



Key Requirements for Selecting an Application-Aware Testing Solution



Application-Aware Networks Require Application-Aware Testing Solutions

Introduction

Service providers, enterprises, network equipment manufacturers and government network operators are using application-aware systems to detect and classify the applications running on their networks to enforce policies that maintain a high quality of service (QoS) and security. However, with applications, such as Netflix, FaceTime, Facebook, Skype, Twitter and BitTorrent, evolving and being introduced at an accelerated pace, these operators and their vendors are struggling to test their network configurations and component systems under real, production-like conditions.

When more than 80% of your traffic is driven by applications, it's easy to see why legacy test tools that focus on network protocols are inadequate. These tools generate artificial application transactions, using random data in the application payload, to try to mimic the applications.

This is a fundamentally flawed testing foundation - you simply can't test real networks with fake traffic.

This document captures the key requirements for helping you select an application-aware testing solution to effectively test the scalability, functionality and security of today's intelligent networks.

The following table summarizes these key requirements:

Category	Requirement	Mu Test Solution	Other Solutions
Accuracy	Application re-creation (not just at L2-L4)	YES	
	Real TCP sessions (e.g. congestion and flow control)	YES	
	Consistency and repeatability of test results	YES	
Speed	1,000s of ready-to-run application tests, available online	YES	
	Rapid creation of new tests	YES	
	Unified platform for scale, security and functional testing	YES	
	Automatic creation of 1,000s of security and fuzz tests	YES	
Flexibility	Ability to rapidly add new tests & enhance existing tests	YES	
	Ability to view & modify entire application payload	YES	
	Ability to parameterize any field on all layers (L2-L7)	YES	

Accuracy - Test Your Network, Your Transactions, Your Reality

1) Re-creating the latest application flows from the production environment

Can your application scenarios be replicated in the lab in a realistic manner? What happens when you run the application traffic against a real application server? Will the application-aware network detect the recreated applications? Are the replays truly stateful and useful or are they superficial – like the information you would receive from a tool such as Tomahawk?

Why it's important – Signatures on the application-aware system you are testing are designed to inspect real application payloads. This enables them to better classify, prioritize and protect the network. If the test tool is unable to maintain application-level state, with regards to cookies, session IDs, NAT translations, ALG-driven modifications, authentication challenges, lengths and checksums as it goes about recreating the application flows, the application-aware network will not detect the application or react to it in a realistic manner. This is further complicated by the fact that this also needs to happen at high speeds and for all versions of the applications in use. Consider, there are thousands of applications and different versions currently running over today's networks. To accurately detect these flows is a huge challenge, but one that must be met if you are to get an accurate picture of what will happen on your production network.

Legacy test tools – Legacy test tools primarily focus on network protocols (L2-L4), making them very limited when it comes to application-level tests. To address this, they generate synthetic application traffic, with random strings of 1s and 0s in the application payload, which creates results that aren't grounded in reality. To illustrate this, take a network capture of a popular mobile or web-based application. If you were to run it against a real application server using legacy load testing tools, you would find that this "dumb" packet replay isn't successful. This is because it's not realistic. It lacks any application state, so the application-aware system will either reject the flow or misclassify the application type, neither of which helps you accurately test using that application

Mu Dynamics – In comparison, the Mu solution accurately recreates applications using test cases derived from real application traffic, with payloads obtained directly from the production network. As a result, it can communicate with real servers, such as a database, VOIP, video, or NFS server, etc. and maintain true application state. The server under test does not differentiate between Mu-generated traffic and the real applications, giving you an accurate recreation that is true to real application behavior.

2) *Use of real TCP sessions*

Are the TCP sessions truly realistic in the face of congestion?

Why it's important – The test tool must use real TCP stacks during scale testing to understand what will happen when at capacity. In the real world, as targets become stressed and as networks become congested, TCP flow and congestion control algorithms (e.g. Van Jacobson) are utilized. This affects the behavior of both the test tool and the target, which will alter its behavior to deal with the congestion. In order to maintain realism in the test lab, it is critical that the test tool behave exactly as a real client would on the production network. Failure to do so will result in inaccurate test scenarios and misleading results.

Legacy test tools – Many legacy load test tools use custom-developed, hardware-accelerated TCP implementations that are optimized for maximum network throughput. As a result, when congestion control mechanisms are activated on the network, they are ignored by the tool, and the load generation continues unchanged. The behavior of the tests is therefore artificial. Time-outs and other congestion situations you would expect to see in a production environment are simply not taken under consideration, which means you are unable to test the true behavior of the network.

Mu Dynamics – With the Mu solution, real Linux BSD TCP stacks are used to conduct scale tests. These tests behave exactly as real clients would in the face of congestion and flow control situations. Time-outs and congestion errors are handled appropriately, and test replay is altered accordingly, so you know exactly what would happen in a production environment.

3) *Consistency and repeatability of test results*

Are the test results repeatable, even when no changes are made to the system under test?

Why it's important – Testing is inherently repetitive and iterative. Test results are often compared between runs to determine whether system changes are effective or not. Repeatable tests with consistent results allow problems to be isolated in an efficient and deterministic manner. The validity of the test results is a fundamental requirement for any testing solution.

Legacy test tools – Many legacy load testing tools show large variances between test runs for the same test configuration. This is a major challenge for scale testing as test tools are often designed for maximum throughput and performance (bits per second), at the expense of repeatability.

Mu Dynamics – The Mu solution is designed to be consistent and repeatable during test execution.

Speed - Test Immediately, No Waiting, No Dependencies

1) Quick access to thousands of ready-to-run application tests

Do you have what you need to start testing right away? How long does it take your test vendor to develop tests for new applications? What kinds of resources can you access to broaden the depth and breadth of your tests?

Why it's important – There are hundreds of thousands of applications on today's networks, and this number is growing rapidly. There is no time to wait - The quicker you can test, the quicker you can identify and resolve any potential issues. Ready-to-run test cases that are derived from real world applications help you jump-start the testing process. They can help test teams who do not have access to many types of application flows get productive quicker. The larger and more diverse the set of test cases, the easier it is for test teams to reflect the ever-evolving types of applications seen on their network.

Legacy test tools – Existing test tools provide only a handful of application test cases. Most of them are focused exclusively on network protocols (L2-L4) as opposed to the applications. Their application tests generate fake application traffic, which is an ineffective way to test. As new applications and protocols emerge, you may have to wait weeks, even months, for test vendors to release new tests.

Mu Dynamics – The Mu TestCloud (www.testcloud.mudynamics.com) offers the industry's only collaborative test platform that provides thousands of ready-to-run application tests. You can download tests for all of today's most popular consumer and business apps, including Facebook, Skype, Netflix, Twitter, BitTorrent, VMware and Google Docs. The list continues to grow, with hundreds of new tests added and updated every month by members of the community and Mu. As a result you can keep up with versions and client types to future-proof your network against tomorrow's new applications.

2) Speed of test creation and validation

How will you create new tests? How long does it take to validate each test case? Does the test tool provide you complete visibility into the application layer, including ALL of the content and ALL of the application payload?

Why it's important – During a test, it's hard to see what's going on at the application layer. If something happens, it's difficult for you to determine what's actually being sent on the wire and what the test tool is receiving. You often have to resort to packet captures and debugging sessions, using tools such as Wireshark, to validate whether the application traffic is legitimate, erroneous, or even contains an application payload at all. The more time you have to spend validating, the less time you have to run new tests to maximize your test coverage.

Legacy test tools - Legacy test tools do not provide complete visibility into the application layer, which means you do not know what the tool is sending over the wire, nor can you control it at the application layer.

Mu Dynamics - Mu provides the most holistic view of all the transactions, field types, payload, and content. As a result, there is no ambiguity of what is being sent or received over the wire during tests of the application-aware network, so you can quickly validate the results and move on to the next test. Mu provides two modes to view the application layer data:

- Scenario Editor Mode: A GUI with full application decode
- Source View: A display of all the transactions, content, and payload in a canonical text view

3) *Unified solution for scale, security and functional testing*

Can the test solution be used for scale, security and functional testing on a single unified platform?

Can the same test assets be used for all types of testing?

Why it's important – It is not uncommon for testing teams to use multiple test tools, from multiple vendors to conduct different types of tests. This adds unnecessary complexity and is very inefficient; just training test engineers on all these various tools can be time-consuming and expensive. Test teams want to maximize the investment they make in testing tools and assets, by using them to conduct all their testing.

Legacy test tools – Many of the existing tools are specialized for a single test discipline. Test assets created by one product are not usable by another, forcing you to waste time by creating duplicative tests.

Mu Dynamics – With the Mu solution, the same test assets can be re-used for scale, security and functional testing – all of which is delivered on the same common platform. This makes testing extremely efficient and streamlined and maximizes the return on your investment.

4) *Automatic creation of thousands of fuzz tests*

Can the same application test cases be automatically converted into intelligent fuzz test cases?

Why it's important - Fuzz testing is widely recognized by industry experts as the only way to quickly and effectively determines the resiliency of an application-aware system. With the sheer number of applications and protocol implementations available, application-aware systems need to be able handle unexpected field data and values in a graceful manner. By testing for the unexpected, otherwise known as negative testing or fuzz testing, you can identify the ability of the system to handle deviations from the norm.

Legacy test tools - Many traditional test tools do not support fuzz testing. Those that do use random “bit flipping” and flooding of malformed packets. This is the most rudimentary form of fuzzing, which means they do not have the capability to automatically build intelligent fuzz test cases for ALL the field types at the application layer. This is vital when testing the robustness of any application-aware device.

Mu Dynamics – Mu offers a best-of-breed fuzzing solution. It automatically converts any application flow into thousands of intelligent fuzz test cases for ALL field types at the application layer. The result is increased code coverage, increased test coverage, and reduced time for test creation, fault isolation and remediation.

Flexibility - Test Any Application, Any Protocol, At Any Time

1) Ability to add coverage for new apps, new versions, and protocol tests

Is the test solution flexible enough to be able to add new test cases on-demand? Can the test tool adapt to the test plan versus the other way around?

Why it's important -Test requirements are constantly changing as the production environment evolves, with new and upgraded applications and protocol flows appearing on the network constantly. Test teams need a future-proof solution that can recreate these new applications as soon as they become available.

Legacy test tools - Legacy test tools cannot support many of the latest applications. The user has to wait for the test tool vendor to develop custom support for the new application, which can often take months, if done at all.

Mu Dynamics - In comparison, because Mu bases test creation on actual production traffic, it's flexible and can adapt to your environment and test plans. Using Mu you can recreate any application or new version immediately. This approach provides the most comprehensive test coverage, both now and in the future.

2) Ability to modify application content

Does the test tool allow the user to easily modify ANY and ALL application fields, content, and payload?

Why it's important –You need the flexibility to modify or add any portion of an application payload, message or field, so you can manipulate your tests to quickly identify potential issues in your production environment. For examples, you may want to modify port numbers, embed IP addresses, session IDs, or URLs, etc. to understand what will happen in a real-world scenario. Without this capability, the tool is rendered static and is not representative of the dynamic nature of your application-aware network.

Legacy test tools - Legacy tools do not provide complete visibility into the application layer and do not allow you to modify any payload or message field.

Mu Dynamics - In comparison, Mu allows you to select any field and edit or modify ALL the content and payload at the application layer. For example, you can modify the contents of the HTTP payload. You could craft a valid GET request, followed by several illegal GET requests in the same transport to see if it by-passes the security inspection engine. As a result, you would be able to uncover and then fix potential weaknesses in your systems or implementation.

3) *Parameterization of fields and data*

Does the test tool allow you to parameterize ANY protocol field at the application layer? Can the tool supply a set of values for data-driven feature or scale/load testing?

Why it's important – Data-driven testing allows you to quickly test different inputs and outcomes. You may want to supply alternate values, such as URLs, user names, passwords, folder names, phone numbers, ports, IP addresses, etc. to test the target's capabilities and understand how it will respond when it encounters these values in the production environment.

Legacy test tools - Legacy test tools do not allow the user to select ALL protocol fields at the application layer and parameterize them for data-driven testing. This is only available for a small number of pre-defined fields.

Mu Dynamics – The Mu solution provides this capability across all layers, from L2-L7. Once the protocol field at the application layer has been parameterized, you can supply a set of custom values, using a list, range, or random values; you can even import a spreadsheet of values. For example, you could test your URL filtering capabilities by modifying the URL field of the payload with the click of a button to see what happens when thousands of alternate values are supplied. This ensures you have the breadth of testing you need to confidently deploy your applications and systems in your application-aware network.

Conclusion

To effectively test today's application-aware networks you require a testing solution that can recreate any application, any protocol, at any time. You need the quickest, simplest solution to increase your test coverage, accelerate your test cycles and accurately recreate the production reality of your network.

With Mu, you have the only unified test solution that can accurately recreate the real application traffic of your network for scale, security and functional testing. With thousands of ready-to-run tests covering the most popular applications and the ability to create new tests on demand, you finally have what you need to keep pace with the rapid changes in your application-aware network environment.

About Mu Dynamics

Mu Dynamics is the leader and pioneer in testing and validating application-aware networks. Our innovative solutions enable customers to rapidly recreate applications and determine the impact to their network. Hundreds of service providers, enterprises, government agencies and network equipment manufacturers use Mu to ensure that their networks are functional, scalable and secure.



Web: www.mudynamics.com

Address: 686 W. Maude Avenue, Suite 104, Sunnyvale, CA 94085, USA

Phone: 866-276-4640 or 408-329-6330

Fax: 408-329-6317

Copyright © 2011 Mu Dynamics. All rights reserved. Mu Dynamics, Mu Test Suite, Mu-4000, Mu-8000, Mu Dynamics logo, and Innovate with Confidence are trademarks of Mu Dynamics.