Group #2 Case Study Report

Tyler Postma, Sai Parwatha, Logan Lyons, Zain Zahid

Central Michigan University

Course# 22445811: BIS521

Professor Helser

February 25, 2024

**Financial Statement Risks:**

The security and integrity of financial systems are essential when working in a field that handles financial reporting. Financial organizations face serious risks when they do not have strong password policies in place, and those risks can have a big impact on their financial statements. Having poor password policies lowers the barrier of entry for malicious actors to gain unauthorized access to financial systems which can lead to the complete compromise of financial data. Strong auditing practices are needed to determine how an organization is exposed to this risk. This may involve reviewing access control logs, permissions, and underlying organizational employment structure. By having strong auditing practices in place an organization can ideally spot any unauthorized access attempts or unusual user login behaviors.

Another serious concern associated with poor password policies is that underlying organizational data could be exposed. Financial data could be exposed, some of which could be sensitive for the organization, which could have serious ramifications on daily business operations. These ramifications could include harm to the company's reputation, legal issues, and financial difficulties. Any financial organization should periodically conduct a cyber security assessment, including penetration testing, to ascertain exposure to their overall risk factor. Through this audit process, exposure to breaches will be assessed, revealing information on how well access and other organizational security controls are working.

Fixing password policy flaws is essential to securing financial data and preserving the integrity and confidentiality of the underlying systems at an organization. By adopting strict password policies an organization can lower the risk of a system compromise. This combined with periodic testing will ensure that users are meeting policy requirements. Failing to manage these risks could have serious repercussions that go beyond just financial losses, affecting the organization's reputation and general stability. It could also jeopardize financial data.

**Password Policy Guidance:**

As part of the audit the team utilized guidance from the Center for Internet Security (CIS) Password Policy guide to provide a breakdown and recommendation for key topics regarding password security (Center for Internet Security, 2021). The results are as follows:

**Password Length Requirements:**

For accounts that use Multi-Factor Authentication (MFA), the CIS Password Policy Guide recommends a minimum password length of 8 characters, and for accounts that use passwords alone, it is 14. The maximum length of a password is not defined, so systems are free to set it based on their capabilities. Although forcing very long passwords can result in predictable and unsafe user behaviors like using repeating patterns, longer passwords are typically more difficult to crack. Therefore, stronger passwords can be encouraged without sacrificing usability by setting a reasonable minimum length without a maximum limit.

**Special character requirements in passwords:**

Require at least 1 special character. The CIS advises requiring at least one non-alphabetic character in the password complexity for accounts that only use passwords. There is not a set complexity requirement for accounts that use MFA. The use of special characters increases the complexity and unpredictability of passwords. When it comes to values, special characters such as #, $, %, and others have more options than just letters and numbers. This increases the quantity of password combinations that users can generate. 

**Capitalization requirements in passwords:**

Use at least 1 Uppercase letter. Users often apply capital letters predictably (e.g., capitalizing just the first letter) in response to these policies. Attackers anticipate this and configure password cracking attacks to account for common capitalization tricks. As a result, forced capitalization rarely improves real-world password security. Instead of mandated rules, consider providing general password guidance to users on varying their capitalization approaches. Recommend that they avoid capitalizing just the first letter, restrict use of all upper-case letters, and utilize some capital letters randomly throughout their passwords.

**Passwords on Deny Lists:**

Keep a list of common or compromised passwords and do not allow users to use those passwords. Adding previously used passwords for that account in the deny list could mitigate the risk of a previous password being used in a system should it be compromised. Comparing each password attempt against known bad passwords is a critical security control for reducing attack surface.

**Dictionary Words vs. Passphrases:**

Use longer passphrases instead of single dictionary words. Passphrases have greater length and entropy while being easy to remember. Dictionary words alone yield passwords that are extremely vulnerable to cracking attempts. Using multi-word passphrases have far more permutations and are not contained in dictionary lists. Also, long passphrases avoid easily guessed dictionary words while enabling stronger passwords that users can easily remember.

**Multi-factor Authentication:**

The use of Multi-Factor Authentication (MFA) is highly recommended to improve account security instead of only using passwords for authentication. MFA significantly increases the difficulty of unauthorized access as it requires a user to have something they know, a password, and something they have, such as a token code. MFA drastically improves account security at the cost of slightly increased user inconvenience. MFA is a critical line of defense across all sensitive systems and should be considered an expectation for accessing any sensitive system. The most common MFA techniques combine a password with a special one-time code that is sent by email, voice call, SMS, or authentication app. However, for accounts that require additional security, more sophisticated options like smartcards or security tokens are advised.

**Password Sample Audit:**

As part of the audit, we had a team member examine 30 users and the system allowed us to extract 30 passwords from it. This by itself is a massive red flag because the system either allows reversible encryption or stores the passwords in plain text. Beyond that though we discovered of those 30 users a variety of concerns that are listed in Table 3 below:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Username** | **Password** | **Concerns** |
| 1 | BowenM655 | xLY]d\_!gzp | No concerns |
| 2 | EPage919 | Tom\_Ere\_2k9 | Potentially has personal info and pattern |
| 3 | ChappellA672 | password123 | Common password, easy to guess |
| 4 | WalkerA614 | LisaMaria-9412 | Potentially has personal info and pattern |
| 5 | RamirezC557 | Password1! | Has length and complexity but weak for a dictionary attack |
| 6 | BarclayH562 | &K{EN^M5nVde7 | No concerns |
| 7 | McculloughL594 | Champions=1995 | Weak with a dictionary word and a year |
| 8 | LugoK615 | 8$}y-kDnKx)3 | No concerns |
| 9 | VassC676 | SuzieAndRocco | Potentially has personal info and pattern |
| 10 | WalkerS706 | L@n3y! | Has complexity but short length |
| 11 | VinceJ198 | Marines#1 | Has complexity but could be guessed |
| 12 | KnightA631 | daisy | Common password, dictionary attackable |
| 13 | NortonA410 | ilovecandy | Common password, dictionary attackable |
| 14 | SwanH279 | Maja&Hayden4ever! | Potentially has personal info, but has complexity |
| 15 | JanssenS126 | E=mc2 | Password length, easy to brute force |
| 16 | SwanH279 | Jennifer1! | Has complexity but could be guessed |
| 17 | JenkinsR89 | xLY]d\_!gzp | No concerns (other than it is used above) |
| 18 | HallidayM689 | Maja&Hayden4ever! | Has personal info so guessable, but also reused |
| 19 | DonnellyB513 | S@tbfflad1 | No concerns |
| 20 | NorrisM666 | teddybears | Common password, dictionary attack |
| 21 | ReddenZ703 | yC4}FvJ=qb>NS | No concerns |
| 22 | ReidK706 | C@veGirl93 | No concerns, decent complexity, and length |
| 23 | LiL867 | PA$$word11 | Common password with substitution |
| 24 | EllisM154 | }/\*!GAxT | No concerns |
| 25 | BakerD760 | babycakes | Common password, dictionary attack |
| 26 | BillingsN726 | >/CRe6}Uxn%EA | No concerns |
| 27 | LambertM256 | H@mish300595 | No concerns |
| 28 | ShafferN129 | {&TgVj3\*dbZUrWPf? | No concerns |
| 29 | BakerB512 | BHU\*8uhb | No concerns |
| 30 | WheelerW440 | xLY]d\_!gzp | No concerns, but repeated password |

**Data Analysis:**

To automate our analysis of passwords we chose to utilize the python programming language since it was permittable by the company. The following results were observed:

The first test was run to determine how many users had passwords with less than eight characters in length. Password length is a critical requirement for password security. The results of Test 4A are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Record | UserName | Password | Length |
| 15 | JanssenS126 | E=mc2 | 5 |
| 12 | KnightA631 | daisy | 5 |
| 10 | WalkerS706 | L@n3y! | 6 |

Password complexity was next analyzed by determining which passwords only contained letters, whether they be upper case or lower case. The results of Test 4B are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 25 | BakerD760 | babycakes |
| 12 | KnightA631 | daisy |
| 20 | NorrisM666 | teddybears |
| 13 | NortonA410 | ilovecandy |
| 9 | VassC676 | SuzieAndRocco |

The next test was written to determine how many users did not have mixed capitalization in their passwords. The results of Test 4C are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 25 | BakerD760 | babycakes |
| 3 | ChappellA672 | password123 |
| 12 | KnightA631 | daisy |
| 20 | NorrisM666 | teddybears |
| 13 | NortonA410 | ilovecandy |

After completing these basic password validation tests a new test was developed to check for repeating passwords in the system. Please note that in any large production system it is reasonable to assume that there will be repeating passwords in a system, however it should not be common. The results of Test 4D are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 1 | BowenM655 | xLY]d\_!gzp |
| 18 | HallidayM689 | Maja&Hayden4ever! |
| 17 | JenkinsR89 | xLY]d\_!gzp |
| 14 | SwanH279 | Maja&Hayden4ever! |
| 30 | WheelerW440 | xLY]d\_!gzp |

To further check password security, a test was written to determine how many users have never changed their password. There were two ways with this test that could be utilized to determine if a password had been changed, thus the change date field was utilized. The results of Test 4E are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 1 | BowenM655 | xLY]d\_!gzp |
| 17 | JenkinsR89 | xLY]d\_!gzp |
| 30 | WheelerW440 | xLY]d\_!gzp |

The next test was written to determine which passwords had been rotated within the last 90 days (about 3 months). In some organizations regular password changes are required despite being a cause for weaker passwords throughout an organization. The results of Test 4F are below:

|  |  |  |  |
| --- | --- | --- | --- |
| Record | UserName | Password | TimeSincePwdChange |
| 27 | LambertM256 | H@mish300595 | 174 |

After these initial tests were performed, we developed a test to compare all the existing passwords that were provided against about 2 million known passwords that were discovered in data breaches. The results of Test 5 are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 29 | BakerB512 | BHU\*8uhb |
| 25 | BakerD760 | babycakes |
| 3 | ChappellA672 | password123 |
| 19 | DonnellyB513 | S@tbfflad1 |
| 2 | EPage919 | Tom\_Ere\_2k9 |
| 15 | JanssenS126 | E=mc2 |
| 12 | KnightA631 | daisy |
| 27 | LambertM256 | H@mish300595 |
| 23 | LiL867 | PA$$word11 |
| 7 | McculloughL594 | Champions=1995 |
| 20 | NorrisM666 | teddybears |
| 13 | NortonA410 | ilovecandy |
| 5 | RamirezC557 | Password1! |
| 22 | ReidK706 | C@veGirl93 |
| 16 | SwanH279 | Jennifer1! |
| 9 | VassC676 | SuzieAndRocco |
| 11 | VinceJ198 | Marines#1 |
| 4 | WalkerA614 | LisaMaria-9412 |
| 10 | WalkerS706 | L@n3y! |

One final test was devised that analyzed all passwords and determined which were not considered problematic by our standards. These passwords were more than eight characters in length, they contained a number or symbol, contained uppercase and lowercase letters, were not in use by another user, the password was changed within the last 90 days (about 3 months), and was not in the list of breached passwords. The results of Test 6 are below:

|  |  |  |
| --- | --- | --- |
| Record | UserName | Password |
| 6 | BarclayH562 | &K{EN^M5nVde7 |
| 26 | BillingsN726 | >/CRe6}Uxn%EA |
| 24 | EllisM154 | }/\*!GAxT |
| 8 | LugoK615 | 8$}y-kDnKx)3 |
| 21 | ReddenZ703 | yC4}FvJ=qb>NS |
| 28 | ShafferN129 | {&TgVj3\*dbZUrWPf? |

Based on the tests performed, we strongly recommend an immediate password change be initiated on all users, including those output from Test 6. This will ensure that users have rotated their passwords which will reset the password reset time and ensure that the initial password is no longer used. Before doing that, the test should be expanded to include all users of the system rather than only the 30 that were sampled.

There were several concerns that were noted while these tests were being created. The organization states that they attempted to follow Center for Internet Security (CIS) Password Policy recommendations, however the CIS policy recommends a yearly password change once a year, not every 90 days (about 3 months) which defeats the purpose of that test (Center for Internet Security, 2021). Additionally, the system allows for a user to have a password set as the initial password without it being changed. If the user has not logged in to change that password that user should be disabled after a certain period. One other concern related to the audit itself. In any secure system it should not be possible to retrieve the values of the passwords in plain text for analysis. That a sample was able to be provided undermines the entire security concept. Access to the database where this password is stored would allow a malicious actor to gain access as any user to the system which violates auditing controls.

One last concern is password security. Passwords are inherently unsafe given that they can be brute forced, guessed, or leaked. Multi-Factor Authentication (MFA) can add additional protections, which was being leveraged at Liberty Data Systems, however a far superior approach would be the transition to a password less authentication system. Other authentication methods, such as certificate-based authentication (smartcards), could be utilized to add an additional layer of security that passwords cannot provide. Utilizing best practice password policies helps maintain the status quo, but to transition to a more secure framework passwords should be discarded.

**Reflection:**

**What part of the audit or analysis process surprised your team?**

One of the most surprising factors with the password audit had to relate to the underlying audit itself. It would be understandable for a penetration test to be performed to test for password security, however in a real-world system having access to plain-text passwords would violate the underlying concepts of confidentiality, integrity, and availability that the security field prides itself in. It was not surprising how many of the passwords were insecure in the system, even though it was a financial system.

**Which part of the case study did you find most interesting?**

It was interesting to see the depth to which the CIS Passwords Guidelines went. In this case study we focused on what CIS advised, however when doing independent research, it was interesting to see what other cyber security frameworks advised for their password guidance. It was interesting to see that password guidelines are still relevant in 2024 since authentication systems have been around for decades yet we still have insecure policies today for password management.

**What areas did you find most difficult to understand or complete?**

It was a bit challenging to develop the code in Python to meet each of the objectives. Overall, it was easy to read the data in from a CSV file, and even the XLSX file only required an additional Python module to parse. The difficult part though was formatting and outputting the data in an efficient way required the use of multiple data structures to organize and sort for the output. Had this data been stored in a database first SQL could have been utilized to structure the data for output and then Python could have looped through that data for a straightforward output. We wrote the code in such a way where it could be modularly loaded from test to test, however for some of the tests certain portions of the code had to be re-written to handle the new use cases which added additional complexity.

**What is the most important thing you learned from this case study?**

Even though MFA was utilized by Liberty Data Systems there was no excuse for the poor password security policies that were utilized by the employees. MFA does add an additional layer of security, however, should a password be compromised a malicious actor only needs to convince the user to give them their MFA code or accept the push notification and they can get access to underlying systems. It is also possible the password could be used for certain systems that allow for other forms of MFA Bypass attacks, therefore having a strong password is still paramount (Papez, 2023).

**How can you apply the knowledge and skills learned from this case study to another area or in the real world?**

The insights and skills gained from this case study can be used in various real-world scenarios. When working in any organization or cybersecurity role, it is important to understand and evaluate password policies, regular security audits, and implement strong controls to enhance the security of yourself and the firm.

References

Center for Internet Security. (2021). *White Paper: CIS Password Policy Guide*. CIS. <https://www.cisecurity.org/insights/white-papers/cis-password-policy-guide>

Papez, N. (2023, April 23). *MFA Bypass: How attackers beat 2FA - Blog | Menlo Security*. Www.menlosecurity.com. <https://www.menlosecurity.com/blog/the-art-of-mfa-bypass-how-attackers-regularly-beat-two-factor-authentication>