

# [NEWS] IGMP Denial of Service Vulnerability

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-06/0053.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 06/15/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: Sat, 15 Jun 2002 21:28:20 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

-----

IGMP Denial of Service Vulnerability

---

## SUMMARY

IGMP, in a bid to optimize its functionality uses a report suppression mechanism to prevent redundant IGMP member reports from being sent to the quicker router. We refer the reader to RFC 2236 for a better understanding of the IGMP protocol. The router on seeing the first membership report for a group from a host in the subnet its serving works towards extending the distribution tree for a Multicast group G all the way to itself. This report suppression mechanism can be exploited as explained below.

## DETAILS

### Attack:

We consider different scenarios in which such an attack can be launched. In these scenarios, we tested each of the following operating systems for vulnerability to the attacks : Redhat Linux 7.3, Microsoft Windows 98, and Microsoft Windows XP. We found all of the above listed operating systems to be vulnerable.

### Scenario 1

Host H1 and H2 are connected to a router R using a hub. Host H1 is a member of the multicast group 230.0.0.1 and is receiving traffic from the designated router R. R sends out periodic IGMP membership query messages soliciting for membership reports from the hosts in the network it is

## Securiteam: [NEWS] IGMP Denial of Service Vulnerability

servicing. Host H1 in response to the membership query message multicasts a membership report to the same group. Now H2 starts the attack. H2 is sniffing the network for membership query messages from R. On receiving the message, H2 unicasts a membership report to H1. H1 on receiving the message infers that some other host on the network is also interested in receiving traffic and suppresses its reports. Little does H1 know that there is something fishy going on. It would know this if it checked the destination ethernet address on the report it received. The address would have been its own and not a multicast ethernet address.

Effectively, now R doesn't receive any membership reports for the group 230.0.0.1 and hence blocks all traffic related to that group from flowing into the subnet. What has just been described is a denial of service attack against H1. This attack is partially unsuccessful if some other hosts on the network also subscribe to the same multicast group. However, H2 on seeing membership reports from those hosts can unicast the reports to those hosts separately. This would cause all those hosts to suppress their reports.

### Scenario 2

Host H1, Host H2 and other hosts are connected to the router in a switched environment. In such a network, host H2, the attacker receives membership queries and membership reports periodically sent out by other hosts to the group 230.0.0.1. The attacker can then send membership reports directly addressed to the ethernet addresses of each host in the network, again launching a subnet wide denial of service attack.

### Scenario 3

Host H1, Host H2 and other hosts are connected to the router in a switched environment with IGMP snooping enabled. In such an environment the attacker will not hear membership reports sent out by other hosts on the network. To launch a denial of service attack, the attacker will have to methodically send membership reports to each and every host on the network. This would result in the same denial of service attack as detailed in the two scenarios given above.

### Solution:

To counter such an attack, a host on receiving an IGMP packet, should check the MAC address. If it is not a multicast ethernet address i.e. with the prefix 01:00:5E, the host must drop the packet. This fix in the handling of IGMP packets is an effective solution to the denial of service attack.

### Linux Patch

For Linux 2.4.18, we patched the kernel source against the denial of service attack. Here is the patched `igmp.c` file that needs to be compiled into the boot image. The changes made to the source are given here. The `igmp.c` file is located in `/usr/src/linux*/net/ipv4` directory.

Note please that the changes only check if the ethernet address is a valid Multicast/Broadcast address or not (basically checks if the broadcast bit

Securiteam: [NEWS] IGMP Denial of Service Vulnerability

in the MAC address is set). The code does not check for the correctness of the ethernet address i.e. checks if the ethernet address corresponds to the Multicast group IP). Having said that, this change is a sufficient change to evade such attacks. However, we believe that tighter checks should be made on operating systems running both at the host and the router.

The changes should work as is for other kernel versions as well. However, we haven't tested the same against other versions.

ADDITIONAL INFORMATION

The information has been provided by <mailto:krishna@cs.ucsb.edu> Krishna Ramachandran, Arun Qamra, Mohit Sang.

=====

This bulletin is sent to members of the SecuriTeam mailing list. To unsubscribe from the list, send mail with an empty subject line and body to: [list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)  
In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====  
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind. In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.



- *Previous message:* [support@securiteam.com](mailto:support@securiteam.com): "[NT] Microsoft SQL Server 2000 pwdencrypt() Buffer Overflow"
- *Messages sorted by:* [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)