



FOCAL POINTS: Sponsored Links
Focus on Office Business
Applications (OBA)

SEARCH

Site Archive (Complete)

ABOUT US | CONTACT | ADVERTISE | SUBSCRIBE | SOURCE CODE | CURRENT PRINT ISSUE | NEWSLETTERS | RESOURCES | BLOGS | PODCASTS | CAREERS

Security

April 01, 2008

Fuzzing, Model-based Testing, and Security

Jonathan Erickson

- [Email](#)
- [Print](#)
- [Reprint](#)
-
- add to:
- [Del.icio.us](#)
- [Slashdot](#)
- [Digg](#)
- [Y! MyWeb](#)
- [Google](#)
- [Blink](#)
- [Spurl](#)
- [Furl](#)

FREE WHITEPAPER

Using Adobe Flash technology in embedded devices can save HMI design time by up to 50%. [Download this FREE WHITEPAPER: Using Adobe Flash to Create Dynamic Human Machine Interfaces](#)

DR. DOBB'S CAREER CENTER

Ready to take that job and shove it? [open](#) | [close](#)

DEPARTMENTS

- [Home](#)
- [Architecture & Design](#)
- [C/C++](#)
- [Database](#)
- [Development Tools](#)
- [Embedded Systems](#)
- [High Performance Computing](#)
- [Java](#)
- [Mobility](#)
- [Open Source](#)
- [Security](#)
- [Web Development](#)
- [Windows/.NET](#)

Security testing via random inputs



Joining us is Ari Takanen, founder and CTO of security testing firm [Codenomicon](#).

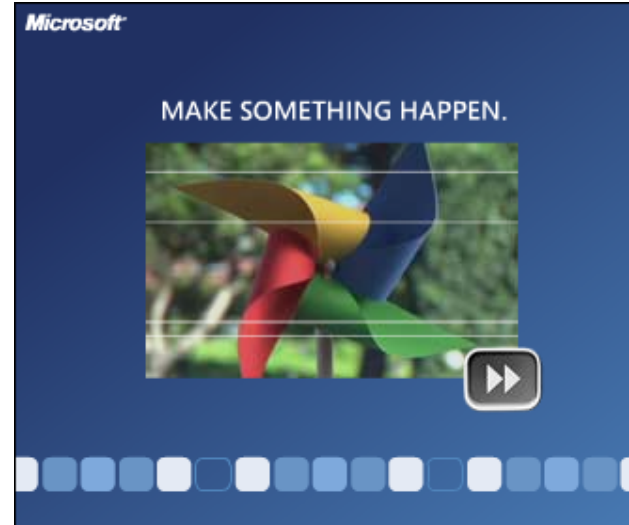
DDJ: Ari, what's "model-based fuzzing" and why is it important?

AT: Model-based fuzzing has many names. Fuzzing itself refers to a security testing approach where random or semi-random inputs are sent to software in attempt to

crash it. The term itself was coined by Dr. Miller in the early 1990s to describe his command-line fuzzer, which was used to test various commands in different operating systems. Most random-based fuzzers generate tests based on a flat and simple template, such as an audio file or a basic web (HTTP) request.

In model-based security testing, also known as grammar or syntax testing, the inputs to the software are modeled using context-free languages, which can describe complex protocols, such as encryption sequences for setting up SSL or TLS sessions. The syntax testing approach was probably first described by Dr. Boris Beizer in 1990, in his book, [Software Testing Techniques](#). Today, for security testing purposes, the models used in syntax testing are augmented with intelligent and optimized anomalies that will trigger the vulnerabilities in code.

Model-based fuzzing is a rather confusing term as it mixes up the



← sponsored

Resource Center for Microsoft® Silverlight™

MICROSITES

FEATURED TOPIC

INFO-LINK

[Learn how to build OBAs using the Microsoft Office system.](#)

[Using Adobe Flash to Create](#)

systematic approach of model-based testing and randomness as indicated by the term fuzzing. Since 1998, Codenomicon has referred to this approach as "robustness testing," because the term fuzzing still indicates ad-hoc and random tests. Randomness in testing is always a bad idea and would do a huge disservice to the industry. The term "robustness testing" has been picked up in the quality assurance domain. But the term "fuzzing" has stuck to being used by the security testers and that is difficult to change.

DDJ: Positive testing and Negative testing. What are they and what do they have to do with software?

AT: In traditional positive testing, test cases are built from positive requirements, resulting in test cases where you try valid inputs to validate the correctness of the functionality. Examples of positive testing include conformance tests and performance tests. For example, the tests can attempt to login into a service using a correct user name and password and for the performance assessment can run multiple test sequences in parallel.

In negative testing, each use case or requirement is reversed into infinite amount of tests trying out all unexpected inputs that the software can take. Examples of security tests include trying out long strings for the user name and password, tests for format string vulnerabilities (%s) and dictionary, or brute force tests to guess the correct password. Fuzzing in all forms is negative testing in the sense that you cannot define the test with a positive requirement. A negative requirement almost always includes the words "SHOULD NOT" or "MUST NOT," such as "it should not be possible to login with a bad password." Testing for a negative requirement requires skill from the tester to define the inputs that he will use to validate the requirement. Typically, these tests are either ad-hoc, or executed using predefined test suites, or conducted with commercial fuzzing tools.

DDJ: In terms of wireless communication, what currently makes security experts nervous?

AT: Wireless networks are always open, and it is very difficult to locate the attacker if he chooses to fuzz a wireless network. The fuzz test cases often take place before any authentication is performed or below the IP layer, and are therefore invisible to detection tools that only look at the application protocols in the communications. The attacker is almost always completely anonymous. Wireless attacks can also come from long distances. For example, attack tools have been demonstrated where Bluetooth devices are attacked several miles away from the target system, even though the wireless connection has been designed for short distances. A security problem in a wireless device can result in total corruption or crash of the system, but tailored active attacks, such as worms or viruses, are not impossible either.

DDJ: Is there a web site that readers can go to for more information on these topics?



[Dynamic Human Machine Interfaces - FREE WHITEPAPER](#)

[THE DOBBS CHALLENGE: Mod The Game, Win \\$10,000!](#)

[THE DOBBS CHALLENGE: Mod The Game, Win \\$10,000!](#)

ADDITIONAL TOPICS

["Get the 411 on OBA PDQ! Learn to build and deploy Microsoft Office Business Applications"](#)

[Are you a Windows programmer? Get the direct route to design interactive environments. Learn Silverlight here.](#)



MARKETPLACE (Sponsored Links)

[Workflow Enabled Help Desk & IT Service Management](#)

Automate service desk activities and integrate processes across IT. Learn more here.

[Deliver REMOTE SUPPORT Easily. Try WebEx FREE!](#)

DOWNLOAD WEBEX SUPPORT CENTER FREE! Deliver efficient, effective support. CRUSH SUPPORT LOG JAMS!

[WinDev 11 - Powerful IDE](#)

Develop 10 times faster ! ALM, IDE, .Net, RAD, 5GL, Database, 5GL, 64-bit, etc. Free Express version

[We Buy & Sell Used Cisco](#)

Safecount.net

TAKE A SURVEY

RESEARCH PURPOSES ONLY

Safecount.net

The Dobbs Challenge Game

AT: Yes, they can start with Codenomicon's [Buzz on Fuzz](#) page.

RELATED ARTICLES

- [Stochastic Optimization Models & Natural Disasters](#)
- [Microsoft Joins MIT Kerberos Consortium](#)
- [Developer Diaries](#)
- [Where Are the Clients In a SOA?](#)
- [Conversations: Jon Bentley](#)

TOP 5 ARTICLES

- [Random Numbers in a Range Using Generic Programming](#)
- [Anatomy of a Failed Agile Adoption](#)
- [The Silverlight 2.0 Security Model](#)

Hula Networks is overstocked on many items including, used Cisco, Juniper, Foundry and Extreme netwo...

[Hyena - Windows NT/2000/2003 Administration Softwa...](#)

Download the award-winning Hyena enterprise system administration tool for Windows NT/2000/2003/XP t...

[Advertise With Us](#)



Use Your Favorite Tools.

Explore the flexible Silverlight™ programming model and AJAX, Visual Basic®, JavaScript, Visual C#®, Python, and Ruby support.



[RSS](#) |

© 2008 [Think Services](#), [Privacy Policy](#), [Terms of Service](#), [United Business Media](#)

Comments about the web site: webmaster@ddj.com

Related Sites: [DotNetJunkies](#), [SD Expo](#), [SqlJunkies](#)