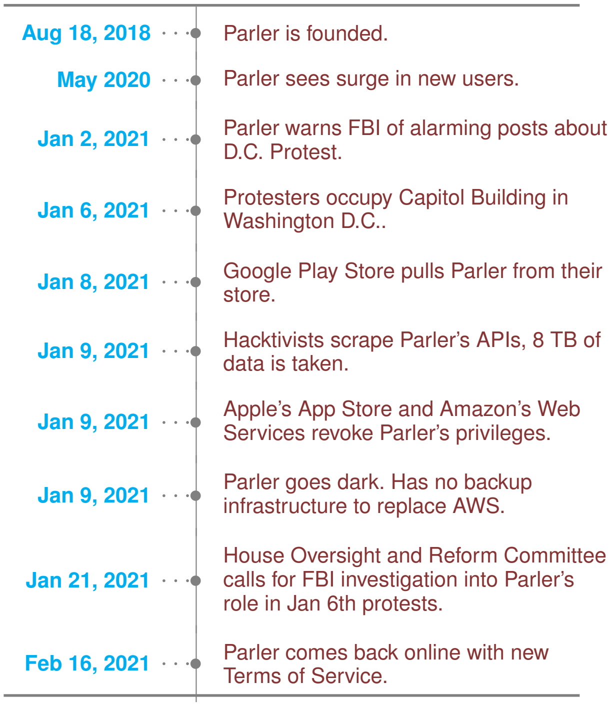
A Case Study of Massive API Scrapping: Parler Data Breach After the Capitol Riot

David Redding, Jian Ang and Suman Bhunia
Department of Computer Science and Software Engineering, Miami University, Oxford, Ohio, USA 45056
Email: reddindm@miamioh.edu, angj@miamioh.edu, bhunias@miamioh.edu

***Abstract*—After the United States Capitol Hill Riots, there was a massive API scraping of Parler, an open social media platform, which resulted in 70 terabytes of user data being collected. The data breach, a serious confidential personal data leak, was not performed illegally. This paper analyzes the data breach and its impact in depth. The breach was a result of a hacktivist going with the alias *@donk_enby*, performing a massive API scraping of Parler's servers. The scraping took metadata from user's public, private, and previously deleted posts, uploaded to Parler's servers. Parler had failed to clear the metadata of these posts. The metadata contained names, dates, locations, and other data about the users who posted content to Parler's site. Over 70,000 GPS locations of Parler's users have been uncovered including users' private properties. These locations have also been used to tie citizens to the Capitol Riots if they uploaded any content about the riot from that day. Forms containing government identification of users were also leaked from Parler's servers that were used for account verification. This paper demonstrate background on the events leading up to, including, and following the Capitol Riots. The paper also examine the hacktivist's methodology for performing the API scraping and discuss possible defensive strategies such as API rate limiting, API request sanitation, and API call authorization.**

***Index Terms*—Parler, Data Breach, API Scraping, Metadata, Capitol Riot**



Aug 18, 2018	Parler is founded.
May 2020	Parler sees surge in new users.
Jan 2, 2021	Parler warns FBI of alarming posts about D.C. Protest.
Jan 6, 2021	Protesters occupy Capitol Building in Washington D.C..
Jan 8, 2021	Google Play Store pulls Parler from their store.
Jan 9, 2021	Hacktivists scrape Parler's APIs, 8 TB of data is taken.
Jan 9, 2021	Apple's App Store and Amazon's Web Services revoke Parler's privileges.
Jan 9, 2021	Parler goes dark. Has no backup infrastructure to replace AWS.
Jan 21, 2021	House Oversight and Reform Committee calls for FBI investigation into Parler's role in Jan 6th protests.
Feb 16, 2021	Parler comes back online with new Terms of Service.

Figure 1. Timeline of Parler Attack

I. Introduction

Parler was founded in 2018, marketing itself as a free speech platform that would allow most content on its site. The platform was seen by some as an alternative to the "strict" oversight of other social media platforms that make up big tech companies like Facebook/Instagram, Twitter, and YouTube [1], [2]. It wasn't until 2020 and *Big Tech's* censorship of politically minded individuals, including top government leaders, that Parler, as a social media service, started seeing large numbers of users [3]. With Parler's seemingly lax rules of content mediation, some extreme groups were seen to take up residency on the application. These groups shared vulgar messages as well as some far-fetched ideas and conspiracies giving the company and the application a bad name.

Parler's troubles only multiplied as political tensions rose nearing January 6th. This is when the United States Congress convened to count the electoral votes of the presidential election [4]. In light of events leading up to this date, with claims of election fraud, members gathered in Washington D.C. in support of the incumbent president. Thousands made the trek to the U.S. Capitol Building and occupied the building for several hours. Many have made claims that these actions were organized through Parler's platform due to the composition of users on the platform [5].

On January 8th, 2021, considering Parler's potential role in the political events in Washington D.C., Google Play Store removed the application from its store. This meant that android users could no longer install or update the application on their device. Following this trend, on the next day, Apple's App Store removed Parler from their store as well. Amazon Web Services (AWS) also decided to no longer host the social media service on its cloud servers. AWS cited Parler's lack of moderation and allowing content that went against AWS's terms of service (ToS) as reasons for removal [6]. With Parler losing access to AWS's cloud servers, they had no infrastructure to host their services. Parler had to go dark and users could no longer access these services.

Before the exit of Parler on AWS cloud services, a single hacktivist with the Twitter handle, *@donk_enby* with unknown intentions (allegedly belonging to opposition party) used public APIs and false accounts to scrape as much data and code

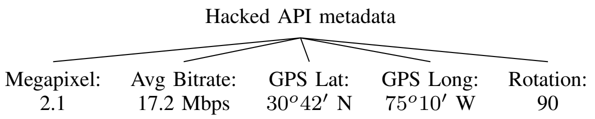


Figure 2. Parler metadata exploited

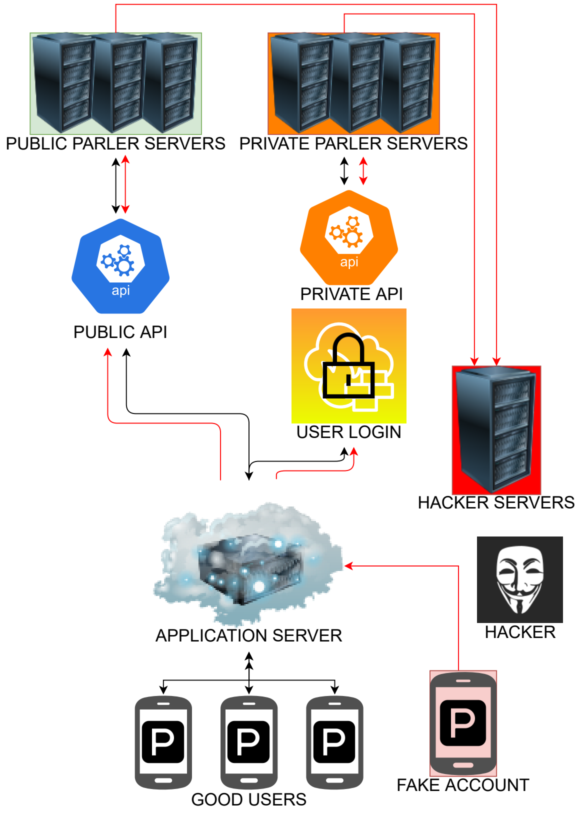


Figure 3. Architecture of Parler Attack

from Parler's servers [7], [8]. As illustrated in Figure 3, the hacktivist is seen using a false Parler account to gain access to Parler's servers using API calls. Once the hacktivist had access to the servers, they were able to download and store data to their server. This action is called massive API scraping. Scraping is the act of using bots to generate many GET requests to the Application Programming Interfaces (APIs) and download as much data from the website/server as possible. Parler's content scraping resulted in more than 70 terabytes of data. 99 percent of posts captured contained metadata. Examples of the type of data can be seen in Fig. **??**. The metadata for a video uploaded to Parler contains: the size of the video, average bit-rate of the video, GPS coordinates where the video was recorded, and the orientation (rotation) of the device during capture have all been recorded and displayed. Now imagine 70 terabytes of data similar to this example, this is what the hacktivist scraped from Parler's servers. The metadata was later examined and led to more than 70,000 GPS locations reconstructed [7], which was a severe compromise in user's privacy. These locations were the exact spot the user was when recording the video or taking the picture, some of which were the user's homes.

The hacktivist scraped the metadata before the social media service went dark scrambling to create infrastructure. The data captured and code snippets gathered were made publicly available on GitHub [9]. Since the D.C. protests, Parler's removal on AWS, and the data breach in January, many people have accessed the repositories and analyzed Parler's code snippets. Many agree and have given their opinion on what mistakes Parler made in misconfiguring their multi-factor authentication. This allowed for the creation of false accounts and unbridled API calls to their servers that caused the data breach. These individuals also discussed how to avoid the same mistakes.

The impact that the data breach had on the private company Parler and its users was severe. Some of the data released in the breach has been used to indite people that took part in the protest. Authorities used the metadata attached to users' posts to tie them to events in and around the country's capitol buildings. Following the impact of the data breach, a discussion has provided insight on how Parler could have avoided these types of attacks. From our research, there are five areas of focus that Parler could have improved to defend against the API scrapping that occurred. They are as follows:

- Secure the mobile application code.
- Implement both authentication and authorization.
- Avoid predictable identifiers and sequential identifiers.
- Use rate limiting to prevent massive API scrapping.
- Protect data used by and for API servers.
- Securely and regularly monitor APIs

In the subsequent sections, the Parler data breach is examined in great detail. In Section II, the background of the attack is given such that one can understand how attackers use API scraping to extort data from web services. Then we discuss the methodology that the hacktivist used in applying the API attack to Parler's servers in Section III. The result of the attack on Parler as a platform and its users is discussed in Section IV. After examining the attack and understanding the methodology the hacktivist used, defense strategies are proposed to defend against future attacks of similar methodology. Several strategies are proposed in Section V. Finally, in Section VI, a discussion of the Parler data breach is concluded with a summarization of points made throughout the paper.

II. Background

Before going deep into the attack methodology, in this section, we describe the Parler application and related background.

A. *Parler application*

The social media service Parler, self-proclaims that they are "the solution to problems that have surfaced in recent years due to changes in Big Tech policy influenced by various special-interest groups... Parler is built upon a foundation of respect for privacy and personal data, free speech, free markets, and ethical, transparent corporate policy" [10]. As a result, Parler has become the center of controversy as the social media platform was cited as facilitating speech that was deemed violence-inciting [3]. This controversy was what Amazon Web Services, Apple App Store, and Google Play App Store used as a reason to remove the social media service's access to their services/stores.

B. Hactivist

The hacktivist, with the Twitter, handle *@donk_enby*, is a female Austrian activist that claimed to have used API scraping to collect terabytes of data from Parler's servers [11]. She claimed to have started the attack to archive all posts and user data related to the events on January 6th. A member of the Chaos Computer Club, CCC, *@donk_enby* offered, "I want this to be a big middle finger to those who say hacking shouldn't be political," [11] as to her motivation behind choosing Parler as a target for the attack. CCC is one of the longest established and most influential civil society organizations dealing with the security and privacy aspects of technology in the German-speaking world [12].

C. API Scrapping

Before Parler's servers were shut down, the hacktivist accessed the company's data held on those servers via API-based attacks [5]. As many organizations are modernizing their applications and virtual spaces, they rely on APIs which allow for communication between services and products. For example, Twitter's APIs allow for users to still interface with Twitter's servers without using a web browser or any mobile application [13], [14]. Developing these APIs can be tricky especially when trying to secure the data that is being accessed. It is important that when building APIs, security is at the forefront of the objectives desired. Without security being closely monitored as the APIs are developed, the APIs that are helping modernize your applications can be the undoing of your company depending on the extent of any data breaches.

API attacks vary widely in severity depending on the extent of the API's reach. In regards to Parler's APIs that were implemented, they failed to authenticate requests properly. From looking at code snippets collected on Github [9], Parler misconfigured their API authentication. This issue made it possible for no authentication to be needed to access Parler's APIs and in numerous cases, this was the biggest issue in letting the hacktivist scrape data off Parler's AWS servers before it was taken offline.

D. Interpreting the results

From the scrapped code that was archived, it can be confirmed that the data breach that occurred with Parler, is based on an API scrapping using false accounts to grant escalated privileges to Parler's servers [6]. The data that the hacktivists were able to obtain was the metadata tags attached to public posts that were otherwise "private", even those that were thought to have been deleted/erased from the user's profile. Many platforms that allow posts from users, automatically remove metadata from content that is uploaded to their site by users. The data available from Parler's servers was a result of Parler failing to manage this metadata, which should have been deleted and erased when users posted. Posts and content deleted by users should have also been purged from Parler's servers. Parler failed to securely protect users' data and allowed the hacktivist to take personal data from their servers. Users of Parler's services had their data released to the public with the data breach that occurred.

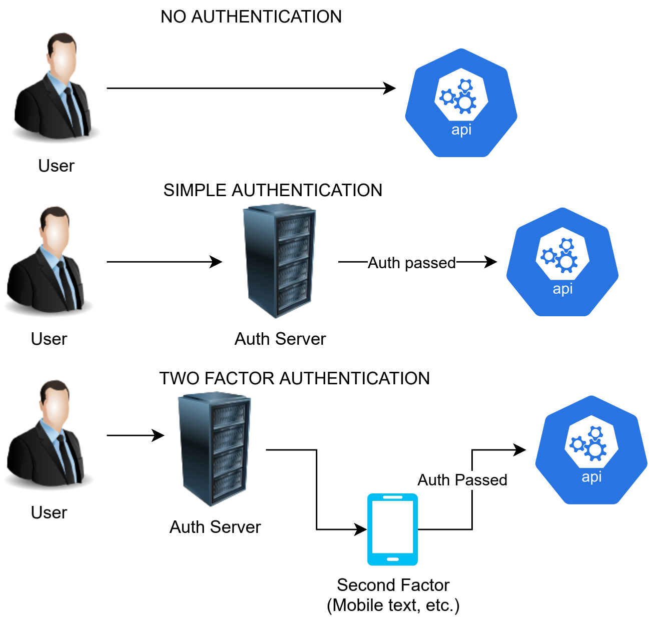


Figure 4. Basic representation of varying levels of API authentication. A multi-factor authentication is preferable for enhanced security.

III. Attack Methodology

Without proper API security implementations, a target becomes much more vulnerable to attack [15], [16]. Once Parler became a target for the hacktivist, the exploitable vulnerabilities within its API were taken advantage of to harvest as much data as accessible before the platform was shut down. The attack performed made use of Parler's lack of authentication and rate-limiting on requests made to its API to gain access to information on their servers that should not have otherwise been accessible to the general public.

A. Reconnaissance/Scanning Phase

When the hacktivist began to target Parler, their first action was most likely to search for vulnerabilities. In the modern era, API security is the logical place to begin looking for this information. API attacks are becoming more and more common, partially due to the lack of awareness of how to properly secure API [17]. As a result, the hacktivist who went after Parler would not have taken very long to uncover the lack of authentication on calls made to the platform's API. Parler failed to implement authentication correctly. This led to no authentication, as seen in Figure 4. Proper authentication would force API calls to send authentication tokens generated by either pre-authorized keys or multi-factor authentication-generated keys that would legitimize the API call.

The vulnerabilities in the API were uncovered by sending requests to Parler's servers and monitoring the responses and traffic created as a result of these requests. Under a properly secured API, seeing a large number of requests from the same source in a short time frame would be an immediate cause for concern that a potential attack is underway. However, Parler's API was not secured or actively monitored to the point where the hacktivists were able to act unopposed in their efforts. Because of this, the hacktivists were able to discover that very little information was protected behind authentication. As soon as the hackers had an entry point, they had access to a vast amount of data [18] [19].

B. Gaining Access

Once the API vulnerabilities were uncovered by the hacktivist, the next stage of attack was to gain access to the data that should have normally been out of their reach. Due to the lack of authentication and security measures to act against large amounts of requests in short time frames, the hacktivists could have used scripts and programs to force their way into uncovering credentials [6]. Without any means of rate-limiting, a computer script can quickly discover a valid password by simply guessing strings of characters until one returns the desired response from the server [20]. However, accessing a large amount of information through this method would normally be somewhat time-consuming. It could be assumed that the hacktivists going after Parler would have to work quickly to get as much data as possible before the platform was shut down, so it was worthwhile for them to find a way to expedite the process.

C. Exfiltration of data

With credentials being uncovered through brute force, the hacktivists were able to exploit another major vulnerability in Parler's data: predictable and sequential identifiers attached to their data. Because of this, it became easy for the hacktivist to make use of scripts to access data. By making use of broken object-level authorization, a common form of API attack which takes advantage of servers that lack authentication and rate-limiting, they were able to uncover large amounts of related data through horizontal privilege escalation after figuring out patterns that were used for data identifiers [21].

The amount of data able to be exposed by the hackers was already a great testament to the weak and exploitable nature of Parler's API. However, it should be noted that the blow could still have been lessened had Parler properly scrubbed their user's posts metadata to get rid of the important information that was stored on their AWS data servers. Metadata is data that contains information beyond just what the physical content of the data entails, such as creation date and even geographical location. Parler did not scrub this data, meaning that once the hackers had access, they had access to the data. Additionally, they also had access to the metadata, information which could be theoretically used to track the exact activities of individual users by determining from when and where they made particular posts. If this information is not scrubbed and disposed of, it can be viewed and analyzed by anybody who gains access to the original data to which it is attached [22] [23].

Table I
IMPACTS FROM THE DATA BREACH

Deplatformed	Parler lost its platform from AWS's servers and mobile app stores
Capitol Hill Protesters	Protesters can be identified and prosecuted per criminal violations
Privacy	Violation of privacy as user data is scrapped
Ethics	Ethical dilemma: Do we condone privacy violation in the name of justice?

D. Results of the attack

Due to all of Parler's vulnerabilities, the hacktivist was able to quickly identify a method of attack, use brute force scripts to gain access to the servers, request and scrape information from many users' data, uncover patterns used for data identification, and even obtain unsecured metadata attached to files, all in a very short time. These attacks hinged primarily on Parler's lack of API authentication and rate-limiting. Without authentication, the hacktivist was able to scrape as much data as accessible off of Parler's servers. 70 terabytes of data were downloaded as a result of this failed implementation. While some data could have still potentially been scraped had Parler used more secure implementations of its API, the severity of the data breach would have been drastically reduced.

IV. Impact of the Data Breach

The Parler Data Breach presents several impacts as a social media and communications platform, considering Parler's role in the events that occurred on January 6th. It is similar to other credential surfing attacks in the past couple of years [24], [25]. It presents legal, ethical, and privacy implications in modern applications. Table I summarizes the impact of the Parler data breach that is discussed in this section.

A. Removing Parler from Major Mobility and Web-hosting Platforms

When Amazon Web Services decided to no longer host Parler on their cloud hosting, Parler could no longer actively run their platform. While losing access to AWS was the greatest issue to arise against Parler, losing access to mobile app stores was equally challenging. Even though Parler was free to use service for users logged in, Parler was able to monetize advertisements that other businesses could buy on their service. Without access to mobile app stores, Parler would no longer be able to get new users to create more revenue for the company, as these mobile stores are the main distribution for all popular mobile devices. Users were no longer able to download the application to their device, update the application from the store, or reinstall the application if previously purchased (free to use) from their respective app store.

B. Capitol Hill Riot

Since Parler was used as a platform to spread information about the Capitol Hill protest before it happened, Parler's data can help identify the involved parties. If data is handed over to law enforcement agencies, the gathered data may be used to identify the individuals who made entry into the U.S. Capitol building and other alleged criminal violations. Additionally, the data could be presented as evidence against offenders in the Capitol Hill protest investigation.

Despite the advantage of identifying the criminal violators, Parler data breach presents an ethical dilemma. Parler has multiple security vulnerabilities that allowed digital activists and data archivists to obtain their data. As presented in the Section II background of the Parler data breach, Parler did not require authentication on its public API, purge user data or scrub content metadata. It also used predictable, sequential identifiers for content and implemented multi-factor authentication incorrectly. Because Parler presents itself as a transparent social media app, in contrast to other social media platforms such as Twitter and Facebook, it supports free speech and does not censor all but the extremely explicit content on its platform. Users who flocked to Parler presumably believed that their data will be secured and maintained confidential, however, the data breach has violated the confidentiality and privacy of the Parler users' data.

C. Privacy

Data that was thought to be secure and confidential by Parler users can be made available to the public by digital activists and data archivists, resulting in privacy violations. Parler users who were involved in the protest can be potentially identified and held responsible for the committed acts at Capitol Hill on January 6th. As digital activists and data archivists collected data from Parler before its shutdown, the data breach presents serious privacy implications to Parler users [5].

D. Ethics

Although the collected data is critical evidence for law enforcement within the country, the Parler data breach brings in serious concerns regarding the security of user data of social media applications. The data breach resulted in the siphoning of 70TB of user data from the site before AWS shutdown Parler services [8]. One terabyte of data can store 250 movies, or 250,000 photos, or 6.5 million document pages equating to about 1,300 filing cabinets [26]. 70 terabytes would add up to 455 million document pages bringing the number of filing cabinets up to 91,000. The sheer amount of leaked data represents the growing concern for user privacy in our increasingly online world.

In this particular data breach, the information is supposedly used to do good in identifying the individuals who committed acts of criminal violations. In the eyes of those individuals, the data breach hurts them. Ethically, we can argue that the means justify the end as those people have chosen to commit criminal violations. It is still unclear the charges brought against these people and whether or not any information was stolen from Parler's servers will be used against the persons held responsible for the incident on January 6th.

Generally speaking, data breaches present a grave privacy problem to normal citizens. Our data can be used to locate us, possibly causing us physical harm if the malicious actor who somehow obtained our data chooses to do so. Our data can also be used in more legal ways like social media marketing to influence our buying decisions. User data is a gold mine of the internet and breaches like this one adds to the increasing privacy concern.

V. Possible Defense Solutions

There are multiple guidelines for collecting, storing and using personal information. The most common regulations are:

- Children's Online Privacy Protection Act (COPPA) - regulates the data collection policy about minors.
- Health Insurance Portability and Accounting Act (HIPAA) - regulates data collection policy for healthcare users.
- Gramm Leach Bliley Act (GLBA) - regulates data handling policy for financial institutes.
- Fair Credit Reporting Act (FCRA) - regulates the collection and use of credit information.

While Parler has resurfaced through a foreign hosting service [27], no publicly available information about the security changes can be found. Thus, we have researched possible solutions that can be done to prevent a similar data breach. In the following subsections, security implementations are suggested concerning the vulnerabilities that allowed the data breach to happen, including:

1) Missing authentication on public APIs
2) Use of sequential identifiers
3) Lack of restrictions and monitoring on API calls
4) Excessive data exposure in the form of content metadata

A. Secure mobile application code

Mobile applications are essentials for customer-side interactions for most businesses today. As a result, mobile applications have entry points to APIs to obtain data from servers. Entry points like these should be secured properly to prevent exploitation. Client-side code should be protected and data should be kept server-side if the client-side code or generated data is compromised. Particularly, secure coding practices and secure design choices should be made. [6]

B. Implement authentication and authorization

Through application modernization and mobilization, internal APIs are unintentionally becoming external-facing through mistakes in authentication and authorization. As illustrated in Fig. 5, internal APIs are entry points into an application's server-side data and can be exploited if they are vulnerable to broken object-level authorization [28]. Thus, it is recommended that internal and private APIs be monitored for

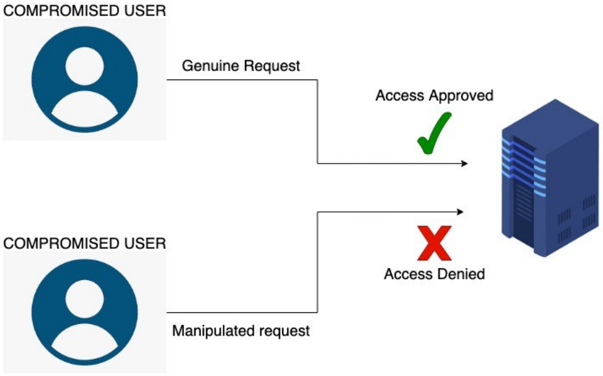


Figure 5. Server side authentication and authorization of user requests to prevent broken object level authorization.

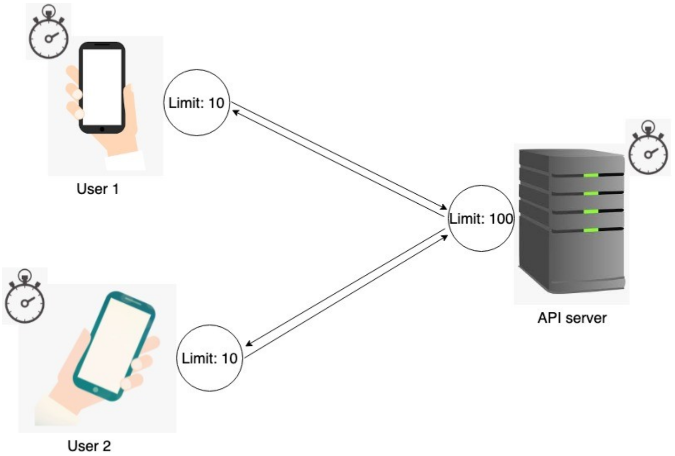


Figure 6. Implementation of API rate limiting where users are limited to a certain number of API calls and the API server is limited to a higher number of API calls.

continuous authentication and authorization. There should be an implementation of a proper authorization mechanism that relies on user hierarchy. This mechanism should check whether the user has permission to access the data that they are trying to access, as shown in Figure 6. Use well-known and established encryption algorithms and implementations to protect the data.

C. Avoid sequential and predictable identifiers

Using sequential identifiers makes data vulnerable to enumeration and brute force attacks [6]. Randomness reduces the risk of data exploits to enumeration and brute force attacks as it is less likely to find identifiers that are in use. To reduce the risk of this vulnerability, ensure that the algorithms are using sufficient randomness for sensitive data. This includes identifiers for records, authentication tokens, and even encryption purposes. Use well-vetted frameworks, SDKs, and libraries with pseudo-random number generators (PRNGs) that are sufficiently random.

D. Use rate-limiting as a starting point to mitigate API abusers

Parler data breach occurred partly due to the lack of rate limiting for API calls, leading to the possibility of such a large-scale data scraping. To reduce the risk of API abuse, rate limiting can be implemented as a starting point as illustrated in Figure 6 [20]. There are alternative solutions to data breaches that are better as rate-limiting can cause issues with app functionality, which becomes even more complex when millions of users are added and requests to the equation. It is advised to seek solutions with more advanced traffic collection and behavioral analysis [6] as such implementation can track and analyze requests and detect API abusers.

E. Protect data and APIs that serve data

User privacy is an important issue to address when collecting and saving user data. There are numerous guidelines governing user and data privacy according to different regions in the world. For better security, it is recommended to implement security at all stages of data transport and delivery, such as TL 1.2 for transport security control and a minimum of 128 bits key encryption for sensitive data. Privacy regulations are often not prescriptive so it is up to the security team to decide on the security implementations based on the type of data that is being transmitted.

F. Monitor APIs

When the mass scraping of Parler data occurred, the sheer size of the data being requested through API calls should have alerted Parler to a data breach. Parler could have shut down its services to stop more data from being collected through the exploitation of its API, but it did not do so. This suggests a lack of security monitoring of its APIs. It is advisable to have security monitoring implementations in place to monitor API calls and alert on suspicious or unusual requests in the event of an attack to shut down services. This can reduce the impact on user and data privacy.

G. Review API responses

APIs are designed to return sensitive data to users. Due to this design, API requests and returns are vulnerable to excessive data exposure. To handle this vulnerability, applications should never rely on the client-side to filter sensitive data [29]. Instead, review the API responses to ensure that they only contain legitimate data. Specifically choose properties that you want to return instead of using generic to string() methods. Consider implementing a schema-based response validation mechanism [29] which defines and enforces data in API responses.

VI. Conclusion

Parlar data breach is one of the most controversial and politically influential data breaches of this decade. Parler was not prepared for the influx of new users between 2020 and 2021. Trying to scale their system to handle the new users left them vulnerable to attacks unforeseen. Parler's failure to implement proper authentication and authorization allowed for access to their servers by the hacktivist. The hacktivist created false accounts to use both the public and private APIs accessing both sides of Parler's servers pulling as much data as they could from the servers. Parler's inability to protect users' private data created a severe data breach that has had many implications with people associated with the events of January 6$^{\text{th}}$. After examining the methodology of the attacks and implications on the company and users, potential solutions were proposed involving correctly implemented authorization, rate-limiting of API requests, and API request sanitation. These three topics were the main points that Parler implemented when they came back online once they created their servers to run their application on.

References

[1] G. Clary, "Parler Wasn't Hacked, and Scraping Is Not a Crime." https://www.lawfareblog.com/parler-wasnt-hacked-and-scraping-not-crime.

[2] E. Caroscio, J. Paul, J. Murray, and S. Bhunia, "Analyzing the ransomware attack on dc metropolitan police department by babuk," in *IEEE International Systems Conference (SysCon)*, 2022.

[3] C. Pedroja, "Rep. Carolyn Maloney Says Parler Sent Warnings Over 50 Times Ahead of Capitol Riots." https://www.msn.com/en-us/news/politics/rep-carolyn-maloney-says-parler-sent-warnings-over-50-times-ahead-of-capitol-riots/ar-AAL5bwq, June 2021.

[4] T. H. Neale, Government, and F. Division, "The electoral college: how it works in contemporary presidential elections," in *The electoral college: how it Works in contemporary presidential elections*, Congressional Research Service, Library of Congress, 1999.

[5] . . p. U. Dan Goodin Jan 12, "Parler's amateur coding could come back to haunt Capitol Hill rioters." https://arstechnica.com/information-technology/2021/01/parlers-amateur-coding-could-come-back-to-haunt-capitol-hill-rioters/, Jan 2021.

[6] M. Isbitski, "Analysis of the Parler Data Breach." https://salt.security/blog/unpacking-the-parler-data-breach, Jan 2021.

[7] Z. Whittaker, "Scraped parler data is a metadata gold mine." https://techcrunch.com/2021/01/11/scraped-parler-data-is-a-metadata-goldmine/, Jan 2021.

[8] V. Petkauskas, "70TB of Parler users' messages, videos, and posts leaked by security researchers." https://cybernews.com/news/70tb-of-parler-users-messages-videos-and-posts-leaked-by-security-researchers/, Jan 2021.

[9] "Parler-Grab Github Repository." https://github.com/ArchiveTeam/parler-grab, Jan 2021.

[10] Parler, "About Parler." https://parler.com/about.php.

[11] S. Dixit, "Who is Parler Hacker @donk_enby? deleted posts provide 'very incriminating' evidence against US Capitol Riots." https://meaww.com/who-is-parler-hacker-donkenby-deleted-posts-very-incriminating-data-dump-us-capitol-riots-trump, Jan 2021.

[12] "Chaos computer club." https://www.ccc.de/en/club.

[13] Twitter, "About Twitter's API." https://help.twitter.com/en/rules-and-policies/twitter-api.

[14] B. Gibson, D. Lewis, S. Townes, and S. Bhunia, "Vulnerability in Massive API Scraping: 2021 LinkedIn Data Breach," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021.

[15] L. Sterle and S. Bhunia, "On SolarWinds Orion Platform Security Breach," in *2021 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, 2021.

[16] J. Huddleston, P. Ji, S. Bhunia, and C. Joel, "How VMware Exploits Contributed to SolarWinds Supply-chain Attack," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021.

[17] B. Doerrfeld, "For Hackers, APIs are Low-Hanging Fruit." https://securityboulevard.com/2021/07/for-hackers-apis-are-low-hanging-fruit/.

[18] Owasp, "API-Security/0xa2-broken-user-authentication.md at master · OWASP/API-Security." https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa2-broken-user-authentication.md, Mar 2020.

[19] L. Tang, L. Ouyang, and W.-T. Tsai, "Multi-factor web api security for securing mobile cloud," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2163–2168, 2015.

[20] Owasp, "API-Security/0xa4-lack-of-resources-and-rate-limiting.md at master · OWASP/API-Security." https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa4-lack-of-resources-and-rate-limiting.md, Dec 2019.

[21] Owasp, "API-Security/0xa3-excessive-data-exposure.md at master · OWASP/API-Security." https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa3-excessive-data-exposure.md, Dec 2019.

[22] M. Breen, "Nothing to Hide: Why Metadata Should Be Presumed Relevant," *Kansas Law Review*, 2008.

[23] E. N. Litzinger, "The ethical dilemma of scrubbing metadata: The pathway to a better approach," *North Kentucky Law Review*, vol. 36, no. 4, 2009.

[24] M. H. Nguyen Ba, J. Bennett, M. Gallagher, and S. Bhunia, "A Case Study of Credential Stuffing Attack: Canva Data Breach," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021.

[25] J. Rudie, Z. Katz, S. Kuhbander, and S. Bhunia, "Technical analysis of the nso group's pegasus spyware," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021.

[26] Dropbox, "How much is 1 TB of storage?." https://www.dropbox.com/features/cloud-storage/how-much-is-1tb.

[27] N. Lanxon and A. Thomson, "Parler reappears with help from Russian-owned security service." https://www.mercurynews.com/2021/01/19/parler-reappears-with-help-from-russian-owned-security-service/.

[28] P. Silva, "API1:2019 Broken Object Level Authorization." https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa1-broken-object-level-authorization.md#api12019-broken-object-level-authorization.

[29] P. Silva and I. Shkedy, "API3:2019 Excessive Data Exposure." https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa3-excessive-data-exposure.md.