

# [NEWS] Bypassing UnrealIRCd IP Cloaking

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0022.html>

---

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

**Date:** 07/07/04

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

Date: 7 Jul 2004 16:30:39 +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>

-----

Bypassing UnrealIRCd IP Cloaking

---

## SUMMARY

<<http://www.unrealircd.com/>> UnrealIRCd is a popular IRC server. One of the features it provides is called 'ip cloaking'. The purpose of this system is to prevent hostile IRC users from getting the IP address of other users. Due to weak hashing function, brute forcing of the cloaked IP is easy.

## DETAILS

Vulnerable Systems:

- \* UnrealIRCd version 3.2 and prior versions

Immune Systems:

- \* UnrealIRCd version 3.2.1

In order to prevent possible brute force attack on the client IP, the Unreal IRC server uses three 'keys'. However, the hash function is weak. This makes it possible to recover the keys of several IRC networks by knowing only one clear text and hashed IP, and another hashed IP.

Vulnerable Code:

The IPv4 hashing scheme is the most vulnerable. Code from cloak.c follows:

## Securiteam: [NEWS] Bypassing UnrealIRCd IP Cloaking

```
/* Do IPv4 cloaking here */
strcpy(h1, host, sizeof h1);
i = 0;
for (i = 0, p = strtok(h1, "."); p && (i <= 3); p = strtok(NULL, "."),
i++)
{
    strncpy(h2[i], p, 4);
}
ircsprintf(h3, "%s.%s", h2[0], h2[1]);
l[0] = ((our_crc32(h3, strlen(h3)) + KEY1) ^ KEY2) + KEY3;
ircsprintf(h3, "%s.%s.%s", h2[0], h2[1], h2[2]);
l[1] = ((KEY2 ^ our_crc32(h3, strlen(h3))) + KEY3) ^ KEY1;
l[4] = our_crc32(host, strlen(host));
l[2] = ((l[4] + KEY3) ^ KEY1) + KEY2;
l[2] &= 0x3FFFFFFF;
l[0] &= 0x7FFFFFFF;
l[1] &= 0xFFFFFFFF;
snprintf(cloaked, sizeof cloaked, "%1X.%1X.%1X.IP", l[2], l[1], l[0]);
free(host);
return cloaked;
```

In the code above:

h2[0], h2[1], h2[2], h2[3] contain the four bytes of the original IP.

l[0], l[1], l[2] contain the hashed IP.

Thus:

```
l[0] = (((crc32("1.2") + key1) ^ key2) + key3) & 0x7FFFFFFF;
l[1] = (((crc32("1.2.3") ^ key2) + key3) ^ key1) & 0xFFFFFFFF;
l[2] = (((crc32("1.2.3.4") + key3) ^ key1) + key2) & 0x3FFFFFFF;
```

crc32(xxx) and l[x] are known. The three keys are used in such a way that the n-th bit of any key does not affect bits below n in the hash.

A program that runs a brute force attack one bit at a time was written as a Proof of Concept. It takes less than one second to do that on a Pentium4 1.8ghz.

Doing this on a known IP produces around 2000 possible key combinations. It is then trivial to test them all in order to find the working ones.

### ADDITIONAL INFORMATION

The information has been provided by <mailto:bartavelle@bandecon.com>  
bartavelle.

The original article can be found at:

<<http://www.bandecon.com/advisory/unreal.txt>>  
<http://www.bandecon.com/advisory/unreal.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:

[NEWS] Bypassing UnrealIRCd IP Cloaking

Securiteam: [NEWS] Bypassing UnrealIRCd IP Cloaking

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.