



Validation Testing Report

September 2004

Test No. ITS04001



ForeScout Technologies

CounterACT 3.1.2



600 Community Drive, Manhasset, NY 11030
Tel: (516) 562-5000 • Fax: (516) 562-5500
Email: info@itslabs.com • URL: www.itslabs.com
© 2004 CMP Media LLC

POWERED BY
Network
Computing
Labs



TABLE OF CONTENTS 2

EXECUTIVE SUMMARY 3

 PRODUCT OVERVIEW3

 INDEPENDENT VALIDATION CLAIMS3

 TESTING OBJECTIVES3

 RESULTS SUMMARY3

TESTING DESIGN 4

 TEST METHODOLOGY4

 TEST TOPOLOGY4

 LOAD CONDITIONS5

 TRAFFIC VERIFICATION5

VALIDATION TEST 1: NATIVE WORM ATTACK 6

 TEST DESCRIPTION6

 BASELINE RESULTS6

 VALIDATION TEST RESULTS7

VALIDATION TEST 2: ZERO DAY CGI WORM ATTACK 8

 TEST DESCRIPTION8

 BASELINE RESULTS8

 VALIDATION TEST RESULTS9

VALIDATION TEST 3: ZERO DAY TCP BACK DOOR ATTACK . . . 10

 TEST DESCRIPTION10

 BASELINE RESULTS10

 VALIDATION TEST RESULTS11

FINAL THOUGHTS 12

 TEST PROFILE12

 TEST INFRASTRUCTURE12

 ABOUT US12

Product Overview

As the time between exploit and worm release continues to decrease, the capability to mitigate “zero day” or unknown worms becomes critical to network administrators. To address this growing requirement, ForeScout Technologies

introduced CounterACT (formally WormScout), which is designed specifically to automatically mitigate both known and unknown worms from propagating across the network.

Independent Validation Claims

During July and August 2004, Independent Testing Services powered by Network Computing Labs™ (ITS) was contracted by ForeScout Technologies to independently validate specific capabilities of the CounterACT product. The following claims were subject to open and independent testing verification:

1. CounterACT identifies worm-infected computers, contains their activity and suppresses them from infecting other network segments.
2. CounterACT identifies and provides mitigation for both known and unknown “zero day” network-based worms.

Testing Objectives

ITS tested ForeScout Technologies’ CounterACT in an isolated test environment designed to duplicate a real-world deployment. The network was populated with 33 vulnerable hosts, then attacked with the following worms:

- **Blaster.Worm** – Several variants of the W32.Blaster.Worm / Win32.Poza.Worm captured from the wild.
- **New, unknown CGI attack** – In-house-developed worm based on exploiting a new and unknown vulnerability in

Microsoft Windows and ITS created specifically for this testing.
• **New, unknown TCP back door** – In-house-developed worm that exploits a custom back door based on TCP port 7771.

The objective of the testing was to demonstrate the capability of CounterACT to isolate and mitigate both known and unknown worms from infecting vulnerable hosts across the network.

Results Summary

CounterACT uses a suite of TCP mechanisms, including the use of host spoofing, TCP session stalling and TCP resets, first to identify the worm and then to block the infected host from communicating to other network hosts. The results of the testing indicated that CounterACT provided an effective mitigation against both known and unknown worms. Tests consisted of a real-world worm (Blaster) and two ITS Labs in-house-developed worms. The in-house worms exploited custom-developed vulnerabilities in Windows 2000 and IIS.

Native Worm Attack: CounterACT completely blocked

several variants of the Blaster.Worm from infecting any vulnerable hosts in the network.

Zero Day HTTP/CGI Worm Attack: CounterACT completely blocked a custom, in-house-developed, “zero day” CGI-based worm.

Zero Day TCP Back Door Attack: CounterACT mitigated a custom, in-house-developed, very aggressive and intelligent worm from infecting all but a few hosts in the network. CounterACT successfully contained the infection so that the vast majority of the network remained unaffected.

Test Methodology

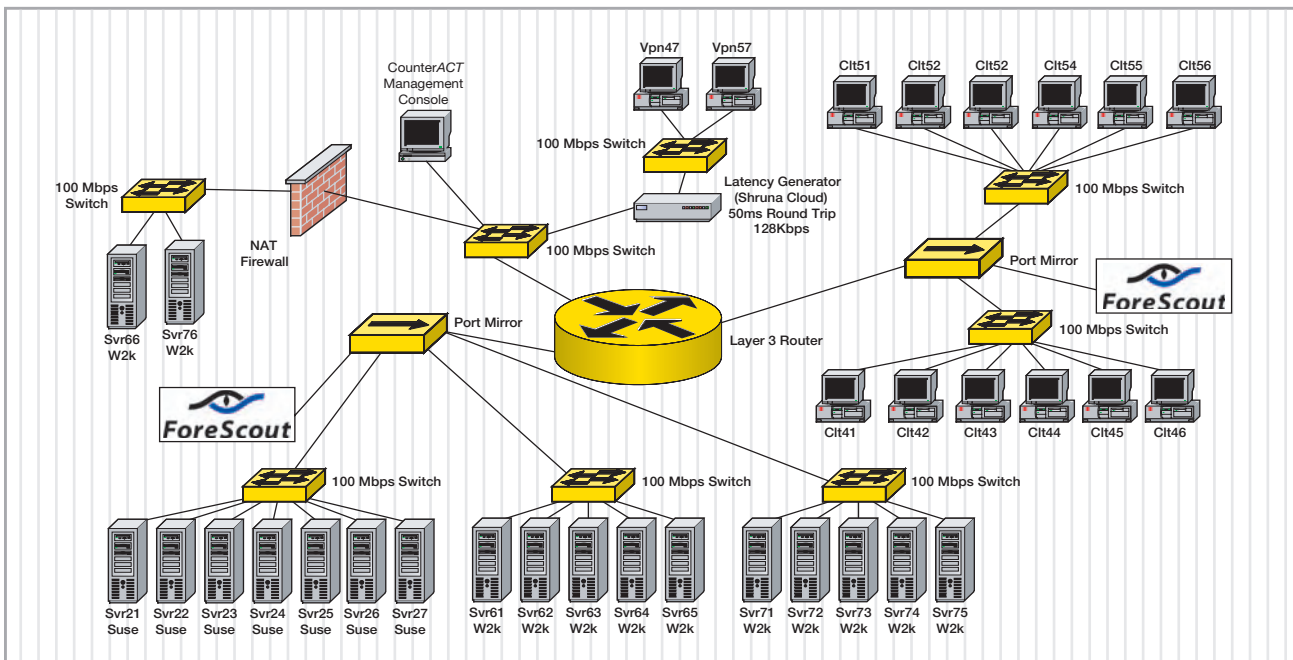
Each test consisted of the following phases:

1. Clear the CounterACT database using the Data Reset option.
2. Reset each host to OS base installation defaults.
3. For custom in-house-developed worm testing, execute vulnerability code on each Win32-based host.
4. Launch a Web browser from each client to a single server. This enables CounterACT to learn the topology of valid hosts. Production implementations exhibit end-user traffic that has the same effect.
5. Begin file copies from server to clients for background traffic.
6. Copy worm via floppy disk to source host.
7. Activate worm by running the .exe via command script and then monitor results.

Test Topology

The goal of the test environment was to mimic conditions in existing enterprises with edge layer switches, distribution layer switches, core routers and NAT firewalls. The test topology consisted of two protected networks (client and server subnets), backbone network, remote access network (VPN) and a NAT-based firewall to a simulated Internet.

CounterACT requires a port mirror or tap to passively monitor traffic in the network. The test topology used port monitoring at the distribution layer, which is indicative of enterprise network monitoring deployments. Individual implementations may vary by topology, i.e., edge layer monitoring and VPN layer monitoring.



Load Conditions

Background traffic was composed of the following:

- Scripted file transfers from the servers to the clients. A script was executed on a subset of the clients that copied the files contained in c:\winnt continuously from the servers. Resulting throughput ranged from 20 Mbps to 30 Mbps. The continuous file transfers were driven by a .bat command script that copied c:\winnt from a remote server, deleted the local copy and infinitely looped back to the copy operation.
- Periodic Web browsing from clients to randomly selected servers to ensure the services were still available was provided by randomly selecting a client workstation every 30 seconds and browsing with Internet Explorer.

Traffic Verification

Passive monitoring was achieved through the use of a network capture appliance connected to each port mirror. The network capture was used to independently validate traffic patterns and TCP transactions. Non-load traffic during each test was captured to a network trace file and then post-analyzed using multiple packet decoding tools such as Ethereal and Netmon.



Validation Test 1: Native Worm Attack

Test Description

Performance Claim Subject to Validation:

CounterACT identifies worm-infected computers, contains their activity and suppresses them from infecting other network segments.

Validation Test Methodology:

To test worm mitigation, the test environment was infected directly by Blaster.Worm through execution of the worm binary on a vulnerable host. Several variants of Blaster.Worm were utilized in the testing:

- Win32.Lovesan.a / Win32.Poza.A
- Win32.Lovesan.b / Win32.Poza.B

Blaster exploits a security vulnerability in Microsoft RPC (MS03-026) and then utilizes TFTP to transfer itself to a compromised host. Additional details on Blaster and its variants can be found at:

www.microsoft.com/security/incident/blast.msp/vil.mcafeesecurity.com/vil/content/v_100547.htm
vil.mcafeesecurity.com/vil/content/v_100551.htm

Baseline Results

The baseline was designed to measure the performance of the worm with no protection in the environment. The worm was copied to a single vulnerable host via floppy disk and then executed via command line. Once activated, the worm immediately began to search for other hosts across the network to infect.

The Blaster worm targets hosts via both random addressing and addresses based on the subnet to which the infected host belongs. Because of the addressing used in the test topology, the worm was only partially successful in infecting the environment.

Blaster was able to infect hosts belonging to the same local class C network in the 24 hours allotted; however, because

of the random nature of the worm's scanning algorithm, it was unable to infect the entire test topology. Test results, collected across several tests from multiple sources and worms, can be seen in *Native Worm Attack: Baseline*.

Native Worm Attack: Baseline

Source Host	Infected Nodes
Clt41	9
Vpn47	1
Srv61	6

Client 41 was able to infect several of its neighbors; however, the randomized scanning was unable to locate the server subnet from the client subnet. Similar results were exhibited from VPN 47 and Server 61. (Note: Infected node quantities do not include the source node.)

While not as efficient as we would have liked, the baseline for Blaster does demonstrate the ability of the worm to infect multiple hosts via the network.

Validation Test Results

During the validation test stage, CounterACT was fully enabled to block the worms from infecting the network. The same procedures used in the baseline were then repeated and verified.

Packet-level inspection from the network monitor indicated that CounterACT identified and began mitigation in three to five seconds from the initial time of infestation.

In all test cases, the CounterACT product mitigated the worm infestation by protecting the client and server subnets from further infection. The only case where Blaster was able to infect another node was from VPN 47 to VPN 57, which are both unprotected hosts sharing the same network segment (local and unprotected).

Infection counts were measured during the six attempts (individual test runs) to infect the network. Test results can be seen in *Native Worm Attack: Validation*.

Native Worm Attack: Validation

Source Host	Infected Nodes
Clt41	0
Vpn47	1
Srv61	0
Clt42	0
Clt44	0
Srv64	0

CounterACT also successfully identified the worm as Blaster from its internal database during each test.

To ensure that CounterACT could fend off the attack and continue to mitigate against further Blaster infections, the infected hosts were left online for a period of 72 hours. During this 72-hour period, CounterACT continued to provide protection, and no other hosts were compromised as the worms continuously probed the network.



Test Description

Performance Claim Subject to Validation:

CounterACT identifies and provides mitigation for both known and unknown “zero day” network-based worms.

Validation Test Methodology:

To test “zero day” worm mitigation, the environment was infected by an ITS Labs in-house-developed HTTP/CGI-based worm. A server-side CGI script also was developed to accept file binaries through an HTTP POST and then to execute the file/worm once received.

The custom worm and vulnerability code has no registered signature, and ForeScout Technologies was never granted access to the binary or the source code for this worm. The worm code and crafted vulnerability have never been disclosed outside the isolated ITS Lab environment. This ensured that the worm was a completely new “zero day” worm and would evade traditional signature-based detection methods.

The HTTP POST behaves like a traditional user, uploading a file to a simple Windows 2000 IIS Web site.

Baseline Results

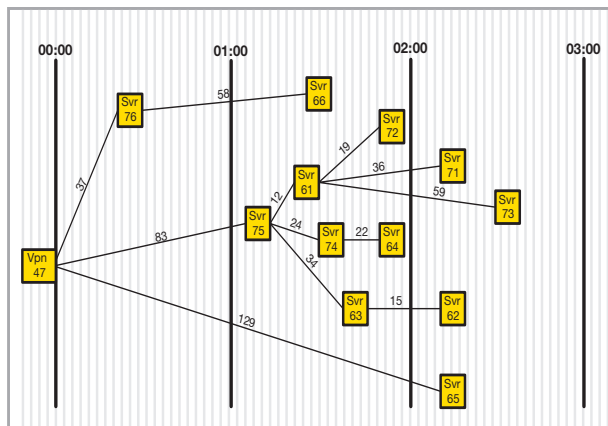
The baseline was designed to measure the performance of the worm with no protection in the environment. The worm was copied to a vulnerable host via floppy disk and then activated via command line. Once activated, the worm immediately began to search for other hosts across the network to infect.

Traditionally, worms exhibit near-random behavior when scanning IP addresses based on portions of class A address space. However, to ensure this worm would be highly effective (hence simulating an intelligent unknown worm), the worm was tuned with the IP addressing of the topology such that approximately two percent of the scans would resolve to vulnerable hosts. The remaining 98 percent of the scans were completely within the routing topology of the environment. Thus, the worm did not waste any time searching outside the five class C networks used in the test environment.

The worm was able to scan for vulnerable hosts at a rate of five host attempts per second. Once a vulnerable host was discovered, the custom worm would execute an HTTP POST of itself to the target host. In *Infection Rate: Baseline*, the worm was

mapped as it progressed from host to host. Each line segment represents the time required for an infected host to find a vulnerable host and self propagate through the HTTP POST. Each box represents a vulnerable host and its time of infestation.

Infection Rate: Baseline



For example, the initial infection point was node VPN 47. Thirty-seven seconds into the test, VPN 47 infected Server 76. Fifty-eight seconds later, Server 76 infected Server 66. At time index 00:00 the worm was activated on a remote client (VPN 47), thus simulating an infected VPN user

connecting to the network. The VPN host was across a 50-ms 128-Kbps link (provided by Shunra/Cloud); thus discovery and infection were somewhat delayed by the constrained link. Once the worm infected a few of the servers, propagation was very fast, with all servers infected in less than three minutes.

Validation Test Results

During the validation test stage, CounterACT was fully enabled to block the worms from infecting the network. The same procedures used in the baseline were then repeated and verified.

In all test cases, CounterACT successfully mitigated the worm infestation by protecting the client and server subnets from further infection. CounterACT's use of TCP stalling and spoofing had the positive effect of locking up the CGI worm so that scanning would stop for several minutes. The custom worm also exhibited a series of socket errors caused by CounterACT's TCP blocking mechanisms. These blocking mechanisms

include TCP stalling and IP host spoofing, which effectively exhaust operating-system-level socket resources on the infected host, thereby impeding further worm infestation.

Test results can be seen in *Infection Rate: Validation*.

Infection Rate: Validation

Source Host	Infected Nodes
Clt41	0
Vpn47	0
Srv65	0

Packet-level inspection from the network monitor indicated that CounterACT identified and began mitigation in five to eight seconds from the initial time of infestation. The network capture trace clearly showed the worm connecting to spoofed IP addresses created by CounterACT. CounterACT would then stall the TCP connection, thus using up socket resources within the worm.

Validation Test 3: Zero Day TCP Back Door Attack

Baseline Results continued

Traffic rates and infestation order were monitored via network capture and validated by a log file written to local disk by the worm.

At time index 00:00 the worm was activated on a remote client (VPN 47), thus simulating an infected VPN user connecting to

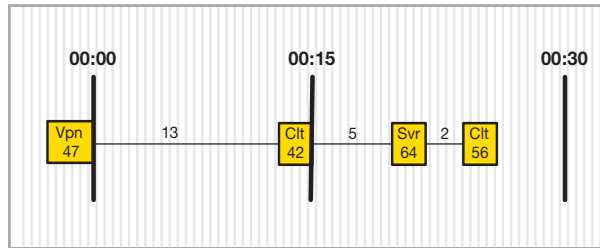
the network. The VPN host was across a 50-ms 128-Kbps link (provided by Shunra/Cloud); thus discovery and infection were somewhat delayed by the constrained link. Once the worm infected a few of the local hosts, propagation was very fast, with all hosts infected in less than 30 seconds.

Validation Test Results

During the validation test stage, CounterACT was fully enabled to block the worms from infecting the network. The same procedures used in the baseline were then repeated and verified.

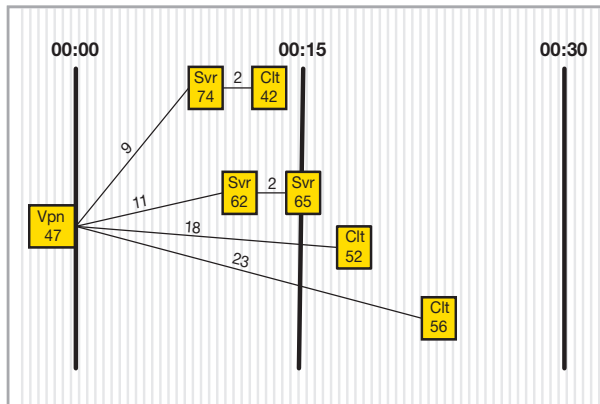
Testing of the “zero day” TCP worm validated the capability of CounterACT to mitigate a worm from infecting the entire network. Analysis of the resulting network capture trace demonstrated that the TCP worm could scan 10 hosts per second and infect a host in 60 ms; thus the worm was successful in infecting a few nodes during each test. However, once CounterACT identified the worm’s characteristics, it successfully locked down the infection to a limited subset of nodes in the network.

TCP Back Door: Validation 1



CounterACT demonstrated a capability to quickly mitigate infected hosts, for which the network traffic of the initial infection was unmonitored—a rather unexpected and quite positive result. For example, in the second test (documented at left), Server 62 and Server 65 are on the same edge switch and both were unable to infect any other nodes.

TCP Back Door: Validation 2



VPN 47 was able to infect a few hosts before CounterACT was able to fully employ its TCP blocking capabilities across the WAN.

Packet-level inspection from the network monitor indicated that CounterACT identified and began mitigation in two seconds from the initial time of infestation.



Validation Testing Report: ForeScout Technologies

Final Thoughts

Test Profile

Vendor:  ForeScout Technologies

ForeScout Technologies offered CounterACT (formally WormScout) as a software Package to be installed on a customer-provided Intel Server:

- Intel Dual Processor P4 Xeon 2.8 GHz, Intel Motherboard, 2 GB RAM, Wide SCSI HD, Intel Pro1000 GE NIC
- ForeScout now offers CounterACT as an appliance, available in 100-Mbps and 1-Gbps configurations.

Product: CounterACT 3.1.2

Test Window: July 2004 – August 2004

Price: Starts at \$11,995

Report Number: ITS04001

Report Date: September 7, 2004

Vendor Contact: www.forescout.com,
(866) 377-8771, (408) 213-3191

Testing Infrastructure:



Product: Shunra/Cloud • **URL:** www.shunra.com

Servers: 6 Windows 2000 unpatched servers per VMWare GSX Dual processor physical server.

Clients: 7 Windows 2000 unpatched clients per VMWare GSX Dual processor physical server.

Layer 3 Router: Lucent Cajun P550

Distribution Layer: HP 100-Mbps hub, which enables effective 100-Mbps port mirroring on all ports. All testing less than 30 Mbps, thus choice of hub or switch with port mirror has no effect on traffic conditions.

Edge Layer: VMWare GSX switch or 3Com Superstack for remote and VPN subnets.

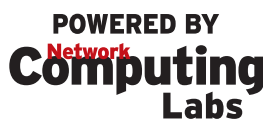
About Us

For well over a decade, Network Computing has set the standard for real-world analysis of enterprise-class products, technologies and services. Creating unbiased testing methodologies and defining what matters most have been the cornerstones of our Labs.

Independent Testing Services powered by Network Computing Labs™ (ITS) offers a credible and well-respected

third-party validation of vendors' product feature claims. The expert practitioners at ITS draw upon their vast experience and expertise to craft test plans with the same care and attention to real-world relevance as found in the pages of *Network Computing Magazine*.

For more information about our testing services, please visit us at www.itslabs.com. If you're interested in testing validation services, please contact us at info@itslabs.com.



INDEPENDENT TESTING SERVICES, POWERED BY NETWORK COMPUTING LABS™ (ITS) prepared this report under contract for ForeScout Technologies, Inc. (Client). While the services were performed in a professional manner and the data and information contained in the report were carefully researched and believed to be accurate, such services were performed and the report provided based on products and information supplied by Client to ITS, and ITS makes no representations as to its accuracy, adequacy or completeness. ITS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE CONTENTS OF THIS REPORT OR THE RESULTS TO BE OBTAINED BY ANY PERSON OR ENTITY FROM USE THEREOF. ITS FURTHER SPECIFICALLY DISCLAIMS ALL WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO ANY PRODUCT MENTIONED IN THIS REPORT. All trademarks are the property of their respective owners.