DOM-based open redirection

Domine: https://asupport.suivo.com/

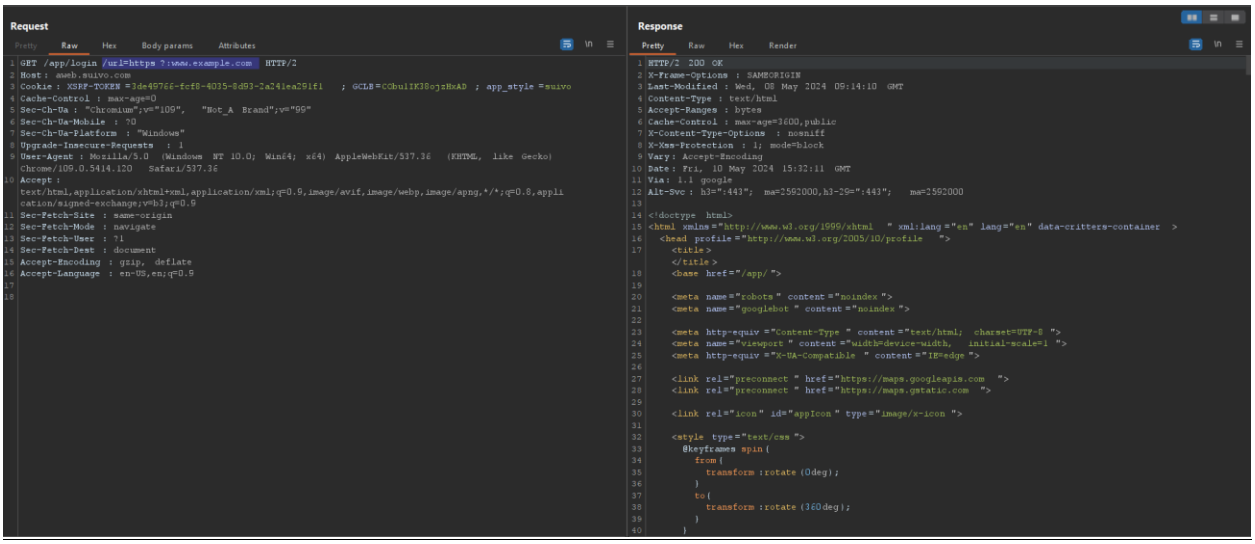
Description

The analysis of a request sent to the aweb.suivo.com host revealed potential protocol manipulation, wherein an additional URL parameter (/url=https?:www.example.comHTTP/2 HTTP/2) was appended to the request. The request was successful, receiving a response with a status code of 200 OK. This unexpected modification raises concerns regarding the security posture of the application handling the request.

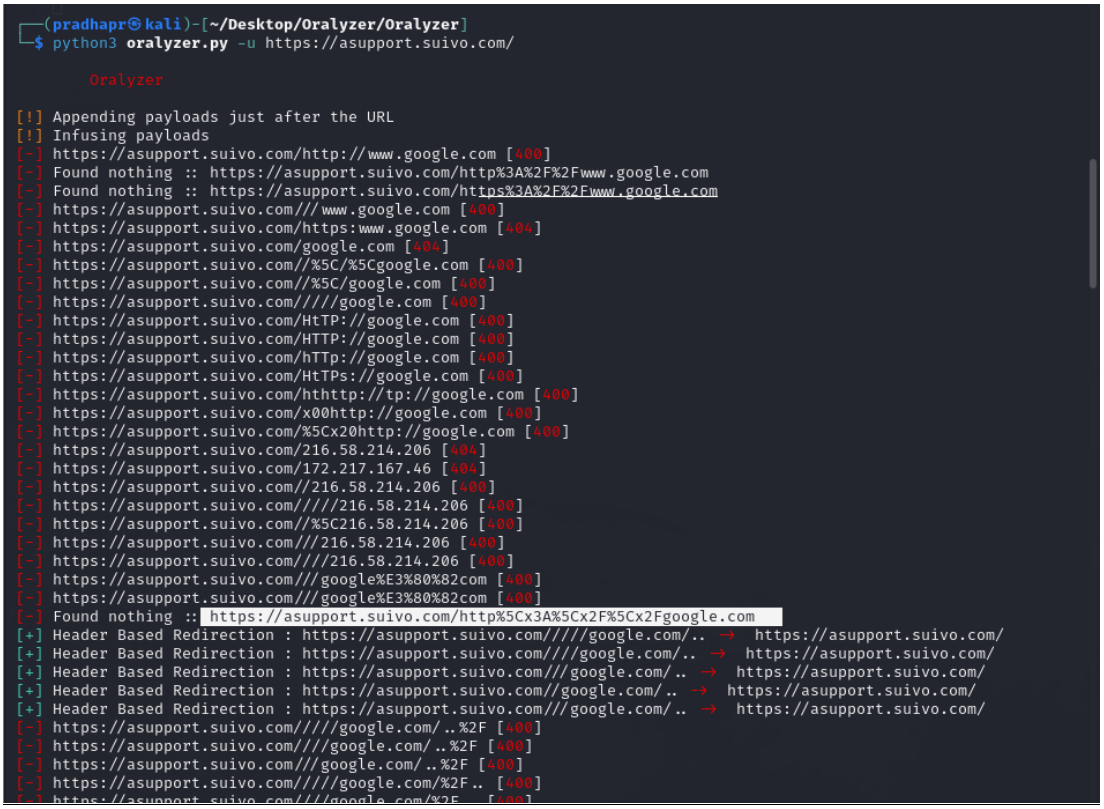
Affected Components

The application endpoint responsible for processing the modified request (/app/login/) and potentially other components involved in request handling and processing.

Proof of Concept



The inclusion of the /url=https?:www.example.com HTTP/2 parameter in the request resulted in a successful response from the server, indicating that the application processed the request without error. However, further investigation is necessary to ascertain the full extent of the vulnerability and potential exploitation scenarios.



Proposed Solution

Input Validation: Implement robust input validation mechanisms to sanitize user-supplied input effectively. Validate request parameters to ensure they conform to expected formats and do not contain unexpected or malicious content.

Protocol Compliance: Enforce strict adherence to protocol specifications to prevent manipulation or injection of protocol elements. Validate and sanitize all components of the request, including the URL parameters, headers, and request body, to mitigate the risk of protocol-based vulnerabilities.

Security Testing: Conduct comprehensive security testing, including vulnerability scanning, penetration testing, and code review, to identify and remediate vulnerabilities in the application's request handling and processing logic.

Security Headers: Implement security headers, such as Content Security Policy (CSP) and Strict-Transport-Security (HSTS), to enhance the security posture of the application and mitigate various types of attacks, including protocol manipulation.

Monitoring and Logging: Monitor incoming requests for suspicious patterns or anomalies that may indicate attempted exploitation of protocol manipulation vulnerabilities. Maintain detailed logs of request and response traffic for forensic analysis and incident response purposes.

Impact

The exploitation of protocol manipulation vulnerabilities could potentially lead to various security risks, including unauthorized access, data manipulation, and denial-of-service (DoS) attacks. Malicious actors may exploit these vulnerabilities to bypass security controls, manipulate application behavior, or compromise sensitive data.

Conclusion

Addressing the potential protocol manipulation vulnerability is crucial for safeguarding the security and integrity of the application and underlying systems. By implementing rigorous input validation, enforcing protocol compliance, and conducting regular security testing and monitoring, organizations can mitigate the risk of exploitation and ensure the resilience of their applications against evolving security threats.