

---

# Summary Security Report

of the audit of "Spryker B2B/B2C" on behalf of Spryker Systems

---

For: Spryker Systems GmbH  
- from here on "Spryker Systems" -  
Julie-Wofthorn-Straße 1  
10115 Berlin  
Germany

By: SektionEins GmbH  
- from here on "SektionEins" -  
Hausdorffstraße 103  
53129 Bonn  
Germany

Author: Christian Horchert

# Audit Summary

Between September 26th and October 1st, 2018 SektionEins performed the first part and between November 7th and December 8th, 2018 the second part of a source code audit and penetration test of the variants of the Spryker Framework, called Spryker B2B and Spryker B2C.

The source code audit was conducted on the following code base:






- Spryker Core: <https://github.com/spryker/spryker>
- Spryker Shop: <https://github.com/spryker/spryker-shop>
- B2B: <https://github.com/spryker/suite-b2b-internal>
- B2C: <https://github.com/spryker/suite-b2c-internal>
- RabbitMq Adapter: <https://github.com/spryker/rabbit-mq>
- Spryker Install: <https://github.com/spryker/install>
- EventBehavior: <https://github.com/spryker/event-behavior>
- SynchronizationBehavior: <https://github.com/spryker/synchronization-behavior>
- UuidBehavior: <https://github.com/spryker/uuid-behavior>
- Loggly: <https://github.com/spryker-eco/loggly>

The main goal of the audit was an evaluation of the framework in order to protect it from attackers with and without knowledge of the source code. Security relevant issues were identified in the code and — if possible — verified in the test systems. The framework consists of roughly 500,000 lines of PHP code in roughly 15,000 files, without comments or empty lines and not including external libraries. SektionEins was provided with test systems for penetration testing. The test systems did not run over HTTPS, so none of the SSL-related functions have been tested.

During the source code audit no actual vulnerabilities could be identified while the penetration test revealed some problems, mainly Cross-Site Scripting (XSS), almost all of which are Stored XSS as well as hard-to-exploit vulnerabilities related to a DOM-based client-side JSON injection and one Cross-Site Request Forgery (CSRF). Other problems included outdated libraries and dependencies and some missing but recommended HTTP headers.

The safe use of functions and the lack of critical vulnerabilities such as SQL injection and shows that the framework seems very robust against a great range of attacks. The source code of this application is well structured and easy to read and extend. Spryker uses a functionality to make sure that no critical PHP functions are used which can easily be extended to cover other probable issues.

All of the high risk vulnerabilities and almost all of the medium risk vulnerabilities in the following overview have already been fixed in the version as of April 10th, 2019:

-  No critical vulnerability could be identified
-  2 high risk vulnerability could be identified
-  6 medium risk vulnerability could be identified
-  6 low risk vulnerability could be identified
-  4 issues have been marked as comments

For each of the security vulnerabilities a detailed risk analysis has been performed that is documented throughout the report. For more information please reach out for a more extensive version of this report to Spryker Systems.