Yuvin Perera

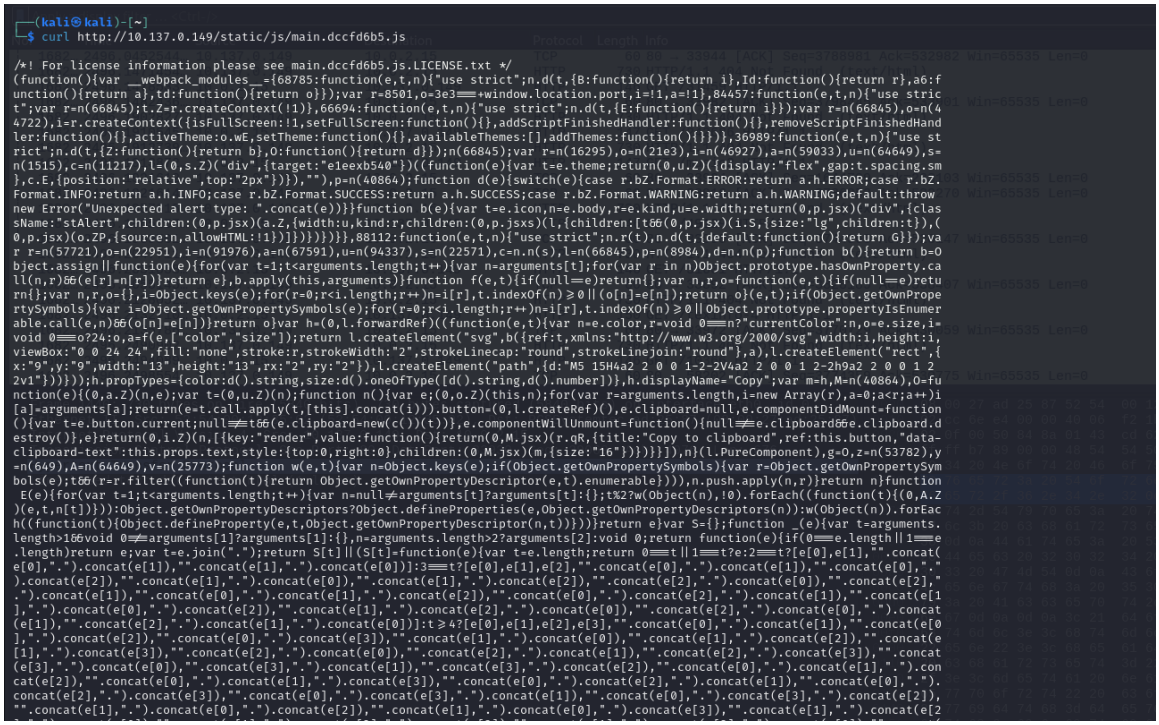
Post-Exploration Report on Target 10.137.0.149

The exploration of the target system at IP 10.137.0.149, running an HTTP service on port 80, was conducted using tools like cURL, Metasploit, Nikto, and Searchsploit. The goal was to identify vulnerabilities, exploitable paths, or misconfigurations. Despite rigorous efforts, no successful exploitation was achieved.

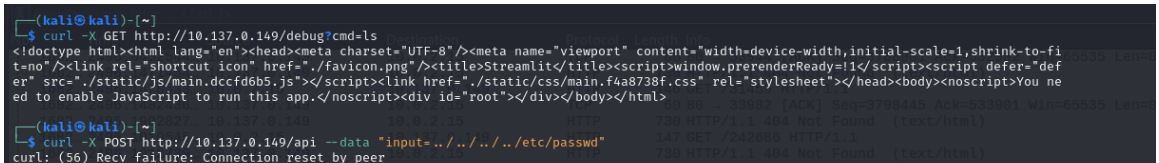
Enumeration and Initial Findings ~

The process began with cURL to test accessible endpoints such as /static, /debug, and /api. The /static endpoint returned a 301 Moved Permanently status, redirecting requests elsewhere without revealing significant content.

Further interaction with the /debug and /api endpoints displayed HTML pages requiring JavaScript to load fully, indicating that the application relies on JavaScript to render its contents.

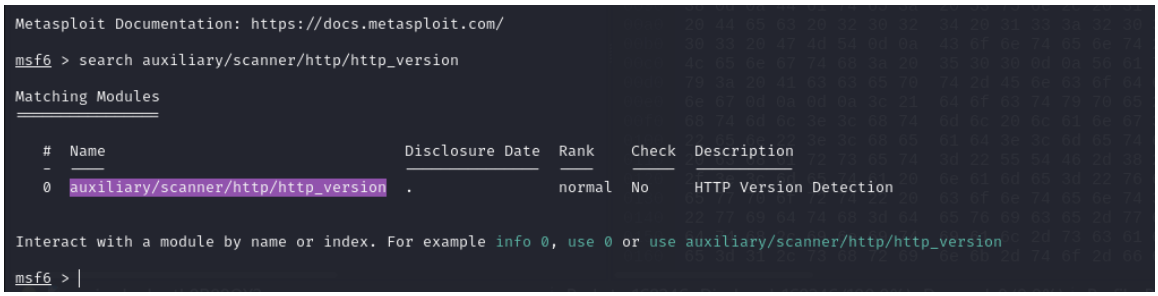


To validate the structure of certain JavaScript files, a manual request to /static/js/main.dccfd6b5.js was made, revealing obfuscated JavaScript content.

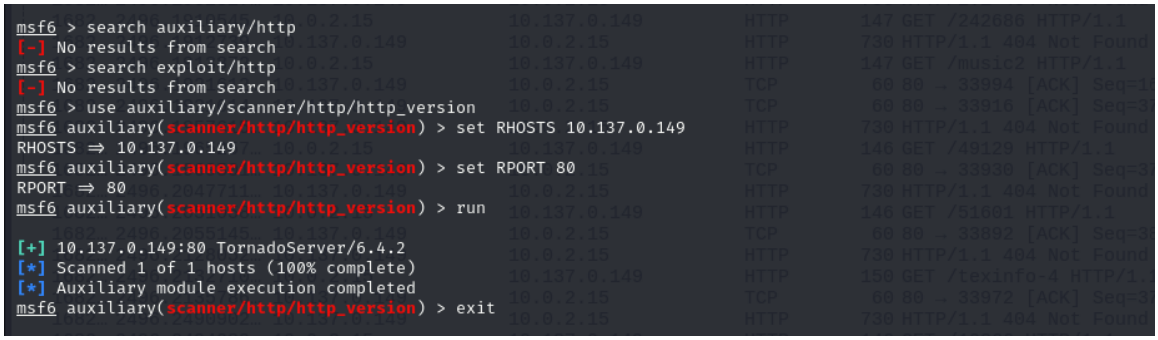


Metasploit Scans and Attempts ~

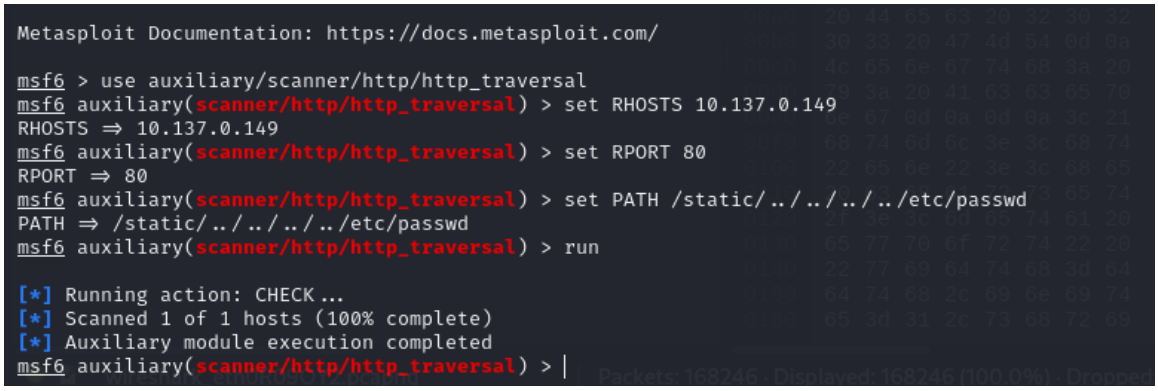
To expand the enumeration, Metasploit auxiliary modules were used. The first module, auxiliary/scanner/http/http_version, confirmed that the target server is TornadoServer/6.4.2 running on port 80.



Following this, a directory traversal attempt using the auxiliary/scanner/http/http_traversal module targeted the /static/../../../../etc/passwd file. Although the module execution completed successfully, there was no confirmation of data retrieval.



Further exploration focused on file upload capabilities through HTTP PUT. Using the auxiliary/scanner/http/http_put module, an attempt was made to upload a reverse PHP shell to the /static/ directory. The server responded with a 403 Forbidden error, indicating that the directory does not allow file uploads. Additional checks confirmed that no writable directories were identified.

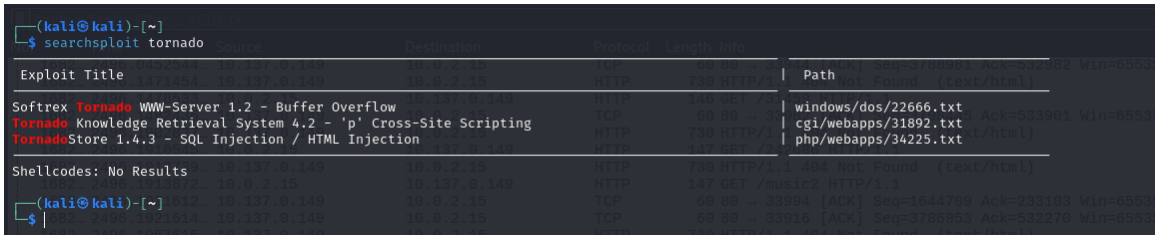


Web Vulnerability Scanning ~

A Nikto scan was performed to identify web server vulnerabilities or misconfigurations. Nikto detected missing security headers, including X-Frame-Options and X-Content-Type-Options, which could expose the server to clickjacking and MIME-sniffing attacks. However, the scan encountered multiple HTTP errors, limiting the depth of results.

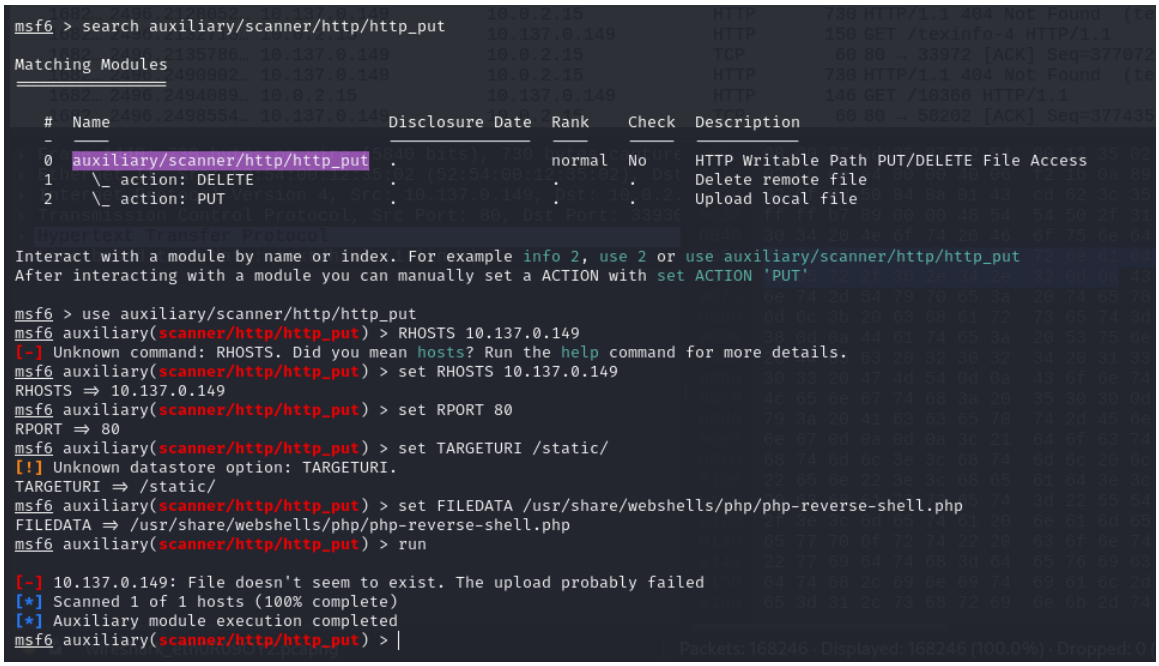
Manual Exploit Searches ~

To verify if there were any publicly known exploits for TornadoServer, Searchsploit was used. The results revealed older vulnerabilities related to buffer overflows and cross-site scripting (XSS), but they were unrelated to the 6.4.2 version running on the target system. No relevant shellcodes or modern exploits were identified.



Manual Command Injection and Debug Testing ~

Manual command injection attempts were made via the /debug endpoint by appending parameters like cmd=ls. The responses remained unchanged, returning the same JavaScript-dependent content as earlier, indicating that command execution is not directly possible. POST attempts to /api targeting the /etc/passwd file also failed, with connection resets preventing further exploration.



Conclusion ~

Despite multiple enumeration techniques and active testing using cURL, Metasploit, Nikto, and Searchsploit, no exploitable vulnerabilities were identified on the HTTP service of 10.137.0.149. The server confirmed as TornadoServer/6.4.2 does not allow unauthorized file uploads, directory traversal, or command injection. While minor misconfigurations like missing security headers were found, they do not present an immediate path for exploitation.

Further testing may involve tools capable of interacting with JavaScript dependent content (Burp Suite with headless browsers) and deeper fuzzing of endpoints for hidden vulnerabilities.