

Securiteam: [UNIX] Remote DoS Vulnerability in Netfilter's Subsystem (tcp-option)

[UNIX] Remote DoS Vulnerability in Netfilter's Subsystem (tcp-option)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0009.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/04/04

To: list@securiteam.com

Date: 4 Jul 2004 19:04:38 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.secureteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.secureteam.com/maillinglist.html>

Remote DoS Vulnerability in Netfilter's Subsystem (tcp-option)

SUMMARY

There is a remotely exploitable bug in all Linux kernel 2.6 series due to using incorrect variable type. Vulnerability is connected to netfilter subsystem and may cause DoS. It's disclosed only when using iptables with rules matching TCP options (i.e. --tcp-option). There is no difference what action is taking up by matching rule.

Vulnerability was detected on i386 architecture. The other ones weren't tested but it seems to be vulnerable too.

DETAILS

Problem lies in `tcp_find_option()` function (`net/ipv4/netfilter/ip_tables.c`). There is local array `opt` defined as:

```
char opt[60 - sizeof(struct tcphdr)];
```

Which contains TCP options extracted from packet. Function mentioned above searches for specified option in this array.

Options in TCP packet, with some exceptions, are organized in the

Securiteam: [UNIX] Remote DoS Vulnerability in Netfilter's Subsystem (tcp-option)

following way:

```
Octet no. Length Field
-----
0 1 Opcode
1 1 Length of all option (N + 2)
2 N Params
```

The function iterates over options in array:

```
for (i = 0; i < optlen; ) {
    if (opt[i] == option) return !invert;
    if (opt[i] < 2) i++;
    else i += opt[i+1]?:1;
}
```

Moving counter by the option length.

But, in case the `length' value is greater than 127, the value of this octet in `opt' is implicitly casted to char, which results in negative number and the loop counter moving back. In some cases it is possible that counter cycles through the contents of this array infinitely.

Impact:

After sending one suitably prepared TCP packet to victim host, kernel goes into infinite loop consuming all CPU resources, rendering the box unresponsive. Of course, there is no need to have a shell access to attacked host.

Exploitation:

Example of packet-of-death:

```
0x0000: 4500 0030 1234 4000 ff06 e83f c0a8 0001
0x0010: c0a8 0002 0400 1000 0000 0064 0000 0064
0x0020: 7000 0fa0 dc6a 0000 0204 05b4 0101 04fd
```

Fix:

There is only need to change type of `opt' array from signed char to unsigned (or, better to `u_int8_t`) as it was defined in 2.4 kernels or prior to version 1.16 of `net/ipv4/netfilter/ip_tables.c` file.

```
--- net/ipv4/netfilter/ip_tables.c.orig 2004-04-04 05:36:47.000000000
+0200
+++ net/ipv4/netfilter/ip_tables.c 2004-06-24 21:24:26.000000000 +0200
@@ -1461,7 +1461,7 @@
     int *hotdrop)
 {
     /* tcp.doff is only 4 bits, ie. max 15 * 4 bytes */
-   char opt[60 - sizeof(struct tcphdr)];
+   u_int8_t opt[60 - sizeof(struct tcphdr)];
     unsigned int i;

     duprintf("tcp_match: finding option\n");
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:adwol@polsl.gliwice.pl> Adam Osuchowski (Computer Centre of The Silesian University of Technology) and Tomasz Dubinski.

=====

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.