Running Security Scans on Lithium Communities (security testing)

Running Security Scans on Lithium Communities - Security Testing & Reporting Policy and Best Practices

Version 2.0, Dated 5/13/2016.
Contact secops@lithium.com

Lithium Security Testing and Reporting Policy

We don't allow any kind of security testing on production systems. Security testing is permitted on approved staging sites only. All security testing is subject to the Lithium Security Testing and Reporting Policy posted at https://lithium.com/security and also subject to the terms of the Lithium External Security Assessment Agreement or EXSAA. Generally speaking if the scope of testing involved a remote automated scan then an EXSAA might not be required. However, depending on the types of tools used and/or should the testing involve manual penetration testing then an EXSAA agreement is required.

Standard Security Testing Disclosure:

Lithium customers are welcome to conduct their own security testing on approved staging sites only and subject to Lithium Security Testing and Reporting policy posted here: https://lithium.com/security. In case of a full pentest there must be an EXSAA agreement in place (attached here).

Please note that since we operate a multi-tenant/shared environment, ***all security testing must be conducted on the approved staging sites only and using the lowest possible speed/setting for you security scanner.***

Please be sure to read and familiarize yourself with the Lithium Security Testing and Reporting Policy before initiating the scan (https://www.lithium.com/security).

Testing Restrictions. Tester shall not do the following: (a) tests (i.e. submission of traffic) against any other servers other than the Hosts; (b) denial of service attacks against any servers or network equipment; (c) attempts at server reboots; (d) installation of bots, viruses, trojans, "rootkits" or other executables which may harm Lithium's services or systems; (e) attempt to access, modify, or delete information without Lithium's authorization. For purposes of clarification 2.5(e) refers to attempts such as, including but not limited to, testing for SQL injection using Union, Insert, Update, Delete, Drop, and Append; modifying or replacing a system binary in memory or on disk; modifying or removing copies of sensitive information such as: password hashes, database files and tables, encryption keys (SSH keys, PGP keys, X.509 certificate keys, Code Signing certificate keys, etc.); adding unauthorized system or user accounts, changing system configuration settings or account passwords, password or decryption brute force attempts, etc.

For clarification purpose, we don't restrict SQL injection testing except for destructive methods like drop, delete, etc. At the end of the day, SQL injection is SQL injection, there are safe methods to test it using industry standard tools and techniques. Similarly, in case of say remote code exec, there is no need to actually install any shellcode, there are clear and safe methods to test and prove such flaws.

Why Not To Run a Web Application Scan (WAS) on Production Communities

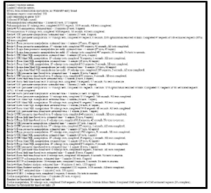
We don't allow any testing on production systems.

A number of people have asked this question before why is web application testing best done on staging or QA instances. I'll summarize the main points here and make them relevant to Lithium using real life examples:

1. **Shared Multi-tenant Platform (privacy and confidentiality concerns) –** we operate a shared multi-tenant SaaS platform. Even in staging sites we have shared instances where customers share clusters and unauthorized access could expose either other customer's data and/or internal Lithium specific confidential and proprietary elements. There are Testing Restrictions in place to ensure responsible security testing on our systems.
2. **Lithium Security Monitoring (platform protection measures) –** our security monitoring process generates alerts to respond to suspected web attacks on our production sites. Most security scanners and security testing will generate traffic that is virtually indistinguishable from malicious attack traffic. In order to protect our platform any IP addresses that generate attack or malicious traffic maybe blacklisted.
3. **Test Data Injection (integrity and stability concerns) –** automated WAS scanning injects test data that may not be appropriate for a live community. This type of activity can confuse users and cause unnecessary workload for community moderators to remove hundreds

or thousands of test entries from the community. An instance of this occurred in 2010 when we scanned Lithoshpere (Lithium Support Community) which resulted in many strange looking posts on the community as well as kudos and tag entries placed randomly by the security scanner.

4. **<u>Resource Issues (performance concerns) –</u>** most WAS scanners will try to crawl the entire community and then use hundreds of test on hundreds of forms and several fields within each from resulting in resource issues and potentially starving out legitimate users and other connection attempts. For instance one type of scanner commonly used in the industry found roughly 8400 links on our test community and used a combined total of 589 tests with 3665 inputs (fields to inspect) during the test. That can result in millions of requests in a very short period of time. (In reality though, we limited the scan to only 5% of the total links since the majority of them are essentially replicas of each other and that took about 2 hours to complete with roughly 120K requests. If we had let the entire test cycle to complete as the scanner intended it would have taken roughly 40 hours to complete and would have generated over 2.2MM requests to the community.)



5. **<u>Application workflow triggers (administrative overhead)</u>** – some community features may generate operational workflows such as creating a service request ticket for support. Although this is not a huge concern for most communities but new features are added every month and escalations may be allowed on some communities to automatically create a support ticket if a post goes unanswered for too long. A typical security scanner can submit these requests hundreds of times resulting in overhead and wasted time and resources.

Recommendations For Scanning Lithium Communities With a WAS Scanner

1. ***Use a staging instance for security testing***. On a typical production site there can be several boards and each user post on the board appears as a link to the WAS scanner. There is no real benefit in testing every single post since the internal structure of each board and message post is essentially the same. Same goes for the other features available on communities as well. Staging sites are typically less complex content-wise so there will be less duplication in the number of links detected and thus more accurate and complete testing of all features and links on the site is possible.
2. ***Use a targeted whitelist approach to testing the application***. Although, a URL whitelist approach is more cumbersome to maintain but it's more targeted and will make sure that all the pages and features you would like to scan are properly tested. It is recommended that you familiarize yourself with the community to know which features and pages (links or URLs) to test.
3. ***Prevent information overload***. Limit the maximum number of links to test to a reasonable size. There are a few practical reasons for this, first, as explained in the first point above most links are just replicas and testing essentially the same link repeatedly is overkill. Second, the scan is more likely to complete within a reasonable timeframe, and three, the resulting scan report will be more likely a reasonable size to digest. Depending on the vendor and scan settings, it's not uncommon that a scan limited to 350 links is going to generate roughly a 100+ page report. Imagine what the report is going to look like for a scan five or six times larger.
4. ***Create a security a baseline of your community and update it regularly***. It's not uncommon that numerous false positives will be reported in each WAS scan, and going over hundreds of pages of either irrelevant or previously reviewed information on a regular basis is not very efficient. The baseline approach will allow you to compare the latest WAS scan results to your previous baseline and quickly assimilate the new information.
5. ***Read and follow the Lithium Security Testing and Reporting Policy***. Last but not least, read and follow the Lithium Security Testing and Reporting Policy at [https://www.lithium.com/security]

Document Control Section

Verion 2.0, 5/13/2016, Faisal Khan
-updated why-not-to section

-Added links to our security testing policy

Verion 1.0, 1/5/2012, Faisal Khan
-this document created.