2019 Armor Games Breach

By Adonis Pineda

What is Armor Games?

Armor Games is a host and platform that was known for its massive library of indie flash games, often made by ordinary users or small-time studios.

It provided them a space where they could monetize their craft without having to publish to a bigger site or console.

As of now, Adobe Flash is soon to be unsupported and will have to be manually enabled by most browsers. There are no new updates for it; stability, security, or otherwise.

The Breach

Around early December 2018 through late January 2019, it was rumored that there was a string of massive breaches on 15 other companies besides AG, including Whitepages and Dubsmash

- ➔ **What was compromised?**
  Usernames, IP addresses, email addresses, and hashed passwords with a one-time password salt of users and creators, as well as executive staff

- ➔ **Why?**
  A cache of the stolen data was posted on Dream Market and Illegal Empire on the DeepWeb for sale.

Why would someone want this data? How would it be useful?

The data compromised would only allow access to basic information, besides access to the account.

More importantly, because of password recycling(using the same password for every site), enterprising identity thieves could use the associated email and hit any significant site they could with the password. Things such as online banking could be compromised as a result.

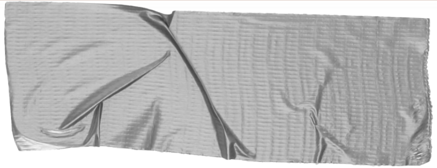
The monetary gain could be from the resale of the raw data dump, or the targeting of insecure account practices.

Tip

Do not use the same passwords for every site you register an email to. In fact, it is more secure to compartmentalize certain emails dedicated to certain accounts. At the very least, have strong and unique passwords.

How could this happen?

The intrusive entity was focusing on soft targets for the collection of viable compromising data. The attack was part of a large-scale campaign, seemingly preying on newborn or small tech companies and firms that have yet to cement good security practices and lessons learned. The one thing that the large majority shared was that too much of the information was stored within “reach” of another attribute(IP addresses in the same table as email, password salts with the hashed password). In other words, it was made too easy. Another point of failure was the lack of up to date encryption, as some of the encrypted accounts were done with MD5, which is nearly obsolete, and they were kept that way while newer accounts would use a slightly more advanced or new encryption, such as SHA(still behind the time zone).



Potential Attack Vectors

This is merely speculative, as Armor Games never released an official statement as to how the breach occurred, but did confirm an instance of intrusion as early as January 2019.

➔ **Cross-referencing**
The attackers may have made accounts and compared the changes to tables they had access to. This would allow them to “map” and have a visual understanding of how data is stored and what is significant by size and the speed of the return of queries.

➔ **Spear-phishing**
AG did confirm the compromising of more than one executive staff, which may have been specifically targeted for their privileges, in order to fell the operation in one axe-swing.

Whose fault is it?

The breach could be the result of human error, as any mistake can be. However, the manner in which the data was disseminated on deep web dump markets and the state of the data definitely points to remote access stemming from insecurity in software, but more importantly security practices(or lack thereof).

Relying on outdated encryption methods, neglecting to update the encryption on legacy accounts, lax password requirements, lack of MFA, and external communication in business matters may have played a role in the breach.

Likely, old software with known fundamental issues was the problem, evident in the use of an unsupported and soon to be derelict Adobe Flash. This is not to say Flash was the issue, rather a symptom of the system.

How can this be prevented in the big picture?

Keep encryption up to date! Keep up with new standards in encryption. This evolves every day. MD5 >> SHA >> Argon and the like

Do not respond to suspicious emails and report any unusual ones.

Require strong passwords and warn against password recycling. Suggest password changes immediately after any security concern.

Practice good security in account information: compartmentalize everything. It's good for organization, retrieval, and only provides what is necessary when requested.

How can I prevent this as a programmer? Can I?

Forcibly require users to change passwords after breaches and require strong and complex passwords. Only allow your code to manipulate the information necessary in a function. Keep up with the newest standards and read the details of a patch or new implementation. Test for expected and unexpected results. Test again. Make use of new libraries and standards of coding. Ensure that your team is aware of your changes and can understand how your code affects their end. Only provide what is n~~ecess~~ary to fulfill your end, even if it seems inconvenient.

Works Cited

Data Breach Notification 2019. (2020, November 23). Retrieved November 17, 2020, from https://armorgames.com/page/data-breach-2019

Nidecki, M., Nidecki, T., Grant Ho | 2 days ago, Avesta Hojjati | 3 days ago, Nicole Bucala | 4 days ago, Richi Jennings | 2 days ago, . . . 16, R. (2019, July 02). Insecure Default Password Hashing in CMSs. Retrieved November 17, 2020, from https://securityboulevard.com/2019/07/insecure-default-password-hashing-in-cmss/

Williams, C. (2019, March 04). 620 million accounts stolen from 16 hacked websites now for sale on dark web, seller boasts. Retrieved November 17, 2020, from https://www.theregister.com/2019/02/11/620_million_hacked_accounts_dark_web/

Thank you!