

Official Sebek 2 client for OpenBSD

Fast study

Arnaud EBALARD <troglocan@rstack.org>

Pierre LALET <pierre@rstack.org>

Olivier MATZ <zer0@rstack.org>

Copyright © 2004 *Droids Corporation*.

March 4, 2004

Contents

1	Introduction	2
2	Stability	2
3	Fingerprinting	2
4	BPF fingerprinting	2
5	Disabling Sebek, getting more information	3
6	Conclusion	5

1 Introduction

This is a fast study of the Sebek official client for OpenBSD (public release version). You can get more information about the HoneyNet Project on <http://www.honeynet.org>. Sebek Homepage is here : <http://www.honeynet.org/tools/sebek/>. You can download Sebek client for OpenBSD here : <http://www.dragos.com/sebek/>.

We are three guys from the Droids Corporation working on Sebek for *BSD (<http://droids.rstack.org/sebek/>¹) with the French HoneyNet Project (<http://www.frenchhoneynet.org/>).

We thought the experience gathered working on Sebek could help the community to improve Sebek-OpenBSD.

We have worked on an OpenBSD 3.4 kernel with Sebek-OpenBSD 2.6.

2 Stability

Although this version is a public and official release, the Sebek-OpenBSD kernel is not stable at all. When we tested it, adding a new user caused the system to reboot. An easy way to reboot your Sebek-OpenBSD computer is to type, for example : `cp /bin/sh /tmp`.

Due to this instability, we could not test all the points we wanted to.

3 Fingerprinting

A normal (understand *non-root*) user can easily detect Sebek, just by reading the kernel file. We are going to use `gdb` and disassemble the function `dofileread`, which calls `sebeklog` on a Sebek-OpenBSD kernel.

Let's try it :

```
# echo "disassemble dofileread" | gdb -q /bsd | grep sebek
0xd01c9bdc <dofileread+292>:    call    0xd01c967c <sebeklog>
```

4 BPF fingerprinting

We can use BPF (Berkeley Packet Filter – see `bpf(4)`, *OpenBSD Programmer's Manual*) to fingerprint Sebek. This can be useful if the kernel file has been stripped.

¹The Sebek client for OpenBSD you'll find on that web page is just our client for NetBSD ported to OpenBSD, and has nothing to do with Dragos'one. Don't reproach him our code...

The Sebek traffic is hidden, but the BPF counter is incremented, and you can see this by using `tcpdump`.

```
# tcpdump -i le1 -w foo &
[1] 18729
# tcpdump: listening on le1
ls
.cshrc  .klogin  .login  .profile  .ssh      foo
# fg
tcpdump -i le1 -w foo
^C
53 packets received by filter
0 packets dropped by kernel
# tcpdump -n -r foo | wc -l
    26
```

Let's explain this : `tcpdump` has only seen 26 packets, while the BPF counter tells 53. As we have not specified any filter, each packet should be seen. Normally we should have got at least 50 packets. We have only 26. Something is wrong.

5 Disabling Sebek, getting more information

First, you need to know that during the configuration of Sebek-OpenBSD, you do not need to set the destination MAC address. This means that Sebek sends its packets using the kernel routing table.

We are going to change this table. If your host has no specific route for him, add a route for your host. For example, if you are connected from 10.0.1.1, and the routing table is :

```
default          10.0.0.1  le1
10.0.0.0/24      link#1    le1
```

you need to add a new route, at least for your host. You can do this with the command `route add -host 10.0.1.1 10.0.0.1`.

When you're ok with the routing table, just do :

```
# route delete default
delete net default
# route add default 127.0.0.1
add net default: gateway 127.0.0.1
```

At this stage, we need to add that if the Sebek log server is not reached by the `default` entry of the routing table (in our example, that means that it has an address in the `10.0.0.0/24` network), you'll need to add more precise routes (`10.0.0.0/25` and `10.0.0.128/25` in our example). While you are doing this, do not forget to add precise routes for important hosts you need to reach.

Now, let's play :

```
# tcpdump -i lo0 -w foo&
[1] 2129
# tcpdump: listening on lo0
ls
.cshrc  .klogin  .login  .profile  .ssh      foo
# fg
tcpdump -i lo0 -w foo
^C
28 packets received by filter
0 packets dropped by kernel
```

And let's read the packets :

```
# tcpdump -nr foo -x | tail -7
11:42:58.210151 127.0.0.1.1101 > 10.11.12.13.1101:  udp 24
                                4500 0064 7a22 0000 8011 2b4e 7f00 0001
                                0a0b 0c0d 044d 044d 0020 0000 0209 0001
                                0001 0000 0000 1bf5 4045 b6b2 0003 3450
                                0000 651b 0000 0000 0000 0008 0000 0000
                                7373 6864 004e 6ad4 0000 0018 7463 7064
                                756d 7020 2d69 206c 6f30 202d
```

You can read the most important information. As we use the `lo0` interface, the source IP address is `127.0.0.1` (`7f:00:00:01`). So we have the destination IP address (which is the Sebek server IP address) : `0a:0b:0c:0d` (`10.11.12.13`). We can read (before the addresses) the protocol (`0x11` : UDP), and the TTL (`0x80`).

The UDP header gives us the UDP source port (`04:4d` : `1101`) and the UDP destination port (the same). The beginning of the UDP datas is the Magic number (`02:09:00:01`).

Someone who knows Sebek knows that all these values are important.

6 Conclusion

There are certainly plenty of other ways to fingerprint and disable Sebek OpenBSD client and to get sensitive information. We have not audited or even read the source code. This report is not supposed to be exhaustive.

Maybe if you are planning to build honeypots using Sebek OpenBSD client, you should consider to wait for a more secure version.