



WHITE PAPER

Black Box Testing and Codenomicon DEFENSICS

Jon Oltsik, Senior Analyst

April, 2008

Table of Contents

Table of Contents	i
Executive Summary	1
Security Testing Creates Problems	1
Black Box Testing For Security	4
The Case For Codenomicon	4
Black Box Testing Improves Quality and Efficiency.....	5
Why Codenomicon?.....	5
The Benefits of Black Box Testing and Codenomicon DEFENSICS.....	6
The Bottom Line	7

All trademark names are property of their respective companies. Information contained in this publication has been obtained by sources The Enterprise Strategy Group (ESG) considers to be reliable but is not warranted by ESG. This publication may contain opinions of ESG, which are subject to change from time to time. This publication is copyrighted by The Enterprise Strategy Group, Inc. Any reproduction or redistribution of this publication, in whole or in part, whether in hard-copy format, electronically, or otherwise to persons not authorized to receive it, without the express consent of the Enterprise Strategy Group, Inc., is in violation of U.S. copyright law and will be subject to an action for civil damages and, if applicable, criminal prosecution. Should you have any questions, please contact ESG Client Relations at (508) 482-0188. This ESG White Paper was developed with the assistance and funding of Codenomicon

Executive Summary

Ten years ago, testing software for security vulnerabilities was an afterthought. Few companies bothered to put their code through any security testing at all while others tested as little as they possibly could. Even firms with more pronounced security needs often lacked the tools and skills to really expose security bugs and discover vulnerabilities.

Those days seem like another era. This white paper discusses four main theories:

- **Security threats demand improved testing.** A combination of sophisticated threats, cyber-crime, publicly disclosed breaches, and open access to networked applications make security protection an essential requirement for network devices and applications. This reality has actually changed the way many technology firms design, develop, and test their software.
- **Black box testing is catching on.** Many software test engineers have embraced black box testing as a way to test system behavior by exercising protocols and interfaces with tools that are external to the actual software being tested. Firms often start their black box testing processes by using open source and freeware tools.
- **The road often leads to commercial black box systems.** As users gain experience with black box testing tools, they often find that open source and freeware can't meet their needs for testing network and application protocols as well as assorted interfaces. Rather than customize open source and freeware, test engineers often purchase commercial black box testing tools.
- **Implementing Codenomicon black box tools can result in numerous benefits.** In preparation for writing this white paper, ESG spoke with several customers of Codenomicon, one of the industry leaders in the black box testing tools market. ESG found that adoption of Codenomicon's DEFENSICS preemptive testing and robustness testing solutions produced a series of positive benefits including improved software quality, accelerated testing cycles, and lower overall costs associated with finding and fixing software bugs.

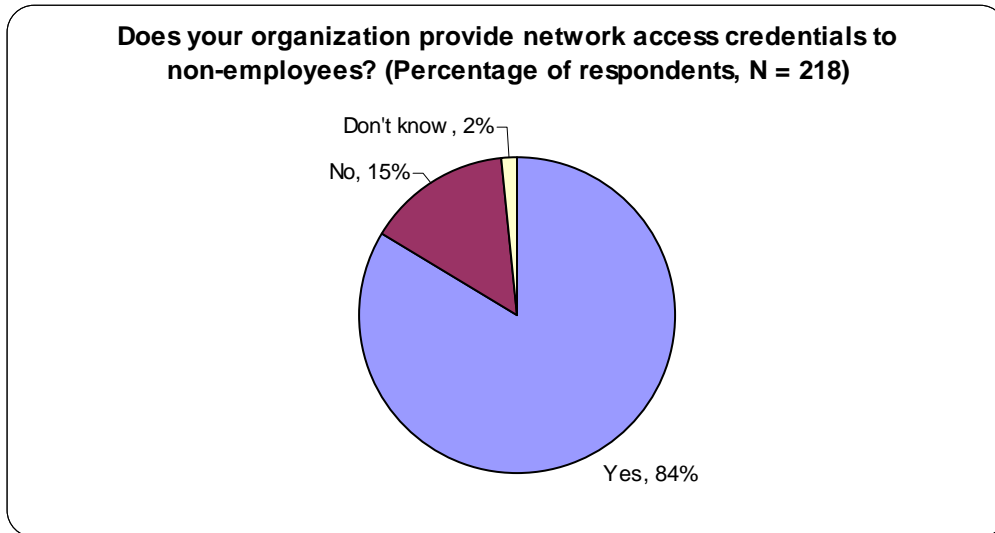
Security Testing Creates Problems

It is often said that developing good software is more of an art than a science. Software developers are often tasked with understanding complex business processes, operational requirements, or reporting needs and then writing software to add value and automate these activities. Good software development requires a combination of creativity, business savvy and highly skilled individuals.

While creativity reigns supreme in functional software design and development, software testing demands process orientation and discipline. Simply stated, a test engineer's role is to "break" the software. In other words, software testing should examine every function, data input, interface, and protocol to make sure that the software produces a desired and expected result in all cases. In this regard, software testing has grown more rigorous and automated over time. Manual processes are now automated to the point where up to 95% of testing is done through custom scripts, source code crawlers, and automated software testing tools. Automated software testing also takes advantage of pervasive high performance hardware to put software through a high volume of test cases within a limited testing time.

Ironically, as functional, quality, performance, and scalability software testing went through a cycle of automation and improvement, security testing lagged behind. This may have been a function of existing habits and history. Most enterprise software was initially developed to serve the needs of trusted insiders laboring safely behind the firewall, but this is no longer the case. According to ESG Research, a vast majority of large organizations now provide outside users such as business partners, consultants, and suppliers with access to their networks and applications, exposing these assets to new and dangerous threats (see Figure 1).

FIGURE 1. MOST ORGANIZATIONS PROVIDE NETWORK ACCESS TO OUTSIDERS

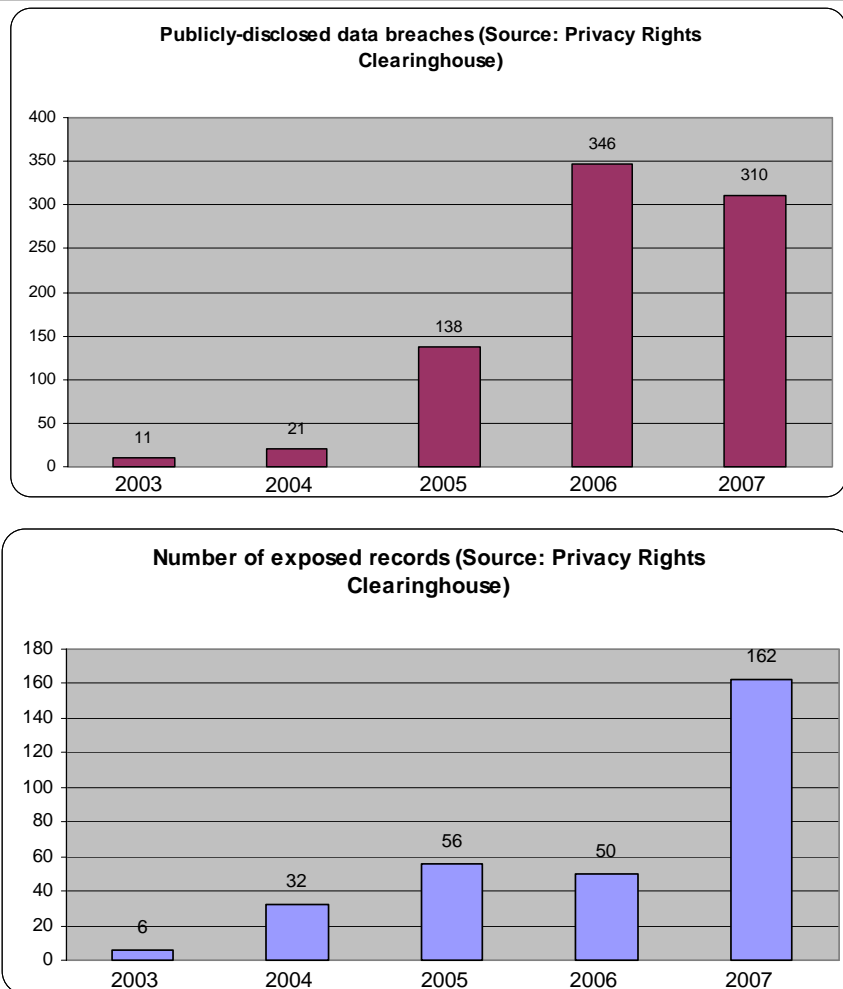


Source: Enterprise Strategy Group, 2008

This transition isn't trivial; as applications and devices are opened up to the Internet, they become far more exposed to security threats because of:

- **A growing cyber-crime economy.** According to a recent article published in the *Harvard Business Review*, some experts estimate that criminal cyber services may have netted as much as \$1.5 billion in 2007. Just how do they "earn" their money? Typically through a combination of extortion, on-line scams, widespread malware propagation, and targeted attacks. The cyber black market economy also has a sophisticated "division of labor" component to it as well. Specialists in malware writing may join forces with others who focus on unpublished vulnerabilities, protocol manipulation, or botnets to launch attacks that spread throughout the Internet or target a single Wall Street bank. As these cyber adversaries grow more pervasive and erudite, security testing becomes a critical part of a layered security defense.
- **Publicly-disclosed breaches.** According to the Privacy Rights Clearinghouse, there were a total of 310 publicly-disclosed breaches in 2007 (source: www.privacyrights.org). This seems like a slight improvement from the 346 breaches of 2006 but looks can be deceiving. While the number of events did improve, there were more than 3 times as many personal records exposed in 2007 than 2006 (see Figure 2). ESG estimates that a publicly-disclosed breach can cost organizations between \$30 and \$150 per record for activities like customer notification, postage, credit protection, etc. With the risks and costs associated with these kinds of publicly-disclosed breaches, improved security testing seems like a prudent insurance policy.

FIGURE 2. PUBLICLY-DISCLOSED BREACH INCIDENTS EXACERBATE THE NEED FOR SECURITY TESTING



Source: Enterprise Strategy Group, 2008

- **Converged networks.** As voice, video, and data networks become synonymous with IP, security must encompass more types of devices and protocols. The combined affect of this convergence will be profound – network security will become exceedingly more complex while a network failure (due to a security attack or severe traffic spike) will have a much greater financial impact on business processes, productivity, and revenue generation.
- **The explosion of network applications.** It is safe to say that most modern applications development is designed to take advantage of pervasive Internet connectivity regardless of whether it is intended for consumer, commercial or government users. Again, this creates a nightmare scenario for security testing. Application and network protocols must be thoroughly tested through a near-infinite set of “what if” scenarios.

The threats described herein open IT assets and software to more onerous and insidious threats each day. There is no longer any option; technology must be thoroughly tested against every conceivable threat. As Michael Howard, one of the pioneers of the Microsoft Secure Development Lifecycle (SDL) states, “Never underestimate your enemy – if you don’t test your code, somebody else will do it for you and with harsher consequences.”²

Black Box Testing For Security

Software test engineers often refer to the dichotomy between “white box” and “black box” testing. According to webopedia (www.webopedia.com), white box testing is defined as a software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Black box testing falls on the other side of the spectrum where testing assumes no knowledge of the software source code or internal structure. Black box testing often incorporates the concept of “fuzz testing,” a methodology where malformed data is consumed by the software under test to see how it reacts.

Black box testing has proven to be particularly effective when testing for software exposure and system failure risks because it:

- **Has broad application.** Since black box testing is independent of individual development projects, it can be applied in numerous use cases. For example, black box testing can be used to analyze the behavior of file formats, networking and application protocols, APIs, etc. In theory, black box testing tools can be used in all of these circumstances, eliminating the need for multiple testing tools, custom scripts, and deep source code knowledge.
- **Provides “wide and deep” coverage.** Leading black box testing tools support a wide range of protocols from Layer 2 through Layer 7 of the OSI stack. This in itself is valuable, but many black box tools offer extensive test suites to exercise all aspects of these protocols. Many users have found problems in secondary esoteric protocols that impact overall software security, performance, and availability. With this “wide and deep” capability, black box tools can test each and every protocol, not just the obvious ones.
- **Eliminates the need for source code and protocol knowledge.** Most test engineers would agree that more testing is always better, but many organizations don’t have the resources, time, or skills to develop hundreds of new scripts to test software on their own. Black box testing tools eliminate this restriction by aggregating testing expertise, fuzzing, and a multitude of test routines into a turnkey solution. With this type of coverage, software engineers can spend their time testing code rather than studying protocols or developing their own test scripts.

In summary, black box testing is a simple matter of doing more with less. In this case, black box testing provides more test suites, protocol support, and fuzzing capabilities while decreasing the need for deep protocol knowledge, test suite development, and source code expertise.

The Case For Codenomicon

The concept of black box and fuzz testing is not new. It was originally developed in the early 1990s to find reliability bugs in software and is now used to find security vulnerabilities as well. From 1990 through 2000, many black box test suites were developed at academic institutions and were freely available for download and use. Predictably, many black box test suites evolved into open source tools. In spite of the fact that these tools were offered as freeware, their use remained limited. Few commercial developers knew these tools existed and even fewer felt that black box testing was necessary. Security testing was still viewed as a domain for software engineering students and researchers, not technology vendors.

As black box testing gained acceptance, problems in the freeware model quickly appeared. Even with open source and freeware tools as a base, test engineers still had to modify the tools for their own use cases. Often times, each open source tool had its own unique interface so actually testing was throttled by a tool-by-tool learning curve. Many tools proved to be flawed themselves or provided very limited functionality. As Internet use grew, many test engineers found that available black box tools often lacked adequate protocol support. When testing a firewall for example, black box tools could test pedestrian protocols like IPv4, FTP, and ICMP, but

couldn't test remote access or application protocols. These limitations also held back black box tools proliferation.

In 2001, black box testing went from an academic exercise to a commercial market with the formation of Codenomicon. The company based its product on the PROTOS test tools research at the Oulu University Secure Programming Group in Oulu, Finland. Since its initial inception, Codenomicon extended its early work in black box testing to develop its DEFENSICS platform, which the company refers to as "the preemptive security test platform." Historically, Codenomicon sold its testing products to telecommunications carriers and network service providers. However, the company has seen recent and growing adoption with enterprise customers in other industries, such as financial services, automotive and government.

Will black box testing continue to proliferate? If so, what value does Codenomicon bring to its customers? ESG had the opportunity to interview several Codenomicon customers recently to get a better understanding of black box testing while exploring the answers to these questions.

Black Box Testing Improves Quality and Efficiency

Before delving into Codenomicon and its product capabilities, ESG had a more basic question: What value does black box testing really deliver? The answer can be summarized in a single word – plenty. Several users commented that black box testing helped them extend the scope of their software testing into new areas:

"There is so much going on in the operating system in terms of protocol support and we wanted to make sure that we understood how these protocols impacted the security and robustness of our software. Black box testing gave us a much broader use case." (Software Company)

"Originally, we found some unstable behavior with some protocols when using Nessus for testing purposes. We adopted black box tools soon afterward, because we wanted to exercise more protocols through a thorough set of test cases. Black box tools certainly fit this requirement." (Telecommunications Equipment Company)

Users also liked the fact that black box testing provided a turnkey solution. This was deemed as superior to developing internal knowledge or homegrown tools:

"We needed protocol specific tests and didn't have the resources to develop our own test for CIFS, RTP, SSL and lots of others. Black box testing gave us support for every protocol we needed." (Software Company).

Another rationale for black box tools was driven by business requirements – customers demanded that their vendors use them.

"Why did we adopt black box tools? That's easy – our customers told us to. We are finding this true of more and more of our carrier customers." (Telecommunications Equipment Company).

Why Codenomicon?

With the benefits of black box testing clearly established, users still had plenty of choices. They could use open source tools, build their own black box testing from the ground up, or select a commercial product. Each of the large global organizations selected Codenomicon. Why? For starters, Codenomicon seems to have a good reputation and wide visibility with software test engineers. Respondents often said that they were familiar with Codenomicon before they purchased it. Some admitted that they had previous experience with Codenomicon making it a logical fit when black box solution requirements arose.

"Some of our team members had worked with Codenomicon before and suggested that we try it. I can't tell you how important these types of personal references are when you are about to purchase and use something you have little experience with." (Software Company).

“We had lots of different black box tools in use around the company and we decided we needed one standard that we used everywhere. Codenomicon was being used in some of our European facilities so it came highly recommended to us. After some initial evaluation, we made it the company standard and now have one uniform process.” (Telecommunications Equipment Company)

As mentioned earlier, protocol testing is a major reason for the growing popularity of black box testing. Users say that this is an area of particular strength for Codenomicon. Out of the box, Codenomicon supports more than 140 interfaces from Layer 2 through Layer 7, providing a distinct advantage over open source tools and competitive offerings. In addition, price and value came into play as well. While the competition tends to charge an additional fee for individual protocol testing, Codenomicon provides this broad protocol support as part of its product license. To its customers, DEFENSICS provides a marked economic advantage over the competition.

“We really thought we would use open source tools, but the more protocols we decided to test, the more work we had to do. Codenomicon supports almost every protocol we wanted to test, so it made economic sense to purchase DEFENSICS rather than spend dedicated time and resources toward customizing open source.” (Software Company)

“We always look for opportunities to automate internal processes or replace them with services. Codenomicon fit the bill.” (Telecommunications Equipment Company)

“Our decision came down to Codenomicon or an appliance-based solution. The appliance was a high quality product, but the licensing terms were onerous to say the least. Codenomicon provided equal technical capabilities at a much lower price. How can you pass on that combination?” (Telecommunications Equipment Company)

“The most important thing is this: Codenomicon’s protocol depth and state is the best in the market. The documentation is also awesome; you can see every test case.” (Telecommunications Equipment Company)

Unlike some competitors, Codenomicon DEFENSICS is offered as a software-based system rather than packaged as an appliance. Customers’ found this seemingly minor distinction is actually an important differentiator. Essentially, a software-only solution could benefit from existing low-cost high-performance, off-the-shelf PC hardware, as well as be easily deployed company-wide or shared through a flexible floating license scheme.

“We like the idea of a software-based system. We can easily modify the system with new operating systems, more network cards, or more complex configurations. We are also able to select preferred hardware vendors that have established good SLAs.” (Telecommunications Equipment Company)

“Since Codenomicon is a software-based system, we were able to purchase DEFENSICS using our operating rather than capital budget. This made the decision easy.” (Telecommunications Equipment Company).

Protocol support, pricing advantages, and a software-based system were important product differentiators. Customers were also impressed by Codenomicon corporate intangibles, such as its experience and customer support.

“This is an extremely unique area where skills are pretty rare. Codenomicon really knows its stuff.” (Software Company).

“Codenomicon is a stand-up company. The technical team was very helpful through our proof-of-concept, and the sales team is top notch. The company is really easy to do business with.” (Telecommunications Equipment Company).

The Benefits of Black Box Testing and Codenomicon DEFENSICS

Codenomicon customers went through a logical progression. They recognized a use case for black box testing, experimented with freeware and open source tools, and then finally purchased and implemented Codenomicon DEFENSICS. The consistent theory through this sequence of events was that each phase would result in software with increasingly higher quality and security. Had these users achieved these results? While the sample size was relatively small, ESG has to conclude that the answer is unanimous – Yes.

Users claim that they were able to streamline their testing by introducing Codenomicon and black box testing earlier in the testing process. This helped companies accelerate and improve overall testing.

“Sometimes we did catch protocol problems with our standard tools, but we had no visibility into what the problem was or how to fix it. With Codenomicon, we can see the test cases and read the failure report. What this means in practical terms is that we are able to find and fix problems in a lot less time.”
(Software Company)

“Based upon our models, we know that if we catch a software problem during the problem verification phase, it takes a lot less time to fix. Codenomicon enabled us to do this.” (Telecommunications Equipment Company)

This ultimate benefit is easily identified. Codenomicon and black box testing helped companies improve software quality and lowered the cost associated with finding and fixing software bugs.

“Once we got comfortable with Codenomicon, we were able to pull black box testing into the build process. By doing this, we could show the development team more specific information on where the system was failing. This helped them understand and address issues they hadn’t considered before. Ultimately, this accelerated and improved our whole testing process.” (Software Company)

“Based upon our models, we know that if customers find software bugs, it cost about \$32,000 to fix. If we catch a bug during our software verification phase, it costs between \$4,000 and \$8,000 to fix. If we find software bugs in development, it only costs around \$600 to \$800 to fix them. Obviously, our goal is to find bugs as early as we can, and Codenomicon is helping us do this.” (Telecommunications Equipment Company)

“Believe it or not, we used the IPv4 test cases and found problems with a number of our routers. You would think that routers would be stable by now, but as I mentioned we found problems. This demonstrates the value with Codenomicon; it finds problems in places where you might not even look.”
(Telecommunications Equipment Company)

The Bottom Line

When it comes to security, too often conversations stray quickly to common technology safeguards like antivirus software or perimeter firewalls. Over the past few years however, the technology industry has reached a new epiphany. Developers now realize that if they improve the quality of their code and dedicate testing cycles and resources to security, they can produce more robust and secure software. In addition, better quality software results in lower maintenance costs. It is cheaper to find and fix software problems in development and test than it is to react with emergency patches to a new vulnerability discovered in the field. Given these positive benefits, industry leaders like EMC, Microsoft, and Oracle have actually altered their software development in order to inject more security focus into the process.

In addition to developer training, new tools, design changes, and enhanced testing, many firms have added black box testing (aka: “fuzzing,” “negative testing,” etc.) into their development processes leading to positive results. Based on empirical observation, ESG believes that black box testing has an addictive quality to it. Companies that experiment with black box testing ultimately want to use it across more tests in more areas over time. As this occurs, software test engineers consistently run out of headroom with freeware and open source. Rather than invest in skills and build on top of open source, many decide that commercial solutions make more sense.

For firms that go through this black box testing evolution, Codenomicon seems to be an extremely logical step. Users consistently appreciated the depth, breadth, and design of the product. More importantly, users indicate that Codenomicon helped them improve software quality, accelerate software testing cycles, and lower the costs associated with software testing and maintenance. In ESG's view, this may be the ultimate proof point of value.



20 Asylum Street
Milford, MA 01757
Tel: 508-482-0188
Fax: 508-482-0218

www.enterprisestrategygroup.com