Hacking misconfigured Cloudflare R2 buckets: A complete guide

BY BLACKBIRD-EU · SEPTEMBER 12, 2024 · LAST UPDATED ON APRIL 5, 2025

Cloudflare R2 buckets are recently becoming more popular as an alternative to AWS S3 buckets for their simplicity, integration support and zero-egress fees. Customers who opt-in to use Cloudflare R2 are not going to be charged for any traffic to and from the bucket. This often means a severely reduced cloud bill for software companies that heavily rely on cloud storage buckets (like AWS S3, Azure Cloud Storage, Google Cloud Storage bucket, ...).

Cloud storage buckets like R2 are easy to use, integrate, and manage and are usually used to store large files and unstructured data objects, including ones with sensitive data (such as invoices and receipts). However, just like with AWS S3, if left incorrectly configured, it can introduce a new attack surface for bad actors.

In this article, we will cover one of the most common security misconfigurations in Cloudflare R2 buckets that developers often make.

What is Cloudflare (CF) R2?

Cloudflare R2 storage is a high-performance storage service that allows developers to store unstructured data objects (such as images, videos, PDF files, and even static HTML and JavaScript files). Furthermore, it is also easy to integrate and the storage bucket can also be used as a public bucket or CDN.

One thing to note is that Cloudflare R2 is still in the early-beta phase, there are several features and functionalities that AWS S3 does support but is lacking in CF R2. One of which is Access Control Lists (or ACLs) and Bucket Policies.

AWS S3 allows developers to refine permissions up to the Data Object level. Cloudflare, on the other hand, allows developers to only manage access to global scopes with API tokens. That's why R2 buckets are less prone to policy misconfigurations. However, R2 comes with R2.dev, a simple feature specifically for testing purposes only that enables developers to make their buckets publicly available.

However, if you are interested in learning more about AWS S3 security misconfigurations, read our latest detailed article that covers the most common security misconfigurations present in S3.

Finding & identifying CF R2 buckets

There are various ways to find or identify a Cloudflare R2 bucket. We will be covering 2 ways in this article.

Examining HTTP responses:

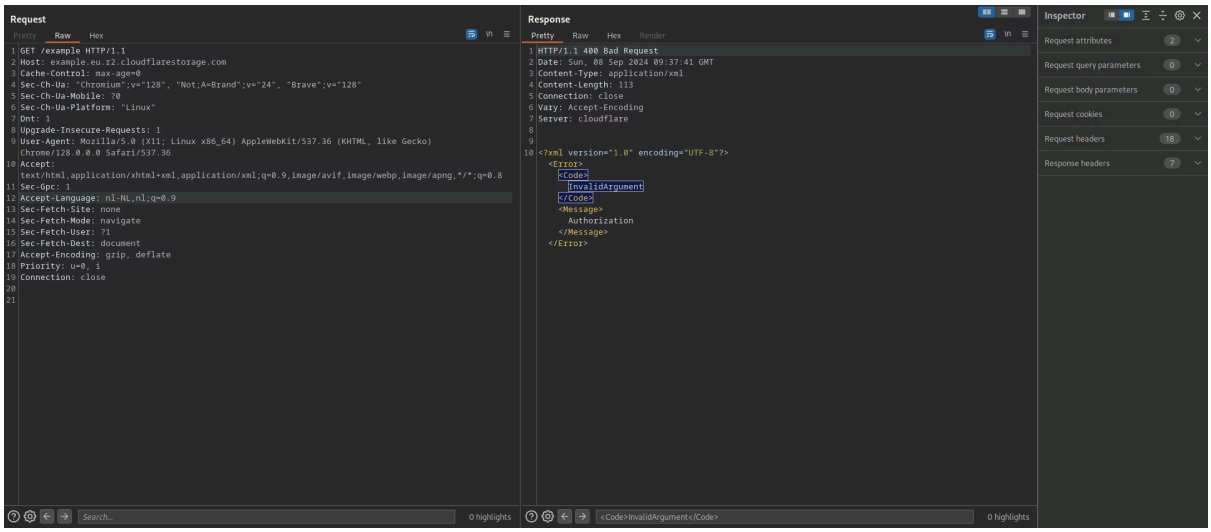
If you're frequently testing web applications, you are likely using a proxy intercepting tool to help intercept HTTP requests for example. You can search for references in HTTP responses as R2 buckets are sometimes used for serving static content such as images and javascript files.

You can search for an explicit reference to a bucket in the response using the following regex pattern:

```
\.r2\.cloudflarestorage\.com\/?
```

Or search for a response of a typical private R2 bucket:

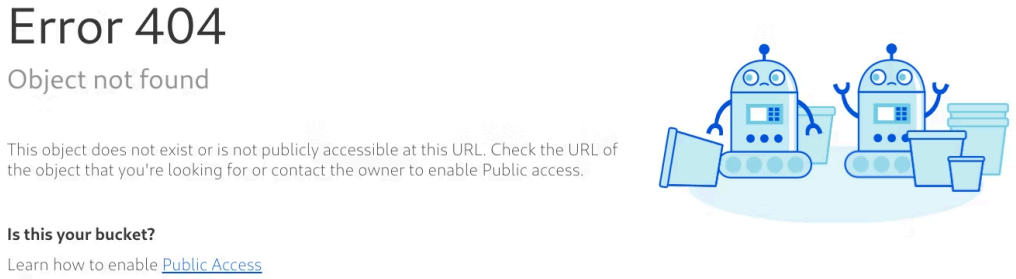
```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>InvalidArgument</Code>
  <Message>Authorization</Message>
</Error>
```



Cloudflare R2 private bucket enumeration

For public storage buckets, you can look for the following response:

```
Object not found
```



Cloudflare R2 public bucket enumeration

Dorking:

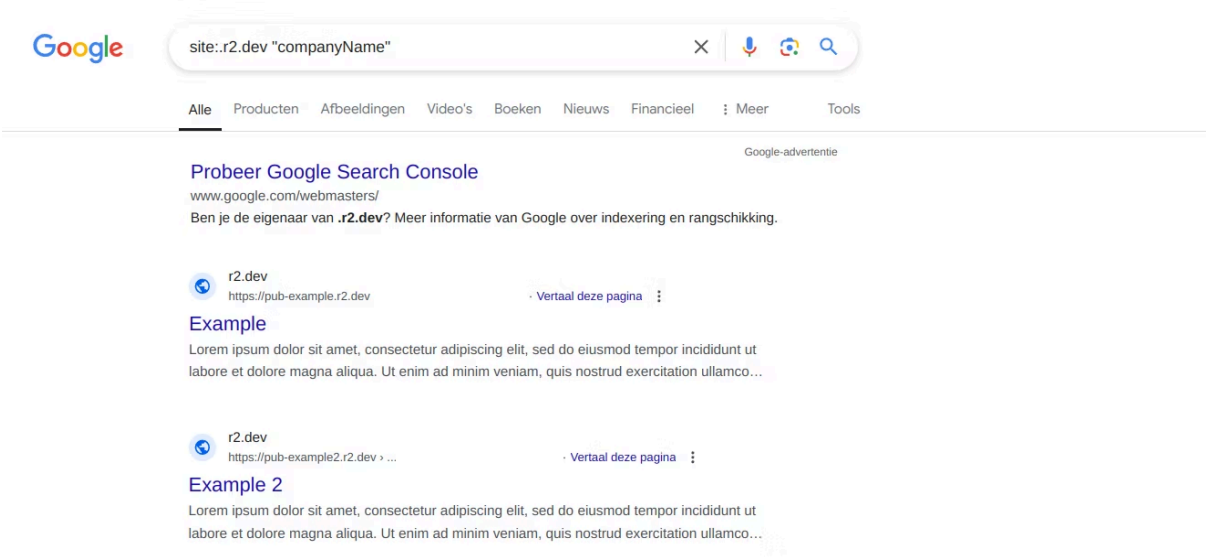
Several popular search engines like Google, Bing, DuckDuckGo and Brave Search support search syntaxis. You can take advantage of this by specifically looking for your target and browsing through any previously indexed results.

Search for private CF R2 buckets:

```
site:.r2.cloudflarestorage.com "company"
```

Search for public CF R2 buckets (with R2.dev enabled):

```
site:.r2.dev "company"
```



Finding CF R2 buckets using search engines

After listing all R2 buckets associated with our target, we can now proceed to the exploitation phase.

R2.dev access enabled

By default, Cloudflare R2 buckets are never publicly accessible and will always require explicit user permission to enable public access. However, developers can opt-in to make their bucket publicly available by assigning a custom domain or by enabling R2.dev.

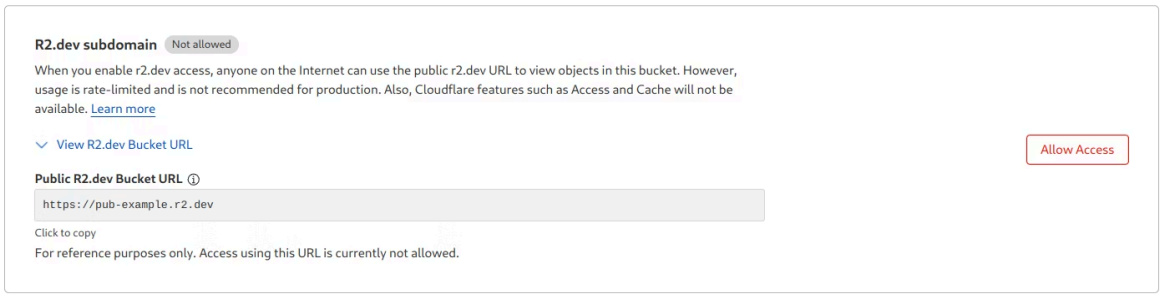
Assigning a custom domain is the default and recommended way to make your bucket public.

R2.dev is a simple feature that enables developers to make their buckets publicly available. It is only meant to be used for testing purposes (especially during active product development). But sometimes, developers forget to turn it off and risk disclosing data objects to unauthenticated visitors.

This means that if the developer did not assign any custom domain to the storage bucket, we could still request private data objects via R2.dev!

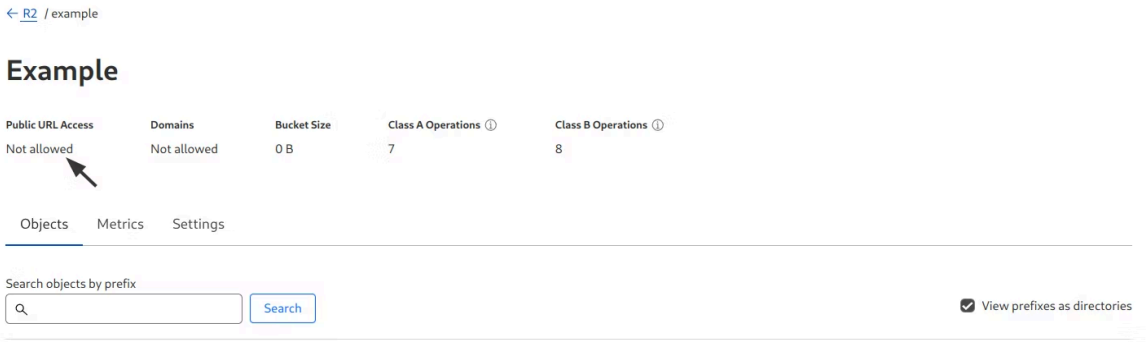
Data objects (or files) can often include sensitive information (such as invoices and receipts) and should not be accessible to anyone.

Attackers can, with Google dorking for example, easily look for sensitive files that have been indexed in the past.



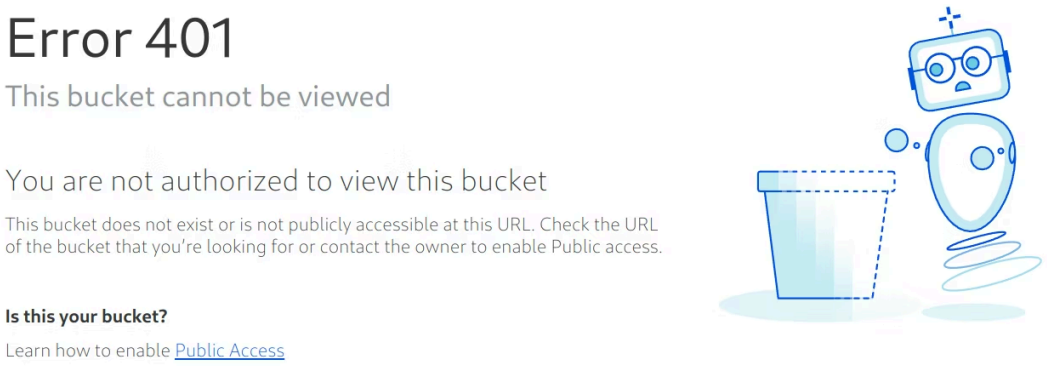
Make sure to disable R2.dev access on your R2 storage bucket

When disabling R2.dev access, make sure to also remove any connected domains. You can verify that public access is disabled by looking at the "Public URL Access" badge located at the top of your page.



Verify that R2 public access is disabled

TIP! You can quickly check as an unauthorized user if an R2 storage bucket has R2.dev enabled by observing the response. If the index page returns "Error 404. Object not found", R2.dev is enabled. If it returns "Error 401. You are not authorized to view this bucket", R2.dev is disabled.



Cloudflare R2 bucket with R2.dev disabled

Missing authorization checks for sensitive files

Since Cloudflare doesn't provide Access Control Lists or Bucket Policies (or similar functionality to control access), developers are responsible for using a middleware that performs basic authorization checks. Especially for sensitive files containing PII data to verify the ownership of each individual file or data object.

Missing middleware for authorization checks

A middleware is an API that should stand between the R2 bucket API and the client. The middleware should be responsible for verifying and authorizing access to any incoming client request. But also prevent data object overwrites if a conflicting filename is supplied in a new client request.

If this middleware is missing, and the bucket is public, then R2 has no way of verifying access to any requested file. Any bad actor could view sensitive files without being authorized to do so.

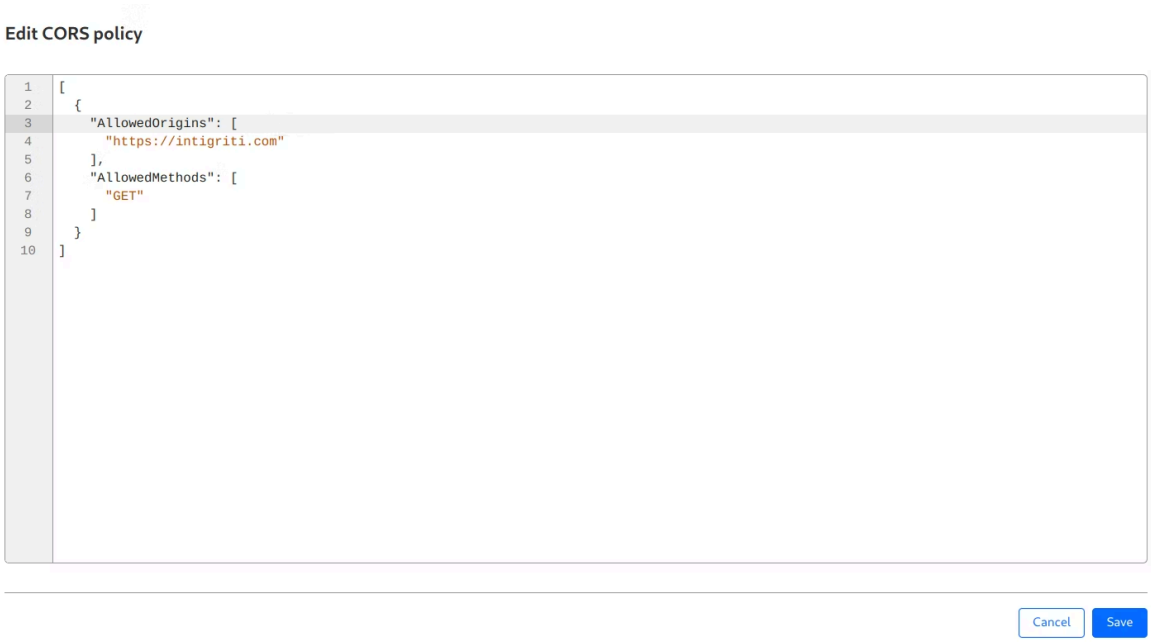
If no measures are taken to restrict file types, file size or file name, a bad actor will be able to upload unwanted files (such as SVG files that can introduce stored XSS vulnerabilities), upload huge data chunks and increase the cloud bill or even overwrite files by supplying an existing filename!

Always test your targets for all the aforementioned security misconfigurations to make sure that your target handles client-supplied files safely.

Misconfigured CORS Policy

Since R2 can also be used as a public storage bucket and host SPAs (single-page applications), Cloudflare allows developers to declare a CORS (Cross-Origin Resource Sharing) policy to whitelist certain origins.

An overly permissive CORS policy can allow bad actors to access the bucket from unauthorized origins.



CORS Policy editor for Cloudflare R2

TIP! You should always make sure to test for common third-party domains (such as code sandboxes) that have been whitelisted for testing purposes.

Conclusion

Few bug bounty hunters are aware of security misconfigurations in Cloudflare R2! We hope that this article has helped you learn something new today!

So, you've just learned something new about Cloudflare R2 security misconfigurations... Right now, it's time to put your skills to the test! You can start by practicing on vulnerable labs or... browse through our 70+ public bug bounty programs on Intigriti and who knows, maybe earn a bounty on your next submission!

START HACKING ON INTIGRITI TODAY