**CYBR 450 – Operating Systems Projects**

NAME: \_\_\_\_Chad Ballay\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_12/20/2020\_\_\_

Chad Ballay

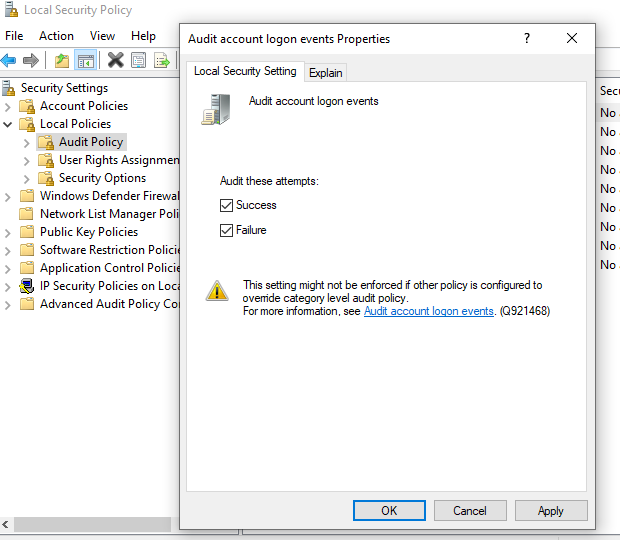
CYBR 450-342N

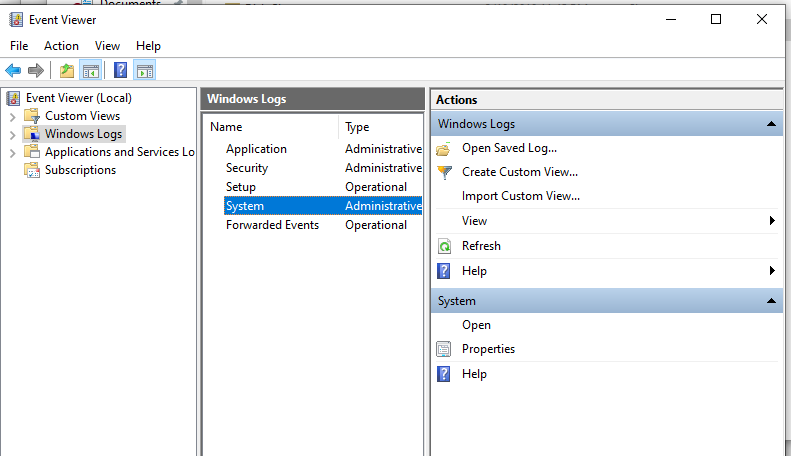
Week 5

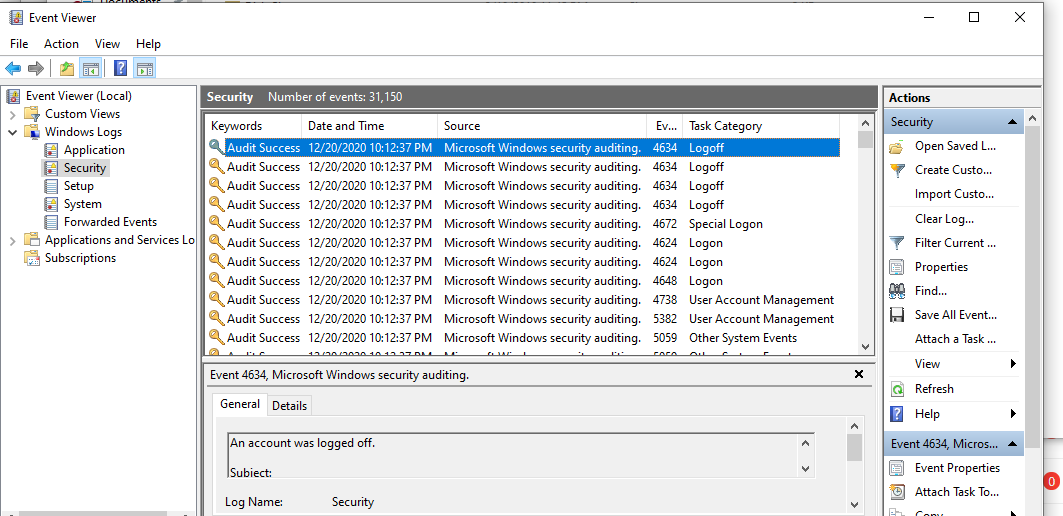
PROJECT 1

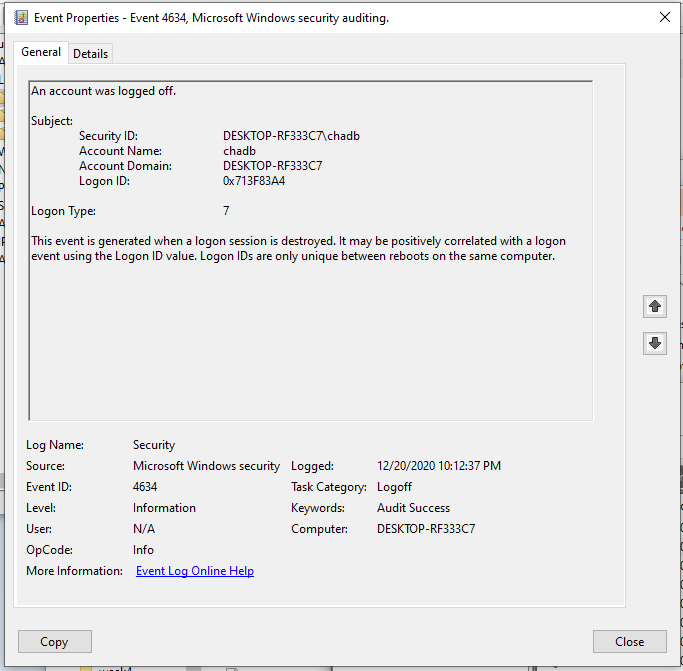
I come from a UNIX and OSX background so much of what was done for this project is new to me. *My first impression is that this is so much more nuanced than the out of the box auditing tools I have been exposed to for Linux as a desktop.* Centrify and all the other third-party tools have some features that can mirror this, but they are not built into the OS (Operating System) from the get-go like Microsoft has done with this. Syslog and that whole ecosystem feels so limited when compared to this.

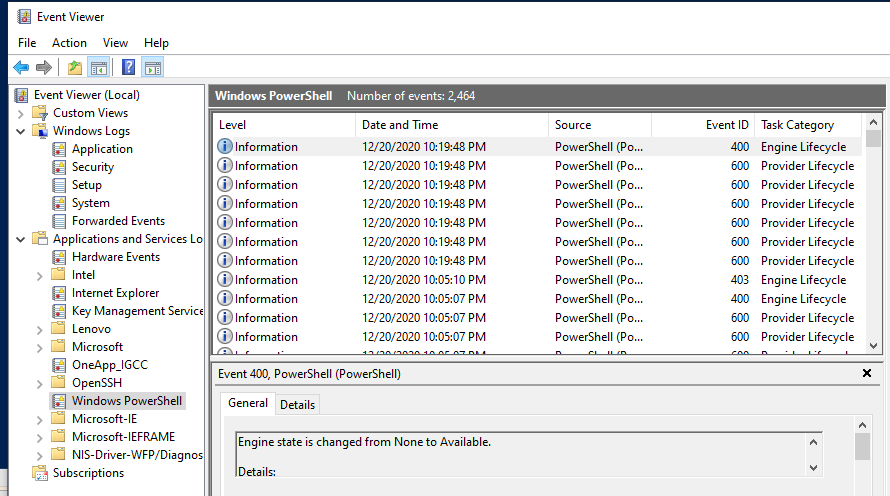
For the first part we are enabling the audit log entries for logoff and logon events to be captured. I would have expected these to have been on by default but there may be performance issues or other externalities for them to not be defaulted to enabled. Once enabled we trigger the events to occur by logging off and back on. This is to generate entries on demand. The individual entries have a lot of data but most of it I am unfamiliar with enough to understand. I did find some additional reading online, but I have not devoured it enough to master what I am seeing. <https://www.ultimatewindowssecurity.com/securitylog/book/page.aspx?spid=chapter5>











**PROJECT 2**

Center for Internet Security

<https://www.cisecurity.org/about-us/>

Apache Http Server 2.4 (Benchmark v2.0)

<https://learn.cisecurity.org/l/799323/2020-11-03/33kk8>

<https://security.uri.edu/files/CIS_Apache_HTTP_Server_2.4_Benchmark_v1.2.11.pdf>

\*=They have a horrible user experience for viewing the pdf’s they put out. It seems like their PR people are making technical decisions. An ugly cookie and session kludge for forcing people to turn over their contact info.

The referenced benchmark takes a user through the process of looking at a default install of Apache Http Server and walks them through several configuration changes that should be made to secure the installation.

It breaks the process down into major components: Installation, Modules, Permissions, Logging, etc., etc... Within these top-level groupings, a collection of specific recommendations is presented with each recommendation having a description, rationale, and very critically an audit check that can be performed to validate the recommendations implementation.

While researching this I came across several implementations where these recommendations have been automated. I am leery of taking random code from the wild and entrusting it to configure my own servers but it does illustrate what could be an internal project to be undertook when attempting to implement a robust CI/CD pipeline and other IaC set of best practices. By automating these steps we could have a secure server spun up in moments. Ensuring a consistent and repeatable environment build is a precondition to moving at speed when implementing a modern software development pipeline. <https://github.com/nikhil1232/apache-http-server-2.4-cis-benchmark-script>

One of the items that surprised me was the recommendation to Remove Default HTML Content. Not that it needed to be resolved but how thoroughly this walked you through removing it. Specific files to edit, using yum to remove the package, lines to change, etc., etc....

