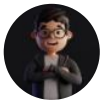


Farhad Karimi (n0lsec) Follow
Feb 11 · 4 min read · Listen

CVE-2022–24992: QRCDR ZeroDay Path Traversal Vulnerability

This post is about CVE-2022–24992 which refers to vulnerability in QRCDR widely used QR-Code generator script.

About QRCDR:

QRCDR is a popular PHP — JavaScript QR-Code Generator, which is widely used for creating customized QR-Code in easy steps.
also, it's used by a few WordPress QR Code Generator Plugins and Mobile applications Which is not covered in this article.

Hello dear readers, it's @n0lsec, Today I going to share with you details of the zero-day vulnerability which I was found in QRCDR (reported to the vendor and patched).

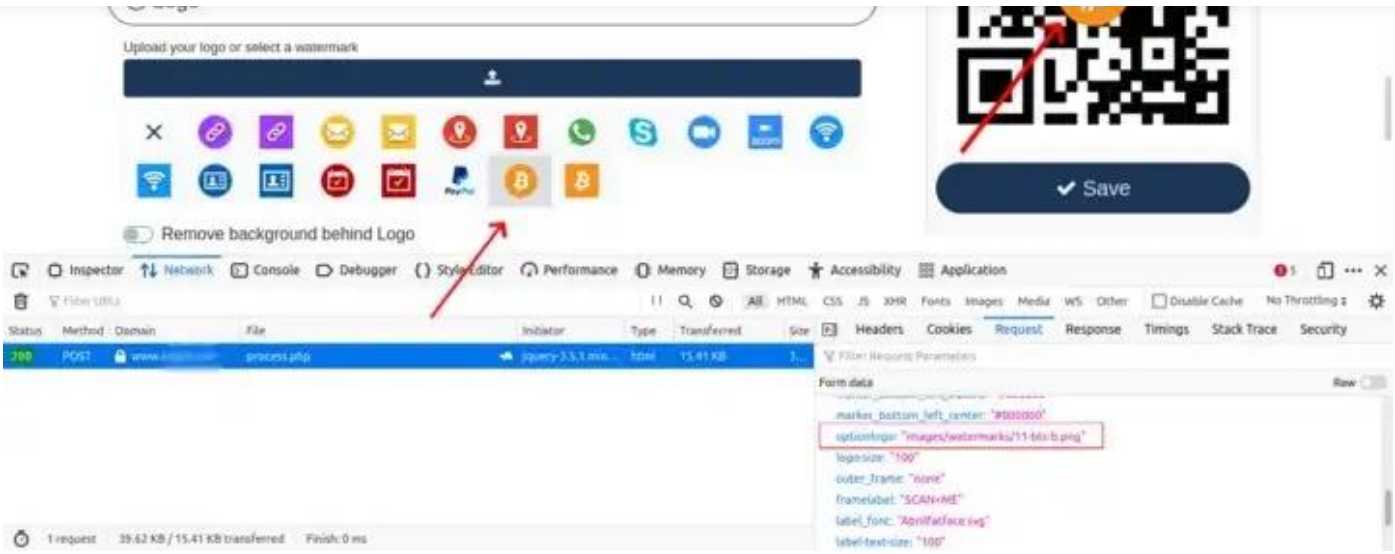
Core finding

- QRCDR(5.2.7 and all prior versions are vulnerable) to Directory Path Traversal Vulnerability.
- POST parameters with `optionlogo=[payload]` which is a malicious payload sent to lead to path traversal
- According to server security configurations, the attacker can read arbitrary sensitive server files, configurations, etc
- An attacker can escalate path-traversal to `RCE` in some cases

Mitigation

- Just update to the latest version, At the time I wrote this post it is Version 5.2.9

How find?
Meanwhile working on a specific Bug Bounty program, which is called REDACTED.COM because of the information disclosure policy, I found quite an interesting endpoint that leads customers to create customized QR-Code with extra options, I opened browser inspector and watched to Request/Responses, so I realized part of script leads users to add a custom logo to our QR-Code which was so interesting to me,
here I opened the browser inspector and look at what we had, there was the main file of the script `process.php` which is in the path: `/ajax/process.php` intercept POST request and body parameters, `optionlogo` looks like below:

the script provided some predefined logos and we can select one of them and load it into our QR-Code, The key part of vulnerability is here which maybe let us find some interesting things like Path Traversal — LFI or maybe SSRF so I decided to check.

After some work I tried Path-Traversal, I tried the first payload without encoding or any bypass method, and I received `/etc/passwd` file content with base64-encoded. So that's it.

REDACTED.COM's Base64-Decoded contents of `/etc/passwd`

Report to Dev-Team

After checking the code, found out that the script is part of the QRCDR script, so I found the Dev-Team email and sent the details of a vulnerability, they responded quickly and fixed the vulnerability after a few hours. Considering that the script is popular and widely used, I decided to write this post.

the code and found that the script does not completely clear the `POST` value of the `optionlogo` parameter and uses the code directly in the QR-Code.

part of the process.php file, look at line 17 which is no kind of sanitization

`<image xlink:href="/etc/passwd(base64-encode)" >` which brings the content of any files to us! :)

Mitigation

The vulnerability is now fixed but if you want to fix it manually should know to remove any malicious content which can help an attacker to change the directory in file name and directory name.

- removing any special characters in file | directory name `[. / \ ]`
- thinking about cross-platform directory traversal characters
- after all maybe configuring the server with `chroot-jail`, `cloudflare` like functionality to prevent escalation may be a good choice. even I think hackers always find their ways:)

Feel free to contact me on Twitter, I would appreciate it if you send me any problem, feedback or opinion.