May 2018, https://searchwindowsserver.techtarget.com/definition/Windows-event-log.

"Linux Logging Basics - The Ultimate Guide to Logging." *Log Analysis | Log Monitoring by Loggly*, 31 July 2019, https://www.loggly.com/ultimate-guide/linux-logging-basics/.

Wallen, Jack. "Viewing Linux Logs from the Command Line." *Linux.com*, 18 July 2018, https://www.linux.com/topic/desktop/viewing-linux-logs-command-line/.

Kuć, Rafal. "10+ Best Log Analysis Tools of 2021 (Free & Paid Log Analyzers)." *Sematext*, Sematext, 30 Aug. 2021, https://sematext.com/blog/log-analysis-tools/.

Lucideus. "Introduction to Event Log Analysis Part 1 — Windows Forensics Manual 2018." *Medium*, Medium, 26 Oct. 2018, https://medium.com/@lucideus/introduction-to-event-log-analysis-part-1-windows-forensics-manual-2018-b936a1a35d8a.

"Critical Windows Events." *Event ID 7031: Service Crash*, https://www.manageengine.com/products/eventlog/kb/event-7031-service-crash-help.html.

tyclonebauer2410. "Bad_Module_Info (Someone Help Please!)." *Tom's Hardware Forum*, 3 Feb. 2018, https://forums.tomshardware.com/threads/bad_module_info-someone-help-please.3232062/.