In this lab investigation, I acted as a Security Analyst I and analyzed a suspected breach by examining both network traffic and Windows event logs. I used Wireshark to go through a packet capture (PCAP) file, looking for any signs of malicious activity on the network. I also reviewed the Windows Event Viewer Application and Security logs to find unusual events on the compromised system. The goal was to piece together what happened during the attack. Like how the attacker got in, what they tried to do, and what errors or warnings showed up as a result. This report explains what I found and how I found it, organized into the investigation process, a timeline of key events, conclusions, and recommendations for preventing such incidents in the future.

Investigation

First, I started with the Network Traffic Analysis tool Wireshark. I started by opening the provided network capture and right away, I filtered the traffic to spot any suspicious login attempts or errors. I noticed some SMB protocol traffic that stood out. In Frame 37 of the capture (see Figure 1), the server's response included a STATUS_LOGON_FAILURE error. The same error appeared again later in Frame 2260 (see Figure 2). A STATUS_LOGON_FAILURE typically means a login attempt failed, probably due to a wrong username or password. These repeated failures suggested that someone was trying to log in remotely and kept getting the credentials wrong.

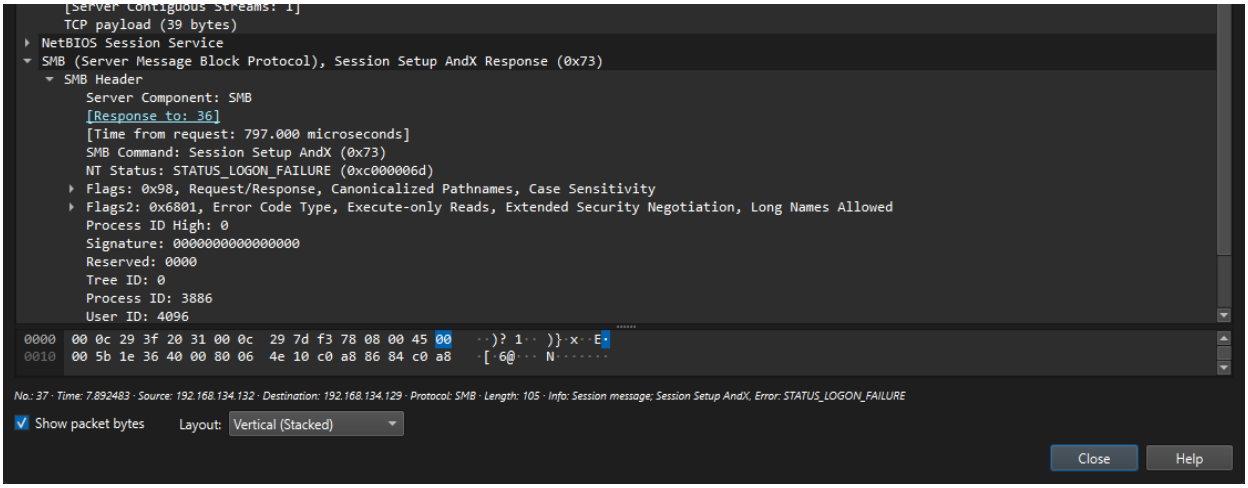

Figure 1

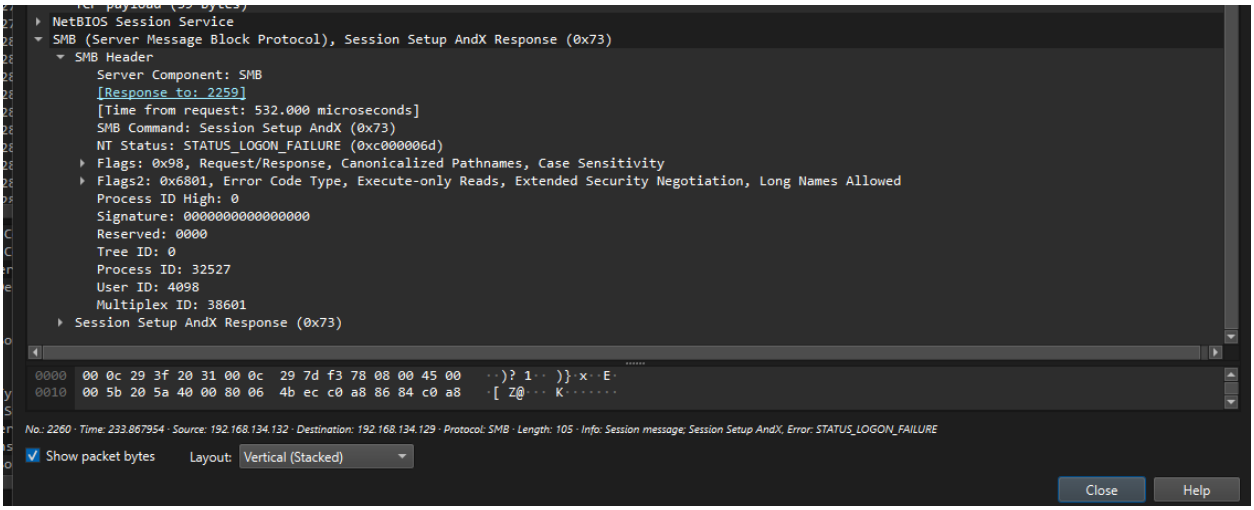

Figure 2

A little after the first failure, Frame 43 showed another SMB response, this time with STATUS_ACCESS_DENIED (see Figure 3). I saw the same STATUS_ACCESS_DENIED message in Frame 2266 as well (see Figure 4). An access denied error usually means the user account tried to access something it didn't have permission for. Seeing these in the network traffic told us that the attacker might have eventually found an account but still couldn't access certain resources or actions because the system was blocking them.

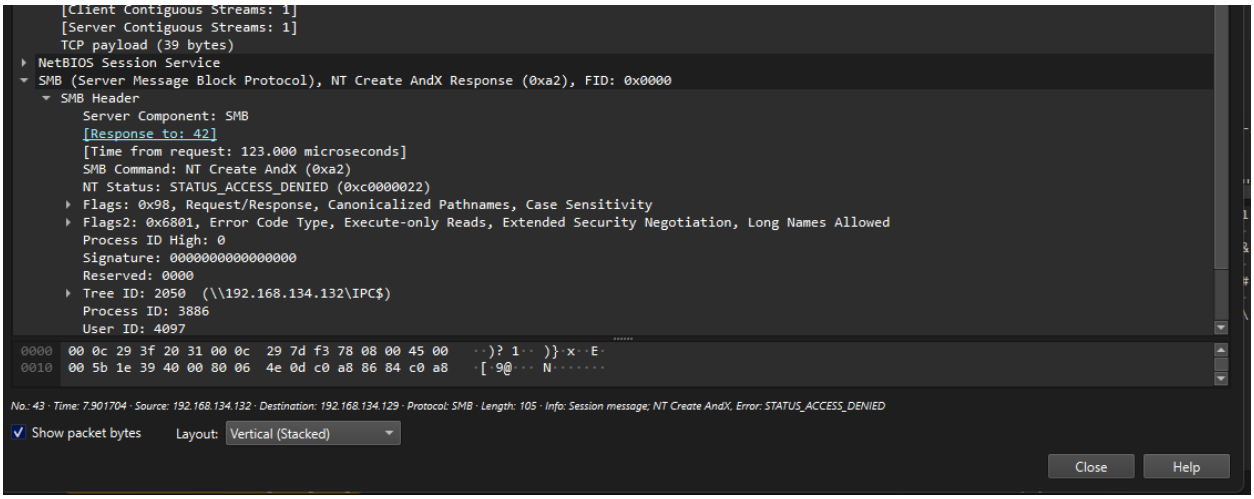

Figure 3

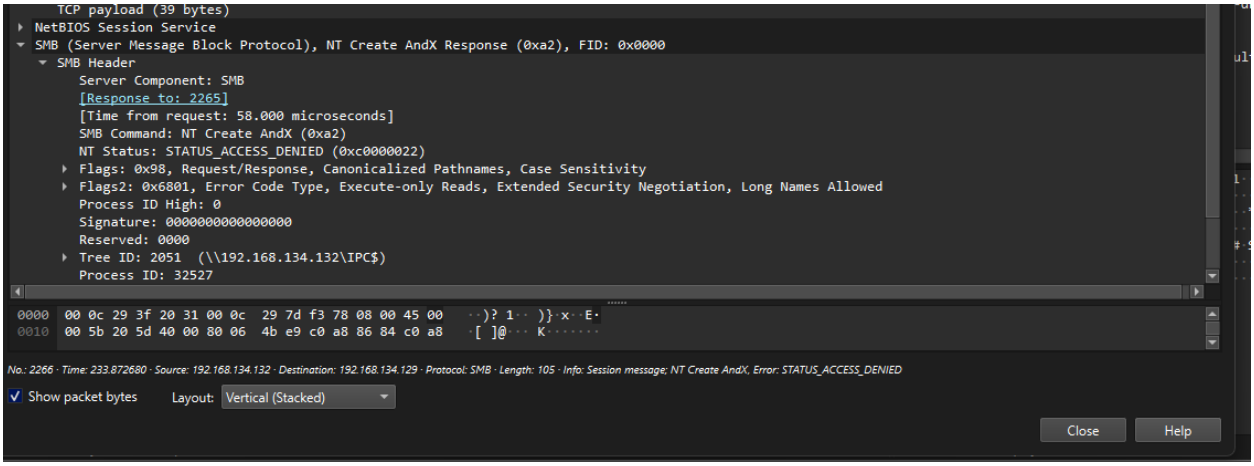

Figure 4

After packet analysis I moved onto Event Viewer. I moved on to Windows event logs to gather more evidence of the attacker's activities. Using Event Viewer, I looked at the application logs for any critical errors or warnings around the time of the suspected breach. I saw a WinMgmt warning event with Event ID 63 (see Figure 5) appearing around the incident timeframe. WinMgmt means Windows Management Instrumentation, so this caught my attention. Event ID 63 in WinMgmt often means there was an issue with a WMI provider or operation. This could mean the attacker tried to use WMI for some malicious purpose and after further research attackers sometimes use WMI scripts for persistence or to run code. So, it triggered a warning on the system.

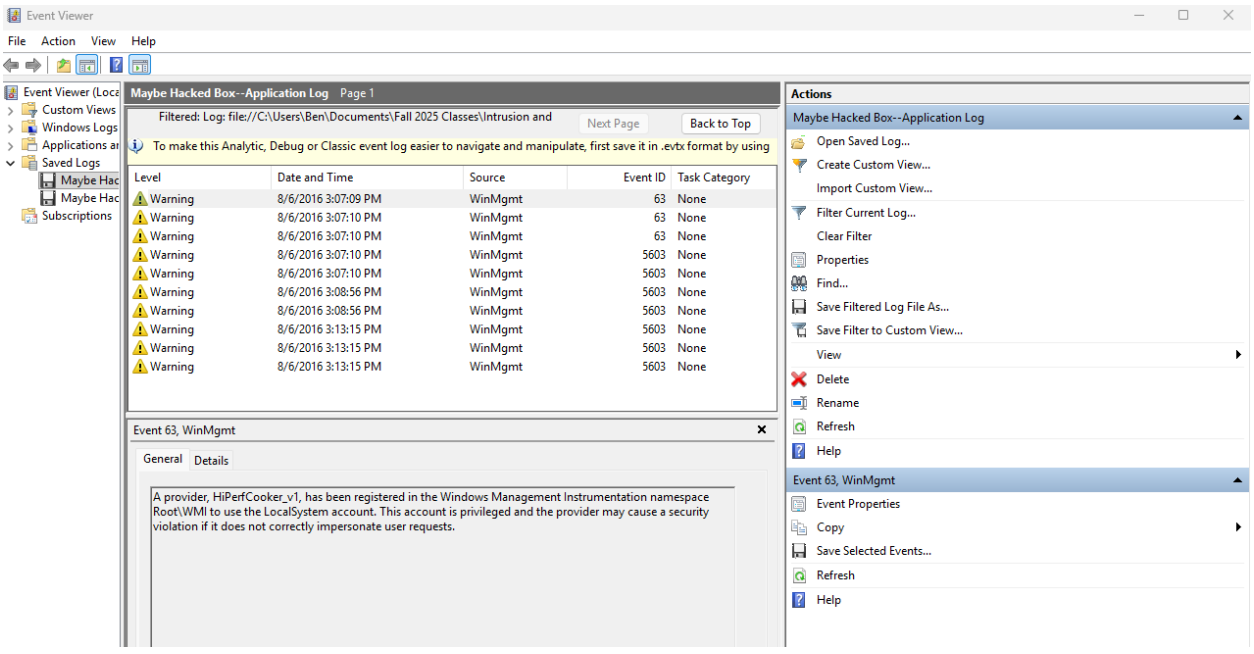

Figure 5

I also found several errors in the logs that had to do with the system services not starting correctly (see Figure 6). These were called Service Control Manager errors and showed up with codes like 7001 and 7026. One of the errors said that the IPSEC service could not start because a device was not working. Another error listed a bunch of drivers like AFD, TCP IP, and NetBT that did not load. All these errors happened around the same time, which makes me think the system was having problems during that moment. It might have happened right after a reboot or crash caused by the attack. A little before these errors, the logs showed that the system shut down unexpectedly (Event ID 6006) and then started up again (Event IDs 6005 and 6009). This tells me the computer probably restarted, and after that, some services had trouble starting up again. It is possible that whatever the attacker did made the system unstable or forced it to restart.

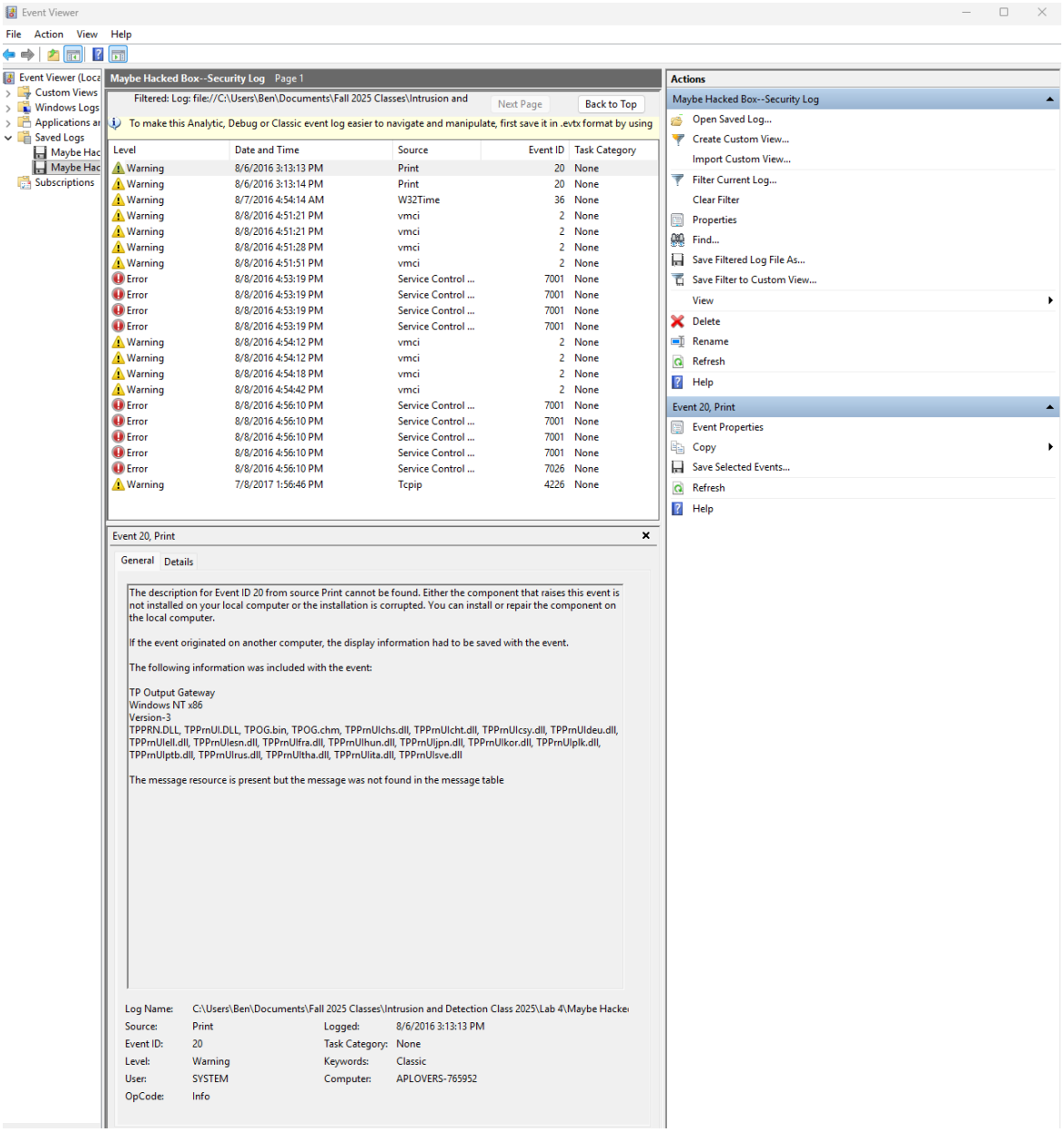

Figure 6

Timeline

Here I attempted to put together a timeline from my findings in Wireshark and the given logs in Event Viewer.

Before August 8, 2016, the system was running normally for the most part. But, on August 6, 2016, there was a WinMgmt warning, Event ID 63, in the application logs. This could have been an early sign of the attacker preparing something or it could be nothing I am not sure. I noted it here because WMI activity stood out when reviewing the logs around that period.

August 8, 2016, around 4:49 PM, the Event Log shows an unexpected stop which usually means the system went down or rebooted. Right after, at 4:53 PM, the system came back up and started logging errors. Multiple services failed to start like Event ID 7001 errors for various services and a 7026 error for drivers not loading. This is likely when the attacker's actions caused a system crash or forced reboot. Around this same time, the network traffic capture recorded the attacker's login attempts. In Wireshark Frame 37, an SMB login attempt was made with incorrect credentials, resulting in STATUS_LOGON_FAILURE. This might have been the attacker's first known attempt to log in but it failed.

A few moments later (August 8, 2016, around 4:50 PM), another SMB login attempt appeared in Wireshark, this time in Frame 43. Unlike earlier frames that showed STATUS_LOGON_FAILURE, this one returned STATUS_ACCESS_DENIED. That usually means the login itself might have been successful, but the user wasn't allowed to do whatever they were trying to do next like accessing a restricted folder or running a command. This could mean the attacker had some kind of working credentials but didn't have full permissions. It's also possible they were just testing what they could reach on the system. The timing of this failed access attempt is important because it lines up with other system errors and instability, which could suggest the attacker was actively probing or tampering with things.

Later on August 8, 2016, the same kind of activity happened again, which shows the attacker was still trying. In Wireshark, Frame 2260 showed another STATUS_LOGON_FAILURE, meaning someone tried to log in but used the wrong credentials or wasn't allowed. Just a few packets later, Frame 2266 showed STATUS_ACCESS_DENIED, which means a login might have worked, but they couldn't access what they were going for. This could mean the attacker was trying the same username and password on a different machine or trying to move deeper into the system. It's also possible they were just making more guesses to try and get in again. This time, there were no signs in the event logs that the system crashed or rebooted, so whatever they were doing didn't break anything. It just looks like they kept trying to get access or poke around the network.

Overall, based on the timeline, it seems like the attacker was active on August 8, 2016, mostly in the late afternoon. Their actions led to a system reboot and some service failures. They made several login attempts. Some of those failed, and some may have worked but only gave limited access. There is no solid proof that I could see they got full administrative access, but they were clearly trying to get further. They might have been testing different accounts or

seeing what parts of the system they could reach. It is possible that the attack actually started earlier. The WMI event on August 6 could be a clue. But most of the activity that stands out happened on August 8. After the reboot and the early login attempts, the later network traffic shows that they kept trying to connect to something else or do more. They may have kept going until someone noticed or blocked them.

Conclusion

Through analyzing the network traffic with Wireshark and reviewing the Windows event logs, I found strong signs that the system was under attack. The Wireshark data showed that someone was trying to log in over the network multiple times. I noticed specific SMB protocol responses like STATUS_LOGON_FAILURE and STATUS_ACCESS_DENIED, which pointed to failed login attempts and blocked actions. These errors matched up with key entries in the Windows event logs. For example, there were Service Control Manager errors, driver failures, and system reboot messages that all happened close together, which suggests the attacker caused instability on the machine.

There was also a WinMgmt Event ID 63 warning that might show the attacker tried to use WMI or other system tools during their activity. While I did not see proof of an administrator-level logon, the attacker clearly had some access and was trying to do more. Based on the pattern of events, they might have gotten in using stolen credentials or brute force, which would explain the login failures followed by access denied messages. Their actions seem to have caused the system to reboot, and after that, some services failed to load properly.

In the end, it looks like the attacker had limited success. They made multiple attempts, caused system problems, and tried to continue their activity, but there is no strong evidence that I could find that they achieved full control. The combination of network traffic and log data helps tell the story of what happened and gives a clearer picture of how the attack unfolded.

Recommendations

Even though I was able to find out what happened after the attack, it is always better to stop something like this before it gets too far. After going through the logs and network traffic, I think there are a few things the company can do to improve its security going forward.

Stronger Passwords and Login Rules

The attacker kept trying to log in over and over, so it seems like they were either guessing or using stolen passwords. To stop this, the company should make sure all accounts have strong passwords and that users cannot keep trying if they fail too many times. One thing we learned earlier in class was how password guessing or spraying can work, and this lab shows what it actually looks like when it happens. Turning on account lockout after a few failures and adding something like two step verification would make it much harder for someone to break in with just a password.

Better Log Alerts and Monitoring

In this lab, I had to look through the logs by hand to see the strange activity. In a real world situation, that would take too long. There should be alerts for things like too many login failures or strange errors showing up on the system. In class, we talked about tools that help monitor logs and send alerts when something is wrong, like SIEMs. Those kinds of tools would help catch something like this right when it starts instead of noticing after the damage is done.

Keep Systems Up to Date and Secure

The attack might have worked because something on the system was outdated or not configured right. Keeping everything updated with security patches can stop a lot of known problems. Also, turning off services that are not being used helps keep the system safer. Earlier in the semester we learned about hardening systems to make them less open to attack, and this situation is a good example of why that matters.

Use Network Rules to Limit Movement

Based on the network traffic, it looked like the attacker was trying to connect over the network. That means they might have gotten in through one machine and tried to spread out. One way to stop that is by setting up the network so that not every computer or device can talk to everything else. This is what we called segmentation back in class. If only certain machines can reach the important servers, then even if one gets hacked the rest stay protected.

Have a Response Plan and Train Everyone

It is also really important to be ready when something like this happens. The company should have a written plan for what to do, like who to call, how to shut down access, and how to fix the damage. Running through this plan with the team every so often can help people respond faster when something goes wrong. Employees should also be trained to spot things like strange emails or weird popups, since those are common ways attackers get in. This way, everyone can help keep the system safe.

Putting these ideas into action would make it a lot harder for something like this to happen again. Even if someone tried, having strong passwords, better logging, and a good response plan could either stop the attack early or help catch it before it gets worse. Doing this lab really helped me understand how everything we learned in class fits together and saw how important those topics are when it comes to real security problems in the real world.

References

- Cybereason. (n.d.). *Fileless Malware WMI*. Retrieved December 5, 2025, from https://www.cybereason.com/blog/fileless-malware-wmi

- Iritt, A. (2022, July 24). *Wireshark Essential Filters for Network Analysis*. Medium. https://iritt.medium.com/wireshark-essential-filters-for-network-analysis-a64239f44bfd

- Joshua R. (2022, August 28). *Event Logs and Wireshark Analysis Lab Walkthrough* [Video]. YouTube. https://www.youtube.com/watch?v=68t07-KOH9Y

- Microsoft. (n.d.). *WMI Start Page*. Microsoft Learn. Retrieved December 5, 2025, from https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

- Microsoft. (n.d.). *Microsoft Defender for Endpoint error and event codes*. Microsoft Learn. Retrieved December 5, 2025, from https://learn.microsoft.com/en-us/defender-endpoint/event-error-codes

- WiredMonkey. (2022, November 6). *How Hackers Use Event Logs in Windows* [Video]. YouTube. https://www.youtube.com/watch?v=r02IxoMeqFI

- Kent, K., & Souppaya, M. (2006). *Guide to computer security log management* (NIST Special Publication 800-92). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf

- Grimes, R. (n.d.). *Security Log Encyclopedia*. Ultimate Windows Security. https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/

- Orebaugh, A., Ramirez, G., & Beale, J. (2007). *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress.