

[UNIX] Legacy Apache mod_jk2 Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2008-02/msg00039.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 13 Feb 2008 11:27:22 +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>

Legacy Apache mod_jk2 Buffer Overflow

SUMMARY

IOActive has discovered a buffer overflow in the Host Header field in the legacy version of the mod_jk2 Apache module (jakarta-tomcat-connectors) which allows for remote code execution in the context of the Apache process.

DETAILS

Vulnerable Systems:

- * mod_jk2 version 2.0.3-DEV
- * F5 BIG-IP version 9.2.3.30

Immune Systems:

- * mod_jk2 version 2.0.4-DEV

mod_jk2 versions prior to 2.0.4 are vulnerable to multiple stack overflow vulnerabilities. Specifically, IOActive has discovered multiple locations where these vulnerabilities are exploitable via the Host request header in any given request. These overflows all result in remote code execution under the user of the running Apache process. Although a legacy module which is end of life, certain vendors may use this module in their

[UNIX] Legacy Apache mod_jk2 Buffer Overflow

products rendering them vulnerable to remote exploitation.

Technical Details:

Within the mod_jk2 module, the module registers with Apache a request handler which parses the entire content of the request, specifically the Host headers, in order to determine which Tomcat worker to forward the request to. For example, multiple buffer overflow opportunities exist within the following code segments:

```
native2\common\jk_uriMap.c: line ~269
if (port) {
if (vhost) {
if (strchr(vhost, ':'))
strcpy(hostname, vhost);
else
sprintf(hostname, "%s:%d", vhost, port);
}
else
sprintf(hostname, ":%d", port);
}
else if (vhost)
strcpy(hostname, vhost);
```

```
native2\common\jk_uriMap.c: line ~842
char key[1024];

if (!vhost && !port)
return uriMap->vhosts->get(env, uriMap->vhosts, "*");
if (!vhost)
vhost = "*";
sprintf(key, "%s:%d", vhost, port);
return uriMap->vhcache->get(env, uriMap->vhcache, key);
```

In each of these code segments, exploitable stack overflows on the Host request header are visible. Additionally, in every circumstance, the condition occurs when a Hostname is provided within the Host: Header request which is longer than 1024 characters. Exploitation of these overflows is considered trivial.

Remediation:

Upgrade to the latest version of the legacy mod_jk2 (mod_jk2 2.0.4) or migrate to the non-legacy reimplementation of this package, the new jakarta-tomcat-connectors, called mod_jk.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:advisory@xxxxxxxxxxxxx>>
IOActive Advisories.

[UNIX] Legacy Apache mod_jk2 Buffer Overflow

=====

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@xxxxxxxxxxxxxxxxx

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

=====

=====

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.