

[NEWS] Mac OS X launchd Race Condition Vulnerability

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-06/0048.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 06/14/05

To: list@securiteam.com

Date: 14 Jun 2005 18:55:49 +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>

Mac OS X launchd Race Condition Vulnerability

SUMMARY

"launchd manages daemons, both for the system as a whole and for individual users. Ideal daemons can launch on demand based on criteria specified in their respective XML property lists located in one of the directories specified in the FILES section.

During boot launchd is invoked by the kernel to run as the first process on the system and to further bootstrap the rest of the system."

By replacing file sockets used by launchd with symbolic links to a file that the attacker does not have access to, it is possible to gain ownership of this file or run code with root privileges.

DETAILS

Vulnerable Systems:

- * Mac OS X version 10.4

Immune Systems:

- * Mac OS X version 10.4.1

- * Mac OS X Server version 10.4.11

Securiteam: [NEWS] Mac OS X launchd Race Condition Vulnerability

The launchd tool, when invoked, uses the launchd_server_init() function to set up temporary files in the /tmp directory. It first creates a directory with the name of the invoking user's current uid. It then uses the chown() function to modify the ownership of the directory to belong to the invoking user.

After this has occurred a socket (file) is created inside this directory and the chown() function is again called to give this file ownership permissions of the invoking user.

A race condition exists here. If a malicious user removes the newly created socket and replaces it with a symbolic link to any file which they don't own. If this is timed to be in between the function call to create the socket, and the chown() call the symbolic link will be followed. This gives the malicious user the ability to effectively "steal" the ownership of any file or directory on the system.

Successful exploitation of this vulnerability will allow a malicious user to "steal" ownership of any file on the system. Using this to gain access to a root shell is a trivial process.

Vendor Status:

Information about vendor update can be found at:

<<http://docs.info.apple.com/article.html?artnum=301742>>

<http://docs.info.apple.com/article.html?artnum=301742>

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1725>>

CAN-2005-1725

ADDITIONAL INFORMATION

The information has been provided by <<mailto:advisories@suresec.org>>
Suresec Advisories.

The original article can be found at:

<<http://www.suresec.org/advisories/adv3.pdf>>

<http://www.suresec.org/advisories/adv3.pdf>

=====

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.