**CYBR 450 – Access Control Assignment**

CYBR450-342

Chad Ballay

Week 4

12/13/2020

NAME: \_\_\_\_\_Chad Ballay\_\_\_ DATE:\_\_\_12/13/2020 \_\_\_

**Part 1**

**Current Example**



**Previously Used Example**



**Norton Generated One**



I felt pretty good at the results of my current password and a little embarressed at one of my previous ones. The difference is that now I use a password tool to create and track passwords uniquely for each site instead of just reusing the same password over and over. Even with the previous one saying 6 years that is uncomfortably short. Computation power only get cheaper as time goes on. Plus the website doesn't specify 6 years on what type of hardware. If they spin up 12 instances does that mean it'll be doable in 1/2 a year? Plus is that 6 years exhausting the whole key space? Statisitclaly you have a 50% chance of succeeding by only cracking N/2 keys form the total keyspace. All those items start to add some risk of previously captured data dumps being able to come back and haunt me within the lifespan of the data being still relevant. Maybe not crackable today but what about in two years? What if that data was a backup of some ssh keys or maybe a personal journal or some such other item that still poses a threat?

<http://www.experts123.com/q/why-do-the-statistics-measure-estimate-time-to-having-50-of-the-keyspace-searched.html>

I did check several other sites using the same password and recieved vastly different results. From one hour to multiple days. So the jury is out on how precise the time to crack is but anything less than a centurary is probably in need of being strengthened. Nonetheless, stronger is better than weaker. Arguing the precise time durations misses the point that relative to other passwords we can clearly see what makes them more of a problem to attackers.



<https://www.security.org/how-secure-is-my-password/>



<https://www.comparitech.com/privacy-security-tools/password-strength-test/#password-test-tool>



<https://password.kaspersky.com/>



<https://nordpass.com/secure-password/>

**Part 2**

Something you know, something you have, something you are... This is the mantra that was told to us in a previous class. It rings true simply because breaching this process requires failures of two seperate systems which a small percentage times a small percentage is a really small percentage.

To accomplish this, most organizations will implement 2FA based off of the user knowing a password and then having a secure token or authenticator app generating time limited PINS. For instance, you sign into CISCO AnyConnect's VPN with your userid and password. Once those credentials are verified then you are prompted in a phone app to acknolwedge the connection. In doing so an attacker would have to know your password as well as have access to your phone. People lose their passwords, people lose control of their phones but rarely do they lose both at the same time.

**Part 3**

Took some doing since my backgroun is Linux so right of the bat I learned that when all I have is a hammer, everything might look like a nail but that doesn't mean it is.

The windows mechanism looks a lot more userfriendly to me. Looking at the Account Policies, I can clearly see the requirements. It doesn't remember any previous passwords, every 42 days I have to pick a new one, and I can pick an extremely simple password. (These are the defaults that came with the system so somewhere in Microsoft some Hitchhiker's Guide fan is really enjoying that they got the maximum age to be 42.)



The Local Policy branch in this configuration tree causes more concern. Many of the options are unknown to me but it isn't hard to see that having them set to "Disabled" or "Not Defined" it probably means I'm a lot less strict than I could be. The complexity of the behaviors regulated also is daunting. Being tasked with controlling things I don't understand is a recipe for me to enable an overly permissive stance. Much more learning will be needed on my part to make heads or tails of this.

