**Part A:**

On December 15th 2015, security researcher Chris Vickery submitted a post to Reddit claiming that he had found a security vulnerability in Kromtech’s user database for their MacKeeper software (Vickery, 2015). MacKeeper had a total of around 13 million users at the time of the breach and the database which was breached held all user information including name, email address, computer name and model, software licence and activation codes, public IP address, and usernames and passwords for each licenced product (Qureshi, 2015).

Vickery was using IoT search engine, Shodan, and searching for databases listening for incoming connections on port 27017 when he found the vulnerable MacKeeper database. Port 27017 is the default port value of mongos and mongod instances of MongoDB databases (MongoDB, 2017), which is what Kromtech were using for their MacKeeper database. The default port was open to external connections, and unfortunately for the company and its users, the database required no authentication for access. After Vickery gained access to the database, he found that the owner was Kromtech and contacted them to let them know of the security vulnerability.

Kromtech acknowledged the security breach and posted an update on December 17th stating that they had secured the database and upgraded their password hashing to a new hashing algorithm, although they did not specify which algorithm was implemented (MacKeeper, 2015). Vickery stated that the passwords that were being stored in the database were hashed using MD5 without salt (Vickery, 2015). The decision to store users’ passwords in this manner poses a great threat to users’ privacy and data, given that unsalted MD5 password hashes are relatively easy to crack using dictionary and rainbow tables (Kioon, Wang, Das, 2013, p. 2709).

The first vulnerability in this case is, as previously mentioned, the poor and insufficient methods used for the hashing of the users’ passwords. Finding plaintext which is equivalent to that of the hashed value, that has been hashed without salt, can be relatively easy. For instance, if two or more of the 13 million users have identical password hashes, it is a reasonable assumption that the password is likely to be a simple common password. During Imperva’s analysis of 34 million Facebook passwords, nearly 50% of users’ passwords were simple enough that they would be included in password dictionaries (Pfleeger, Pfleeger, & Margulies, 2015, p. 44). Given the user base of software such as MacKeeper, it is likely that this percentage would be similar or higher than that of Imperva’s findings of the Facebook passwords. Keeping this in mind, an attacker would then be able to run a dictionary attack on these accounts and given the identical hashes, would have a high likelihood of finding a match.

Another way that an attacker could attack unsalted password hashes is by using rainbow tables, which are tables of pre-computed hash values. This type of attack is much quicker than a hash table attack. The solution to defend against these types of attacks is to use salt when hashing with MD5, which will add a randomised string to the plaintext of the password before hashing. For this to be most effective, the salt will need to be different for each user and will be stored in the database along with the rest of the users’ data. Even though the salt will be easily accessible to an attacker who has gained access to the database, the identical passwords will no longer have identical hashed values and the salt protects against a rainbow table attack.

Even if the passwords were hashed with salt, Wang and Yu (2005) showed that the MD5 algorithm is vulnerable to differential attacks when he was able to find collisions as quickly as 15 minutes. Because of this vulnerability, MD5 is no longer suitable for password hashing. There are currently 3 algorithms which are known to be safe to use for hashing passwords, those algorithms are PBKDF2, bcrypt and scrypt. Using one of these algorithms instead of MD5 would be the first configuration change in order to solve the largest vulnerability in this case.

The second vulnerability is the fact that there is currently no authentication required to access the MongoDB instance. The resolution of this issue is relatively straightforward; authentication will need to be enabled on the MongoDB instance and then an administrator account with a strong password will be configured.

The third and last vulnerability is the default port used to listen for incoming connections. Databases using default ports are considered the low hanging fruit when attackers are searching for vulnerable databases. If the administrator has not reconfigured the database with a non-default port, it is likely that they are also not following other security standards as well. Reconfiguring to a non-default port is a technique which adds protection by obfuscation. Once these 3 vulnerabilities are rectified as described above, the database would have sufficient security.

**References**

Kioon, M. C. A., Wang, Z. S., & Das, S. D. (2013). Security analysis of MD5 algorithm in password storage.*Applied Mechanics and Materials, 347-350*, 2706. doi:http://dx.doi.org.ezproxy.csu.edu.au/10.4028/www.scientific.net/AMM.347-350.2706

**MacKeeper (2015, December 14). MacKeeper™ Security Advisory [Blog post]. Retrieved from** <https://mackeeper.com/blog/post/173-mackeeper-security-advisory>

**MongoDB. (2017). Default MongoDB Port -** MongoDB 3.4 Manual**. Retrieved from** <https://docs.mongodb.com/manual/reference/default-mongodb-port/>

Pfleeger, C. P., Pfleeger, S. L., Margulies, J. (2015). Security in Computing (Fifth Edition). Upper Saddle River, United States: Pearson Education.

**Qureshi, N. (2015). MacKeeper Hacked: Researcher Finds 13 Million Credentials Online. Retrieved from** <https://www.hackread.com/13-million-mackeeper-credentials-found-online/>

Vickery, C. [FoundTheStuff]. (2015, December 15). I have recently downloaded over 13 million sensitive account details related to MacKeeper, Zeobit, and/or Kromtech [Reddit post]. Retrieved from <https://www.reddit.com/r/apple/comments/3wq9fc/massive_data_breach/>

Wang, X., & Yu, H. (2005, May). How to break MD5 and other hash functions. *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 19-35). Springer Berlin Heidelberg. Retrieved from <http://merlot.usc.edu/csac-f06/papers/Wang05a.pdf>

**Part B:**

On October 18, 2015, the New York Post reported that an American high school student had hacked the private email account of the director of the CIA, John Brennan (Messing, Schram, & Golding, 2015). Once the teenager had access to Brennan’s email account, he found sensitive files some of which he publicly leaked to Twitter. The leaked information contained email addresses, phone numbers, social security numbers and clearance levels of John Brennan and more than 20 other CIA agents. The hacker describes other sensitive files, which were not publicly leaked, including a government letter describing “harsh interrogation techniques of terrorism suspects” (Messing et al., 2015), and a 47-page application from John Brennan requesting top-secret security clearance.

The hacker who only identified as CWA (Crackas with Attitude, referring to himself and a classmate), explained that the motivation for the attack was due to support for Palestine and his disapproval of US foreign policy. CWA was able to gain access to Brennan’s personal AOL email account by using a technique called social engineering in which the hacker uses manipulation of people, usually administrators or the target themselves to give passwords, or enough information allowing them to reset a password. Social engineering is known as the easiest type of attack (Pfleeger, Pfleeger, & Margulies, 2015, p. 50) and is used by 84% of hackers (Barth, 2017, para. 4).

CWA used social engineering on workers at Verizon and convinced them into disclosing Brennan’s personal information, which in turn further enabled social engineering of workers at AOL into resetting Brennan’s email password. With a newly reset password, CWA could log into Brennan’s email account, all without having to find any technical vulnerabilities. As evident in this case, the target of social engineering is not always the target of the attack itself, which can create trust issues between users and services. The target of this attack was John Brennan, but the outcome of the attack affected many more including the other 20 CIA agents whose details were leaked on Twitter, Verizon and AOL due to a flaw in their security system, and the US government itself due to damning evidence of interrogation techniques.

Social engineering is a very powerful tool because of the ubiquitous presence of social media and the immediate availability of personal information. By default, a Facebook profile’s privacy settings are set to display a user’s profile to “Everyone”, which essentially means that privacy is an opt-in setting. A user that is not overly savvy to social media or privacy will not know what settings they should choose, or even what those settings mean, and Facebook profiles like these are a gold mine to a hacker.

Security questions are often used for password recovery systems; these questions have been shown to be far less secure than a password (Bonneau, Burzstein, Caron, Jackson, & Williamson, 2015) and the availability of personal information on social media severely reduces the effectiveness of security questions as a security measure. In fact, 37% of users lie when giving answers to security questions believing that they will be increasing their security, however studies show that these lies are predictable and less secure (Bonneau et al., 2015). Even if the answer is correctly given and unique, Mark Stockley (2015) reported that a researcher named Chris Karlof was able to extract the security answers using phishing techniques from 92% of his targets.

There are several methods for authentication of a password recovery system, all with their own strengths and weaknesses. As previously stated, security questions are not reliable as a standalone authentication method, however they prove to be a decent first line of defence. To increase effectiveness of security questions, there should be a limit of guesses before the account is locked. An appropriate second line of defence in a password recovery situation would be to have the user then provide a token code via hard token, soft token, or another email address. This token code should have a time limit, so that the attacker does not have sufficient time to access this second stream of data. Each of these events should also trigger an email or SMS to the owner of the email address.

If an attacker is able to get past these methods, the next line of defence would be to implement two factor authentication on logon, either via hard token or soft token. Implementing these solutions is a balancing act of creating a system that is user friendly and also one that is secure for the user’s data and privacy. Both Verizon and AOL failed to keep the user’s account and data secure by not implementing any of the above solutions, or at least allowing a bypass of these systems.

Lastly, the director of CIA should not have been trusting a private company with the security of top secret federal information. The CIA have extensive network security so that information like this is not easily accessible by attackers. Serious backlash against government employees using private services for sensitive data is another way to prevent attacks like this in the future.

References

Barth, B. (2017). SCMagazine. Retrieved from <https://www.scmagazine.com/survey-explores-the-minds-of-hackers-81-claim-they-can-compromise-target-in-under-12-hours/article/640255/>

**Bonneau, J., Bursztein, E., Caron, I., Jackson, R., Williamson, M. (2015). Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google. Retrieved from** <https://research.google.com/pubs/archive/43783.pdf>

Messing, P., Schram, J., Golding, B. (2015, October 18). Teen says he hacked CIA director’s AOL account. *New York Post*. Retrieved from <http://nypost.com/2015/10/18/stoner-high-school-student-says-he-hacked-the-cia/>

Stockley, M. (2015) Yup, we really are terrible at those password recovery questions. Retrieved from <https://nakedsecurity.sophos.com/2015/05/26/yup-we-really-are-terrible-at-those-password-recovery-questions/>