CHAMELEON

FOR OUR SMARTER WORLD

MOP Hidden Files and Additional Vulnerabilities

Adam Sarin
217342706

Contents

Contents

Introduction:

The purpose of this report is to potentially identify files that are hidden on the MOP web-server that we might take advantage of in some shape or way, this will be achieved by performing spider attacks using burp suite and ZAP using methods previously used on the Chameleon Site last trimester.

Tools used:

- Kali Linux
- OWASP ZAP (V 2.15.0)
- Burp-Suite (V 1.7.36)

Scope of Testing

The scope of our testing will be isolated to the MOP site, with all methods being undertaken as a user with no permissions nor knowledge of the inner workings of the site and its security.

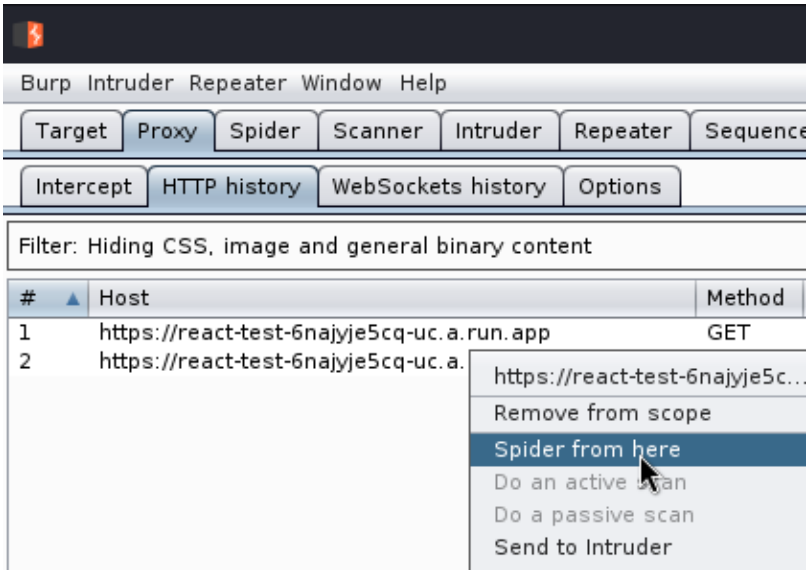
Site: https://react-test-6najyje5cq-uc.a.run.app/

Methodology

Testing was performed with OWASP's "ZAP" which is an open-source penetration tool, designed specifically for web applications, both it and the other tool we will be using; "Burp-Suite" act as a man-in-the-middle proxy, intercepting the traffic between the users browser and the website being accessed, allowing users to even modify the contents if needed.

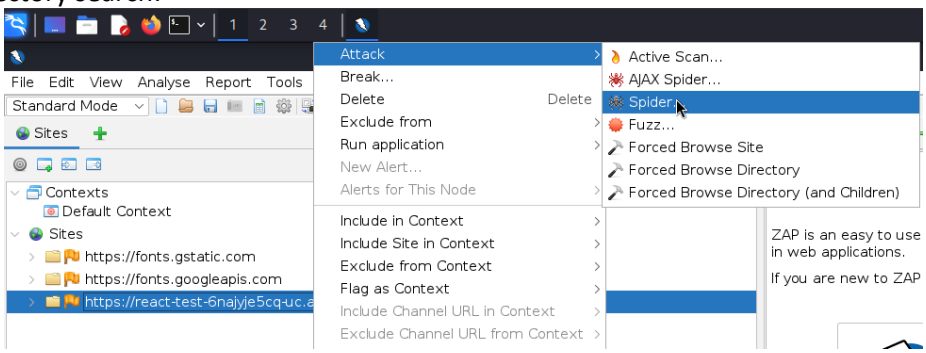
The testing was done with the latest ZAP, but the older 2018 community version of Burp-Suite due to spidering tools being moved from the free version of the software into the professional paid version of the scan, all past versions of burp-suite are available for free on their website.

Running a spider scan was quite simple, requiring the website we are working with to be either added to the scope within burp-suite or accessing the site via our browser using the intercepting proxy to pick it up within the application.

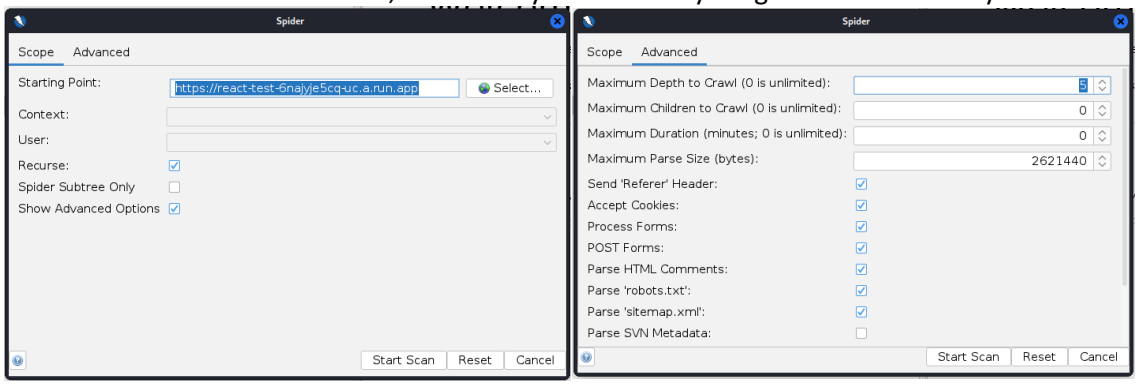


Spidering, otherwise known as "web crawling ", allows burp-suite to trawl the website that's in our scope and attempt to catalogue all resources that can be found, which is something most search engines use.
However, with burp-suite and zap we are using a more invasive search that ignores normal spider limitations and finds more pages and resources then what is listed on the robots.txt file.

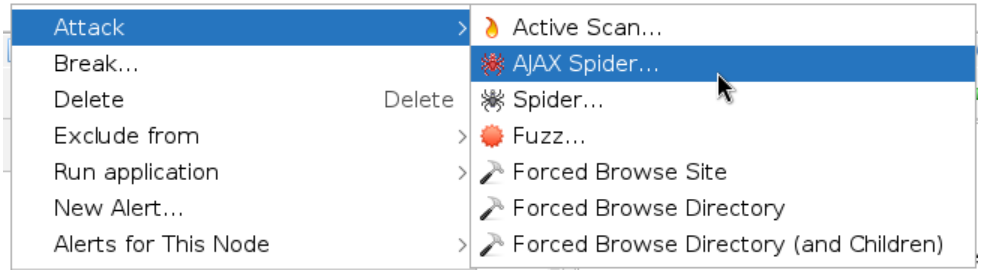
We will also be taking advantage of ZAP's spider scan too as well as its AJAX spider scan and Forced Browse Directory search:



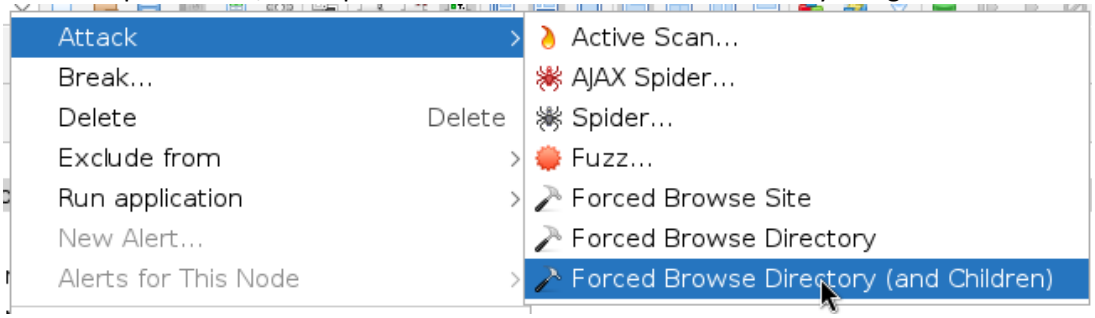
Initiation of the scan is quite simple and just requires the site we are testing to be in ZAP, which is easily done using the Firefox browser built into ZAP and then is as simple as opening the attack menu and specifying the spider scan which we will also be checking the advanced settings to maximise the effects of the scan, however by default everything we need is already enabled.



Additionally due to limitations of the normal spider search due to the website design we will also be running an AJAX spider scan.

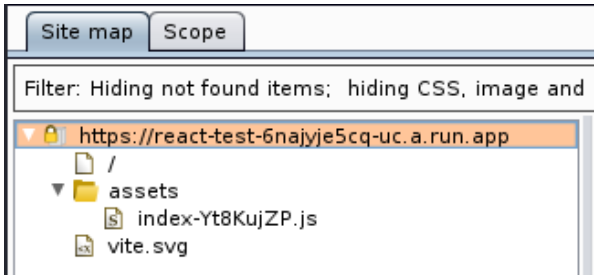


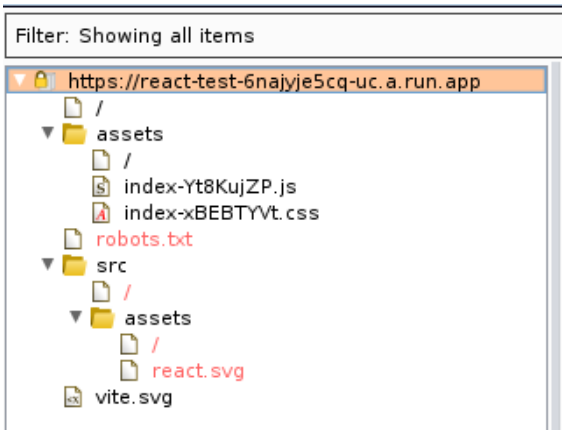
And lastly for good measure we will run another ZAP feature “Forced Browse Directory (and Children)” this scan does exist in Burp-Suite’s collection under “Content Discovery” however it is a pro-exclusive tool so therefore we shall use the ZAP alternative, which instead of attempting to crawl like the spider scans, attempts to access files and directories directly using an inbuilt list in ZAP.

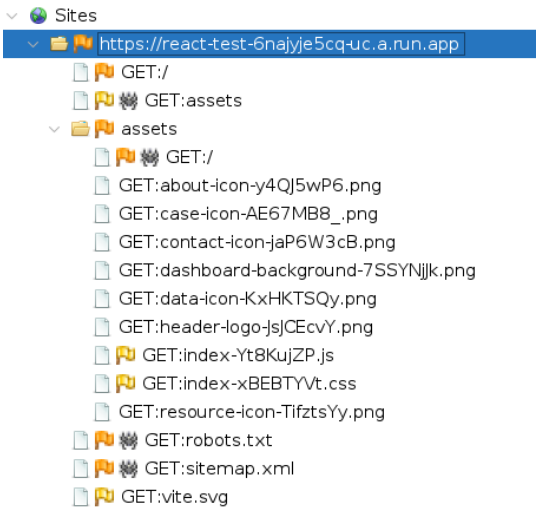


Results

The results of the test run by Burp Suite and its spider scan reveal very little, revealing only an image resource and the JavaScript for the site, and when showing all items only the robots.txt file, as well as some additional CSS is shown.





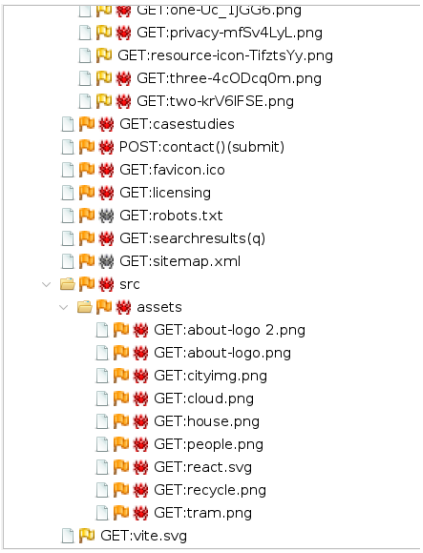


However, unlike Burp-Suite ZAP reveals more than the previous scan, unfortunately for us everything it finds are just general image assets, but still a bigger haul then the Burp’s spider scan.

But using the AJAX scan which employs a crawler that is made specifically for AJAX rich sites “CrawlJax” we can identify more results.

Unfortunately, though while the AJAX crawl identified more files then the rest, it was mostly just asset files such as png and svg files, and nothing new was really identified.

Also using the Forced Browse Directory attack in hand with the AJAX crawl to potentially identify any files or further vulnerabilities we can take advantage helps, but in this case no additional resources were detected in the scan.



Recommendations & Conclusion

In regards to my recommendations, when it comes to the MOP site there were some alerts that were picked up by ZAP that should be addressed, as they may cause potential vulnerabilities, but in relation to the point of the report no additional files or directories seem to allow for us to gain access to things we are not supposed to, and therefore it is my recommendation to the site owner to visit the ZAP site in reference to these alerts that were detected and fix the code accordingly.

References:

What is a web crawler? | how web spiders work | cloudflare. Available at: https://www.cloudflare.com/en-au/learning/bots/what-is-a-web-crawler/ (Accessed: 1 May 2024).

Automated content discovery with BURP suite - PortSwigger. Available at: https://portswigger.net/burp/documentation/desktop/testing-workflow/mapping/hidden-content/automated-discovery (Accessed: 1 May 2024).

Discovering hidden content with burp suite - PortSwigger. Available at: https://portswigger.net/burp/documentation/desktop/testing-workflow/mapping/hidden-content (Accessed: 1 May 2024).

Professional / community 1.7.36 (2018) *Burp Suite Release Notes*. Available at: https://portswigger.net/burp/releases/professional-community-1-7-36 (Accessed: 1 May 2024).

Forced-Browse ZAP. Available at: https://www.zaproxy.org/docs/desktop/addons/forced-browse/ (Accessed: 1 May 2024).

Spider ZAP. Available at: https://www.zaproxy.org/docs/desktop/start/features/spider/ (Accessed: 1 May 2024).

AJAX-Spider ZAP. Available at: https://www.zaproxy.org/docs/desktop/addons/ajax-spider/ (Accessed: 1 May 2024).