

# [Full-Disclosure] EEYE: Symantec Multiple Firewall NBNS Response Processing Stack Overflow

**Source:** <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2004-05/0639.html>

---

**From:** Marc Maiffret ([mmaiffret\\_at\\_eeeye.com](mailto:mmmaiffret_at_eeeye.com))

**Date:** 05/13/04

To: <[full-disclosure@lists.netsys.com](mailto:full-disclosure@lists.netsys.com)>

Date: Wed, 12 May 2004 17:01:39 -0700

Symantec Multiple Firewall NBNS Response Processing Stack Overflow

Release Date:

May 12, 2004

Date Reported:

April 19, 2004

Severity:

High (Remote Kernel Code Execution)

Vendor:

Symantec

Systems Affected:

Symantec Norton Internet Security 2002

Symantec Norton Internet Security 2003

Symantec Norton Internet Security 2004

Symantec Norton Internet Security Professional 2002 Symantec Norton

Internet Security Professional 2003 Symantec Norton Internet Security

Professional 2004 Symantec Norton Personal Firewall 2002 Symantec Norton

Personal Firewall 2003 Symantec Norton Personal Firewall 2004 Symantec

Client Firewall 5.01, 5.1.1 Symantec Client Security 1.0, 1.1, 2.0(SCF

7.1) Symantec Norton AntiSpam 2004

Description:

eEye Digital Security has discovered a critical vulnerability in the Symantec firewall product line that would allow a remote, anonymous attacker to execute arbitrary code on a system running an affected version of the product. By sending a single specially-crafted NetBIOS Name Service (UDP port 137) packet to a vulnerable host, an attacker could cause an arbitrary memory location to be overwritten with data he or she controls, leading to the execution of attacker-supplied code with

kernel privileges and the absolute compromise of the target.

The vulnerability exists due to a flaw in the way these products process incoming UDP packets with a source port of 137 (NetBIOS Name Service). If such a packet is received, it is validated as a proper NBNS packet and certain information from the packet is stored. A specifically crafted packet can cause the code that copies information out of the packet to instead write packet data to an arbitrary memory location, a flaw that can be leveraged in order to maliciously execute on an affected system. In order for this vulnerability to be exploitable, the firewall must be configured to allow incoming UDP/137 packets, a setting which is not present by default, but may be enabled by the user or network administrator in order to facilitate Windows file sharing.

#### Technical Description:

The SYMDNS.SYS driver included in the Symantec firewall product line validates DNS and NetBIOS Name Service responses before allowing them through the firewall. As it turns out, the handlers for both types of packets have grave security issues, but this advisory focuses on NBNS packets and leaves DNS up to Barnes and Karl. The intended protocol is determined by the source port of the UDP packet — 53 for DNS, 137 for NBNS — and after verifying that the incoming packet is marked as a response according to the header, it is passed off to the appropriate analysis routine, both of which perform similar but protocol-specific processing on the answer data contained therein (although no further validation takes place).

In the case of the NBNS routine, the questions in the packet are skipped, and the answers are only examined if they have Class 01h (INET) and Type 01h (A) or 20h (NB). For answers meeting these criteria, the name is first-level decoded, the IP addresses are stored in a list, and both are later recorded internally in a global array. (As a refresher: first level encoding represents each byte of a name as two letters from 'A' to 'P', which correspond to the high and low hexadecimal digits of the byte's value — 'A' is 0, 'B' is 1, 'C' is 2, and so on. For example, "eEye" is represented in hexadecimal as 65h 45h 79h 65h, and is therefore encoded as "GFEFHJGF". See RFC 1001, Section 14.1, for more information.)

The first of many problems that make this vulnerability possible is that the first-level decoding routine will decode an amount of data corresponding to the length byte preceding the encoded name, making it possible to store up to 127 arbitrary bytes (plus a null terminator) into a 32-byte stack buffer provided by the main NBNS processing routine. Although this condition is insufficient to overwrite the return address directly (the buffer begins at EBP-118h, but only an 80h-byte write is possible), there is an index variable that can be overwritten in order to manipulate the IP address copying loop later in the function. The NBNS processing routine's stack frame can be represented as follows:

```
PBYTE var_11C;  
char var_118[0x20];  
DWORD var_F8;  
DWORD var_F4;  
DWORD var_F0;  
PBYTE var_EC;  
DWORD var_E8[0x18];  
char var_88[0x80];  
PBYTE var_8;  
PBYTE var_4;  
(saved EBP at EBP+0)  
(saved EIP at EBP+4)  
...
```

var\_118 is the destination buffer passed to the first-level decode routine, and just about everything after it is initialized after the decoding overwrite occurs, or is otherwise useless: var\_E8/var\_88 is memset to 0; var\_EC and var\_F0 get wiped out; var\_F4 is just an outer loop counter (infinite loop: DoS); and var\_8 and var\_4 aren't even reachable. The exception here is var\_F8, which is initialized to 0 at the beginning of the function, used to index into a stack array (var\_E8), and is not checked for any out-of-bounds values other than the exact size of the array in elements (18h). The fact that the variable is located immediately after the overflowable buffer just adds to the convenience.

Once the answer name has been decoded, the NBNS processing routine enters another loop to copy IP addresses from the response into var\_E8. Since the contents of the list are supposed to be accumulated from across all answers in the packet, var\_F8 is not reinitialized when the loop begins, and furthermore, the terminating condition of the loop is only that var\_F8 equals 18h (no greater-than). As a result, once the variable has been overwritten with a sufficiently high value, "IP addresses" within the packet will be written onto the stack at [EBP-E8h+(var\_F8\*4)] until the answer's data length has been exhausted (up to roughly 64KB).

Because the length of the first-level encoded name must be at least 40h in order to touch var\_F8, the routine that skips a length-prefixed name component will mistake the length byte for a compressed name pointer, and will only advance by two bytes instead of (length of name + 1). This means the data that normally follows the encoded name actually begins "inside" the name, but this doesn't matter because the first-level decoding routine does not validate that the name consists only of characters from 'A' to 'P'. Additionally, it does not check for compressed name pointers and will happily accept any value for the length byte. The result of this stack buffer overflow / consistent lack of validation combo is another UDP remote kernel vulnerability.

Protection:

Retina Network Security Scanner has been updated to identify this

vulnerability.

Vendor Status:

Symantec has released a patch for this vulnerability. The patch is available via the Symantec LiveUpdate service. For more information please refer to the Symantec security advisory.

<http://securityresponse.symantec.com/avcenter/security/Content/2004.05.12.html>

Credit:

Discovery: Derek Soeder

Related Links:

Retina Network Security Scanner – Free 15 Day Trial

<http://www.eeye.com/html/Products/Retina/download.html>

Greetings:

damodadudewn; CMC, BG, Jenna, Lan, BP, JKP, Daryl, RLS, KT, NV; Brett Moore; Riley; and Colleen R. (thanks anyways for the offers!)

Copyright (c) 1998–2004 eEye Digital Security Permission is hereby granted for the redistribution of this alert electronically. It is not to be edited in any way without express consent of eEye. If you wish to reprint the whole or any part of this alert in any other medium excluding electronic medium, please email [alert@eEye.com](mailto:alert@eEye.com) for permission.

Disclaimer

The information within this paper may change without notice. Use of this information constitutes acceptance for use in an AS IS condition. There are no warranties, implied or express, with regard to this information. In no event shall the author be liable for any direct or indirect damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.

Feedback

Please send suggestions, updates, and comments to:

eEye Digital Security

<http://www.eEye.com>

[info@eEye.com](mailto:info@eEye.com)

---

Full-Disclosure – We believe in it.

Charter: <http://lists.netsys.com/full-disclosure-charter.html>

---

- application/ms-tnef attachment: [winmail.dat](#)