

[UNIX] Linux Virtual Server/Secure Context Proofs Shared Permissions Flaw

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0015.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/05/04

To: list@securiteam.com

Date: 5 Jul 2004 14:10:47 +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>

Linux Virtual Server/Secure Context Proofs Shared Permissions Flaw

SUMMARY

<<http://www.linux-vserver.org/>> Linux Virtual Server "extends the Linux kernel to provide the ability to run several virtual servers on a single host system. In contrast to other virtualization attempts Linux Virtual Server uses a split userland architecture under a single kernel to optimize sharing of all resources and reduce resource consumption overhead per VM to the absolute minimum". During a security audit on the vproc security scheme a permission sharing vulnerability was discovered.

DETAILS

Vulnerable Systems:

- * Linux Virtual Server version 1.27 and prior (Linux 2.4 stable branch)
- * Linux Virtual Server version 1.3.9 and prior (Linux 2.4 devel branch)
- * Linux Virtual Server version 1.9.1 and prior (Linux 2.6 devel branch)

Immune Systems:

- * Linux Virtual Server version 1.28

While auditing and experimenting with VServer proofs and vproc security

Securiteam: [UNIX] Linux Virtual Server/Secure Context Procfs Shared Permissions Flaw

Veit discovered a problem sharing permissions on the procfs mounted directories:

Within any context users are still able to change permissions on /proc, both access permission and ownership. That is just fine as many people would like to restrict access to /proc to the root user or a group of trusted users.

But as changes to a procfs mountpoint do not apply to the mountpoint itself but to procfs in general, these changes affect all contexts (VServers) and even the host system.

All tests were done against the stable branch (1.2x) but regarding to Herbert Poetzl, the problem exists on both devel branches (1.3.x, 1.9.x), too.

Version 1.28 (stable branch) resolves this problem.

Exploitation:

The vulnerability may be locally exploited in two ways:

1. From within a virtual server a denial of service attack (DoS) may be provoked towards other virtual servers and the host system. By setting permissions that prevent users other than root to read information from procfs (i.e. process information) will disable a wide range of services.
2. On systems where access to procfs is allowed to root only (or to a group of trusted users; i.e. shared hosting environments), an attacker may use access to another virtual server to gain critical information about processes or other data on the primary target virtual server (or the host system).

Workaround:

To work around this problem, procfs may be mounted read-only. On the host-system do:

```
# mount -o remount,ro /proc
```

As this also prevents the host system from changing any values in /proc, this should just be a temporary solution.

Disclosure Timeline:

2004-06-30 Vulnerability discovered

2004-07-02 Vendor informed

2004-07-03 First vendor response, confirmation

2004-07-04 Official fix available, advisory release

ADDITIONAL INFORMATION

The information has been provided by <<mailto:cru@zodia.de>> Veit Wahlich.

The original article can be found at:

<<http://ircnet.de/article.shtml?vsproc>>

Securiteam: [UNIX] Linux Virtual Server/Secure Context Proofs Shared Permissions Flaw

<http://ircnet.de/article.shtml?vsproc>

=====

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

In order to subscribe to the mailing list, simply forward this email to: 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.