

# [NEWS] Cisco IOS Stack Group Bidding Protocol Crafted Packet DoS

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-01/msg00072.html>

---

- *From:* SecuriTeam <[support@xxxxxxxxxxxxxx](mailto:support@xxxxxxxxxxxxxx)>
  - *Date:* 19 Jan 2006 16:54:27 +0200
- 

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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.  
<http://www.securiteam.com/maillinglist.html>

-----

Cisco IOS Stack Group Bidding Protocol Crafted Packet DoS

---

## SUMMARY

"Multilink PPP (MLP) allows users to combine multiple PPP links into a single logical network connection, thus enabling on demand bandwidth allocation. When implemented across multiple device chassis, this is known as Multichassis Multilink PPP (MMP). The Stack Group Bidding Protocol is the mechanism by which devices participating in MMP locate each other and negotiate for a connection termination point".

The Cisco IOS Stack Group Bidding Protocol (SGBP) feature in certain versions of Cisco IOS software is vulnerable to a remotely-exploitable denial of service condition. Successful exploitation of this vulnerability may cause the affected device to become unresponsive and trigger a hardware reset, resulting in a denial of service condition.

## DETAILS

Vulnerable Systems:

\* Check original advisory for full system list (  
<<http://www.cisco.com/warp/public/707/cisco-sa-20060118-sgbp.shtml>> here)

## [NEWS] Cisco IOS Stack Group Bidding Protocol Crafted Packet DoS

The SGBP implementation provided by the Cisco Internetwork Operating System (IOS) is susceptible to a denial of service attack when presented with a crafted UDP packet. Sending such a packet to port 9900 of an affected device will cause it to freeze and stop responding to or passing traffic. After a delay, the system watchdog timer will detect this condition and force a reset of the device. The system recovery behavior will be controlled by the device configuration register; for example, the router may reload or drop to the ROM monitor.

This vulnerability affects any device that runs Cisco IOS and has enabled the SGBP protocol. SGBP is enabled by defining a stack group, which is done using the global IOS command "sgbp group <name>". The presence of this command will cause the device to begin listening on port 9900, even if the remaining SGBP parameters are not fully configured.

The following examples demonstrate device configurations for which SGBP is enabled:

```
Router#show sgbp
Group Name: test Ref: 0xA3728C00
Seed bid: default, 50, default seed bid setting
```

Or:

```
Router#show running-config | include sgbp
sgbp group test_group
```

If your device displays output similar to either of the above examples, please consult the IOS software table below to determine whether your version of IOS is affected.

Cisco products that do not run IOS, do not contain support for SGBP, or do not have SGBP enabled are not affected by this vulnerability.

Systems on which SGBP is not supported or enabled will return either blank output or an error message. The following examples demonstrate device configurations that are not affected by this vulnerability:

\* A system that supports but is not enabled for SGBP returns this output:

```
Router#show sgbp
Router#
```

\* A system that does not support SGBP returns this error message:

```
Router#show sgbp
Router#show sgbp
^
```

% Invalid input detected at '^' marker.

Workaround:

\* Configure Access Control Lists (ACLs)

## [NEWS] Cisco IOS Stack Group Bidding Protocol Crafted Packet DoS

For sites that require the SGBP protocol to be enabled, it may be possible to apply Access Control Lists (ACLs) to prevent untrusted hosts from exploiting this vulnerability. The following extended access-list can be adapted to your network. This example assumes that the SGBP members communicate using the 192.168.10.0 network.

```
access-list 101 permit udp 192.168.10.0 0.0.0.255 192.168.10.0
0.0.0.255 port eq 9900
access-list 101 deny udp any 192.168.10.0 0.0.0.255 port eq
9900
access-list 101 permit ip any any
```

The access-list must then be applied to all interfaces using configuration commands such as:

```
interface ethernet 0/0
ip access-group 101 in
```

### \* Enable Control Plane Policing

The Control Plane Policy (CoPP) feature may be used to mitigate this vulnerability, as in this example:

```
! Do not police SGBP traffic from the trusted network
access-list 140 deny udp 192.168.10.0 0.0.0.255 any eq 9900
! Police SGBP traffic from untrusted hosts and networks
access-list 140 permit udp any any eq 9900
! Do not police any other type of traffic going to the router
access-list 140 deny ip any any
!
class-map match-all sgbp-class
match access-group 140
!
policy-map control-plane-policy
! Drop all traffic that matches the class "sgbp-class"
class sgbp-class
drop
!
control-plane
service-policy input control-plane-policy
```

Note: CoPP is only available on certain platforms and IOS release trains. Additional information on the configuration and use of the CoPP feature can be found at the following URL:

[http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products\\_white\\_paper09186a0080211f39.shtml](http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_white_paper09186a0080211f39.shtml)  
Deploying Control Plane Policing White Paper

### \* Infrastructure ACLs (iACL)

Although it is often difficult to block traffic transiting your network, it is possible to identify traffic which should never be allowed to target

[NEWS] Cisco IOS Stack Group Bidding Protocol Crafted Packet DoS

your infrastructure devices and block that traffic at the border of your network. Infrastructure ACLs are considered a network security best practice and should be considered as a long-term addition to good network security as well as a workaround for this specific vulnerability. The white paper entitled "Protecting Your Core: Infrastructure Protection Access Control Lists" presents guidelines and recommended deployment techniques for iACLs:

<<http://www.cisco.com/warp/public/707/iacl.html>>  
<http://www.cisco.com/warp/public/707/iacl.html>

\* Configuring Receive Access Lists (rACLs)

For distributed platforms, rACLs may be an option starting in Cisco IOS Software Versions 12.0(21)S2 for the 12000 series GSR and 12.0(24)S for the 7500 series. The receive access lists protect the device from harmful traffic before the traffic can impact the route processor. Receive path ACLs are considered a network security best practice, and should be considered as a long-term addition to good network security, as well as a workaround for this specific vulnerability. The CPU load is distributed to the line card processors and helps mitigate load on the main route processor. The white paper entitled "GSR:

Receive Access Control Lists" will help identify and allow legitimate traffic to your device and deny all unwanted packets:

<<http://www.cisco.com/warp/public/707/racl.html>>  
<http://www.cisco.com/warp/public/707/racl.html>

ADDITIONAL INFORMATION

The original article can be found at:

<<http://www.cisco.com/warp/public/707/cisco-sa-20060118-sgbp.shtml>>  
<http://www.cisco.com/warp/public/707/cisco-sa-20060118-sgbp.shtml>

=====

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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====  
=====

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.

---

- Prev by Date: [\[NT\] BitComet URI Buffer Overflow](#)
- Next by Date: [\[NEWS\] Cisco Call Manager DoS](#)
- Previous by thread: [\[NT\] BitComet URI Buffer Overflow](#)
- Next by thread: [\[NEWS\] Cisco Call Manager DoS](#)
- Index(es):
  - ◆ [Date](#)
  - ◆ [Thread](#)