Data Breaches Using Credential Stuffing

Recently US car manufacturer GM disclosed that it was the victim of a credential stuffing attack last month that exposed some customers' information and allowed hackers to redeem rewards points for gift cards.

General Motors operates an online platform to help owners of Chevrolet, Buick, GMC, and Cadillac vehicles manage their bills, services, and redeem rewards points.

Car owners can redeem GM rewards points towards GM vehicles, car service, accessories, and purchasing OnStar service plans.

GM disclosed that they detected the malicious login activity between April 11th and April 29th, 2022, and confirmed that the hackers redeemed customer reward points for gift cards in some cases.

When the hackers successfully breached a GM account, they could access certain information stored on the site. This information includes the following personal details:

- First and last name,
- personal email address,
- personal address,
- username and phone number for registered family members tied to the account,
- last known and saved favourite location information,
- currently subscribed OnStar package (if applicable),
- family members' avatars and photos (if uploaded)
- profile picture
- search and destination information.

Credential stuffing attacks are one of the most common causes of data breaches because 65% of all people reuse the same password on multiple (and sometimes all) accounts.

The opportunity for cybercriminals to use credential stuffing is only growing as more credentials are exposed through breaches; at present, literally billions of compromised credentials are circulating on the dark web.

However, credential stuffing attacks are preventable if you implement the right cybersecurity measures. Below is what executives need to know about credential stuffing attacks and what can be done to reduce or prevent the likelihood that their organizations will be a victim of one.

What Is Credential Stuffing

Credential stuffing is a cyberattack method in which attackers use lists of compromised user credentials to breach into a system. The attack uses bots for automation and scale and is based on the assumption that many users reuse usernames and passwords across multiple services. Statistics show that about 0.1% of breached credentials attempted on another service will result in a successful login.

Credential stuffing is a rising threat vector for two main reasons:

- Broad availability of massive databases of breach credentials, for example, “Collection #1-5” which made 22 billion username and password combinations openly available in plaintext to the hacker community.
- More sophisticated bots that simultaneously attempt several logins, and appear to originate from different IP addresses.

These bots can often circumvent simple security measures like banning IP addresses with too many failed logins.

Credential Stuffing vs. Brute Force Attacks

Credential stuffing is similar to a brute force attack, but there are several important differences:

- Brute force attacks try to guess credentials with no context, using random strings, commonly used password patterns or dictionaries of common phrases
- Brute force attacks succeed if users choose simple, guessable passwords
- Brute force attacks lack context and data from previous breaches, and so their login success rate is much lower

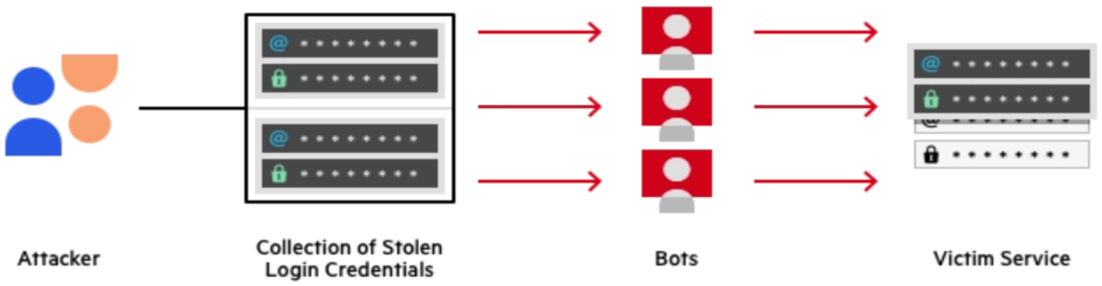
In a modern web application with basic security measures in place, brute force attacks are likely to fail, while credential stuffing attacks can succeed. The reason is that even if you enforce strong passwords, users may share that password across services, leading to a compromise.

How Credential Stuffing Attacks Work

Here is a typical process followed by an attacker in a large-scale credential stuffing attack. The attacker:

1. Sets up a bot that is able to automatically log into multiple user accounts in parallel, while faking different IP addresses.
2. Runs an automated process to check if stolen credentials work on many websites. By running the process in parallel across multiple sites, reducing the need to repeatedly log into a single service.

3. Monitors for successful logins and obtains personally identifiable information, credit cards or other valuable data from the compromised accounts.
4. Retains account information for future use, for example, phishing attacks or other transactions enabled by the compromised service.



Credential stuffing attack example

Credential Stuffing Prevention Measures

The following measures can help you protect your website from credential stuffing attacks.

Multi-Factor Authentication (MFA)

Requiring users to authenticate with something they have, in addition to something they know, is the best defense against credential stuffing. Attacker bots will not be able to provide a physical authentication method, such as a mobile phone or access token. In many cases, it is not feasible to require multi-factor authentication for an entire user base. If so, it can be combined with other techniques, for example, MFA can be applied only in combination with device fingerprinting.

Use a CAPTCHA

CAPTCHA, which requires users to perform an action to prove they are human, can reduce the effectiveness of credential stuffing. However, hackers can easily bypass CAPTCHA by using headless browsers. Like MFA, CAPTCHA can be combined with other methods and applied only in specific scenarios.

Device Fingerprinting

You can use JavaScript to collect information about user devices and create a “fingerprint” for each incoming session. The fingerprint is a combination of parameters like operating system, language, browser, time zone, user agent, etc. If the same combination of parameters logged in several times in sequence, it is likely to be a brute force or credential stuffing attack.

If you use a strict fingerprint with multiple parameters, you can enforce more severe measures, like banning the IP. To capture more attacks, you can use a combination of 2-3 common parameters, and enforce less severe measures like a temporary ban. A common fingerprint combination is an Operating System + Geolocation + Language.

IP Blacklisting

Attackers will typically have a limited pool of IP addresses, so another effective defense is to block or sandbox IPs that attempt to log into multiple accounts. You can monitor the last several IPs that were used to log into a specific account and compare them to the suspected bad IP, to reduce false positives.

Rate-Limit Non-Residential Traffic Sources

It is easy to identify traffic originating from Amazon Web Services or other commercial data centres. This traffic is almost certainly bot traffic and should be treated much more carefully than regular user traffic. Apply strict rate limits and block or ban IPs with suspicious behaviour.

Block Headless Browsers

Headless browsers such as PhantomJS can be easily identified by the JavaScript calls they use. Block access to headless browsers because they are not legitimate users, and almost certainly indicate suspicious behaviour.

Disallow Email Addresses as User IDs

Credential stuffing relies on the reuse of the same usernames or account IDs across services. This is much more likely to happen if the ID is an email address. By preventing users from using their email address as an account ID, you dramatically reduce the chance of them reusing the same user/password pair on another site.

Armoryze Security Solutions

Armoryze industry-leading bot management solution implements all the best practices above to protect against malicious bots. Moreover, it adds a layer of automated security logic, to prevent credential stuffing, carding, ticketing, and many other automated attacks performed via malicious bots.

In addition to malicious bot protection, Armoryze provides multi-layered protection to make sure websites and applications are available, easily accessible and safe. Our application security solution includes:

DDoS Protection—maintain uptime in all situations. Prevent any type of DDoS attack, of any size, from preventing access to your website and network infrastructure.

CDN—enhance website performance, cache static resources and reduce bandwidth costs with a CDN.

WAF—cloud-based solution permits legitimate traffic and prevents bad traffic, safeguarding applications at the edge. Gateway WAF keeps applications and APIs inside your network safe.

API Security—protects APIs by ensuring only desired traffic can access your API endpoint, as well as detecting and blocking exploits of vulnerabilities.

Account Takeover Protection—uses an intent-based detection process to identify and defends against attempts to take over users' accounts for malicious purposes.

RASP—keep your applications safe from within against known and zero-day attacks. Fast and accurate protection with no signature or learning mode.

Contact Us for a free consultation, demo and trial.

Website: www.armoryze.co.uk

Email: info@armoryze.co.uk

Tel: +44 – 0208 427 1131

Source References:

1. https://www.bleepingcomputer.com/news/security/gm-credential-stuffing-attack-exposed-car-owners-personal-info/

2. https://auth0.com/blog/what-is-credential-stuffing/

3. https://www.imperva.com/learn/application-security/credential-stuffing