CISC-680 – Software Engineering

Assignment No. 2

Due: 10/27/2019

Eric Webb

[ew774@mynsu.nova.edu](mailto:ew774@mynsu.nova.edu)

Dr. Jeffrey Kane

**Table of Contents**

**Cyber Breach Overview**

In this report, we are going to discuss one of the most prolific hacks in cryptocurrency history, the infamous Mt.Gox hack. This attack was the biggest bitcoin heist in history to date and occurred in a crucial time during the infancy stage of cryptocurrency development. This security breach would deal a blow to the crypto community that is still discussed today.

To begin, Mt.Gox actually stands for “magic the gathering online exchange” and was initially set up to do just that- act as an online exchange for a card game. (Tabbaa, 2018, P.3). After a while, the owner Jed McCaleb decided to switch from the online card exchange to being one of the first online crypto currency exchanges and quickly grew to handling around 70% of all the worlds’ bitcoin transactions in its prime. (Norry, 2019, P.1) The founder, Jed, sold Mt.Gox to Mark Karpelés in March 2011. The first hack on Mt.Gox should have served as a red flag to what was to come. It first happened in 2011 when a computer was compromised believed to be belonging to an Mt.Gox auditor. This attacker was able compromise user’s private keys and were able to access their online accounts. From there, the attacker sold the victims bitcoins for a penny on the dollar. (Norry, 2019 P. 8) Effectively stealing 2000 BTC’s for about $20. Aside from that, a former business partner, coinlab, would go on to sue the company for around $75 million, while on top of all this, the US government DHS would seize another $5 million for being an unregistered money transmitter. These were arguably red flags for the major hack that was simultaneously taking place covertly and unbeknownst to the exchange until years later. In 2014, Mt.Gox would discover that over the years around 850,000 BTC had been stolen from the exchange. (Tabbaa, 2018, P. 6).

**Security Failures**

How were these attackers able to steal Hundreds of Thousands of Bitcoins covertly under the radar from Mt.Gox without them noticing? It is still disputed on what truly happened to Mt.Gox, but the general consensus is that private keys stored in the online wallets were stolen because until the Bitcoin 0.4.0 release, the wallets private keys were un-encrypted. (Song, 2017, P.6). Anyone could have copied the wallet.dat files and viewed the plain text private keys, compromising the security of the wallets and their digital assets. Many people misconceive this attack as a quick process, but the attacker was much more dubious then that. This attack was prolonged over a period of time and comparable to a bucket of water with a hole in it. Because of the Mt.Gox stored key share, the attacker sent smaller portion of bitcoins to outside wallets confusing the website into thinking they were deposits being sent to more secure wallets. Mt.Gox had failed at keeping their private keys secure and auditing there stored key share. It is also said that Mt.Gox did not use the concept of version control and test environments properly. On top of that, it is said that changes were only approved from the owner and would take a great deal of time to get approved. (Croft, 2014, P.2).

**Cybersecurity Posture**

From the discussion above, we can see that Mt.Goxs weak security posture ultimately led to its demise. In this component, we will review and discuss some more secure concepts that could and should have been implemented. To begin with the technical aspect Mt.Gox used md5sum for hashing passwords when it was widely agreed upon by that time that this was an insufficient encryption standard. By that time sha-2 should be used for encryption while md5sum should be used for integrity rather than confidentiality. Meaning you should use md5sum for creating and validating checksums rather than hiding data with encryption. (Tabbaa, 2018, P.9). When hackers were able to access the administrative account, Mt.Gox faced their next security posture issue of having too many privileges. Remote accounts should have had read-only access along with bad management on the security team.

To put it simply, Mt.Gox should have had contracted out to a trusted penetration testing team to audit their posture. Hopefully finding their vulnerability of the weak md5sum, lack of a good cybersecurity management practice, and might have introduced the concept of secure agile methodologies. For instance the concept of the keypool having too many key shares, this lead to too many available hot wallets for the perpetrator to access. This should have been locked down by removing excess wallets and amounts in the keypool by sending the digital assets to securely stored offline cold wallets. This could have alerted at an earlier time that finances were going out faster than they were coming in. Second, hopefully the penetration team could have found vulnerabilities in their postures such as SQL injection or cross-site scripting that could have been nefariously used to take advantage of the weak md5sum hashed passwords. This could have been prevented with stronger input validation, better password encryption, and an outside security audit. When it came to secure software development practices they could have implemented some type of version control such as git. (Tabbaa, 2018, P.12).They should have had a dedicated quality assurance team to verify invalid user inputs and regression testing using products such as Selenium or Mocha for testing and validation. As for a merge request being approved from the top, the concept of committee based review could have been implemented for integration from test environments to production. This could have been implemented with products such as JIRA to build story boards for people to vote on whether things got approved or not eliminating the impossible task of one person reading copious amounts of code for review and approval. The concept of continuous integration and continuous deployment could have cut out a lot of unnecessary data availability and human resources. Using tools like Jenkins for CI/CD you could have saved resources by having code automatically test itself constantly and deploy the final production if everything passes simultaneously saving money and access to admin accounts because you would no longer have to pay so many workers to do it and thus less access to your admin rights.

Ultimately, Mt.Gox could have been better served by following more secure Agile / Scrum methodologies while also adhering to best security practices such as a more updated password hashing algorithm. The concept of 2-Factor authentication could have played a positive role in this as well such as RSA tokens or a google authenticator. That way if a private key was compromised you would need an additional factor to access assets.

**Business Continuity Plan**

When it comes to business continuity, the truth is, it’s a little opaque because this attack was estimated to be prolonged over about 3 years. (Song, 2017, P.1) Most arguably having enough funds saved in cold wallets and setting a threshold on when serious investigations on the business structure should be assessed should have been implemented. Essentially to set an alert to be notified that way too much money is going out and savings are needing to be accessed. Once they learned there was an issue they should have immediately been vocal and very public about it with their customers. This could have led to many users removing their holdings and employing better security practices such a cold wallet storage and two factor authentication. Holding back information about these issues is a big argument on what ultimately led to their demise. As far as continuing business a good insurance policy definitely could have helped financially. Technically they should withdrew all company wallets to cold storage and paused withdrawals from consumers till the security breach was located and stopped. Effectively keeping the business alive but no more money going out till the situation was under control.

**Conclusion**

In conclusion, the Mt.Gox hack was a pivotal moment for crypto history being the largest theft of BTC to date. (Tabbaa, 2018, P. 6). Starting as card game website to becoming the world’s largest crypto exchange sure had its rollercoasters of ups and downs for this company. It taught us that assets stored online on in hot wallets are not truly under your control and should be stored in cold wallets. It taught us the importance secure web development methodologies for continuous integration and deployment of code with automation and peer review. (Bookchin, 2019, P.2). I emphasized the fact that a company should be vocal and loyal to its customer base. It even helped to the development of more secure crypto wallet methodologies by encrypting wallets private keys now being an industry standard. Mt.Gox was an early mover and shaker of the crypto community with such a story of growth from small grass roots to being one of the biggest gainers and crashers of its market. As crypto currency finds its place in the worlds markets and gains value it will be pretty unfathomable that so much Bitcoin was stolen at a time when its value would never be imagined to be so great. This is why that Mt.Gox hack should really be the de-facto standard when it comes conversations about cryptocurrency and web development.

**Work Cited.**

Bookchin, A. (2019, February 13). Mt.Gox Source Code Too Simple Even for a Newbie. Retrieved July 14, 2019, from <https://cointelegraph.com/news/mt_gox_source_code_too_simple_even_for_a_newbie>

Croft, J. (2014, March 04). Mt. Gox Didn't Use Version Control, Source Code Was Controlled By CEO. Retrieved July 14, 2019, from <https://reckoner.com.au/news/mt-gox-didnt-use-version-control-source-code-was-controlled-by-ceo/>

Norry, A. (2019, June 07). The History of the Mt Gox Hack: Bitcoin's Biggest Heist. Retrieved July 11, 2019, from <https://blockonomi.com/mt-gox-hack/>

Song, J. (2017, August 12). Mt. Gox Hack Technical Explanation - Jimmy Song. Retrieved July 11, 2019, from <https://medium.com/@jimmysong/mt-gox-hack-technical-explanation-37ea5549f715>

Tabbaa, B. (2018, October 29). The Mt. Gox Hack. Retrieved July 11, 2019, from <https://medium.com/@bishr_tabbaa/the-rise-and-fall-of-mt-gox-whats-in-your-bitcoin-wallet-bd5eb4106f4e>

**Certification of Authorship**



Submitted to: Professor Yair Levy.

Student’s Name: Eric Webb

Date of Assignment: 10/27/2019

Title of Assignment: Assignment No. 2

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB