What this report contains

Executive Summary

This section provides a non-technical overview of the scope of the project, the key findings from the testing work carried out and highlights the areas we see as most important or urgent in needing attention to improve the security of the system under test.

Target Information

This lists what was tested and the type of testing carried out.

Methodology

This is a short, technical description, of physical way that the testing was undertaken.

Issue Summary

This is a high level, technical, summary of each issue found. It's common for this section to contain multiple examples of the same type of issue as we list every instance of an issue we find so that there is a clear list of everything that needs to be fixed – i.e. If we find a form that is insecure it's possible that each individual field within the form has the same type of issue and so if the form has five fields; First name, Last name, Email address, Company name and comments these would be listed as 5 issues as each field needs attention.

Vulnerability by Severity

This is a visual representation of the different levels of issues found to provide a quick overview of the number of each type individually and in relation to the application as a whole.

Details of Individual Issues

This is a detailed list of each issue found containing, where possible:

- Affected assets – what in the system has the issues
- Severity – on a 5-point scale from 'Critical' to 'For information only'
- OWASP / CWE mapping – how the issue relates to security testing methodologies or databases
- Technical description – a detailed technical explanation of the issue
- Steps to reproduce – how to verify the issue exists and to check that it has been fixed
- Business impact – a non-technical description of the potential impact of the issue
- Solution and/or mitigation – what's required to fix the issue
- External technical reference links – additional information in the public domain to provide more insight into the type of issue, other ways to fix it etc.

As with the issue summary, every instance of an issue is listed individually to provide a detailed, step-by-step guide for all fixes required.

Executive Summary

Appsecco was contracted by Law Firm Services to conduct Web Application Security Testing to determine if there were security weaknesses in the LFS Gateway application and the new features implemented in the tenant applications that can render the environment insecure and allow an attacker to gain access to any data that is accessible via them or gain access to the underlying operating system.

The testing was carried out on 14th, 16th, 17th and 18th January 2019 on the testing setup of the LFS Identiy Server, LFS Gateway and the associated API servers and web applications

The OWASP Top-10 2013, OWASP Top-10 2017 and CWE was used as the reference frameworks to evaluate and categorise the discovered security issues.

Summary of Results

The new features implemented in the LFS application were tested extensively for any authentication and authorization weaknesses that could allow an attacker to access data across tenant boundaries by any vulnerabilities prevalent in the applications in scope.

The testing showed that the environment is secure and the authentication and authorization controls are implemented properly preventing an attacker from gaining access to the data or account of a different user either in the same tenant space or across the tenant boundary.

Low severity issues were discovered wherein the server side library used to generate the PDF summary of client questionnaire reveals the name and version number of the software and the Author who created the documents. Owing to no public exploits available for the version of the library and access required to leak this information, this issue has been rated low.

It was also found that the application does not clear localStorage within the browser when a logout request is made. The localStorage was found to contain potentially sensitive information in the form of client identifiers. Although not exploitable, this leak of information was recorded for brevity.

Conclusion

The application's environment was found to withstand injection and parameter modification attacks. The backend is built with authentication and authorisation in mind that prevented any cross tenant data access to become possible. This was tested by manipulating the tokens, cookies as well as the host headers that the API's use to send data back. The application exhibits sturdy access control and allows only valid and authorised users to perform actions.

The new features implemented within the application, namely "My Tasks", "My Quotes" and "Case Tracking", along with individual subsets of these features like file upload and download were all checked for authorisation issues, Insecure Direct Object References and manipulation of access tokens in an attempt to gain access to cross user and cross tenant data. We were unsuccessful in achieving cross boundary access with the credentials that were provided to us.

Multiple low severity vulnerabilities were discovered wherein the application leaks the name and version number of the library that was used to generate the PDF summary file under the "My Tasks" feature, via the PDF meta data section and the Authors name as David Green.

It was also found that the application does not clear the localStorage within the browser when a logout is performed.

The application is able to withstand attacks originating from an authenticated as well as an unauthenticated user session. No weaknesses were discovered that would allow cross user or cross boundary access to data or the account itself.