MWR InfoSecurity Security Advisory

OpenSC - "Get Serial Number" Stack-based Buffer Overflow

13th December 2010

MWR INFOSECURITY

MWR INFOSECURITY

OpenSC – "Get Serial Number" Stack-based Buffer Overflow

Package Name:	OpenSC
Date Reported	3rd November 2010
Affected Versions:	Confirmed in Version 0.11.13 and earlier

CVE Reference	Not Yet Assigned
Author	Rafael Dominguez Vega
Severity	Medium Risk
Vulnerability Class	Stack-based buffer overflow
Vendor	OpenSC - http://www.opensc-project.org
Vendor Response	The vendor has implemented a fix. https://www.opensc-project.org/opensc/changeset/4913
Exploit Details Included	No

Overview

MWR InfoSecurity identified a vulnerability in OpenSC. The vulnerability can be triggered using a malicious smart card.

Impact

An attacker could use this vulnerability to execute arbitrary code in the target system. To successfully exploit this vulnerability the attacker will be required to insert a specially crafted smart card in the target system.

Cause

A buffer overflow vulnerability was identified in the code handling the smart card's serial number in the following drivers:

- card-atrust-acos.c
- card-acos5.c
- card-starcos.c

Interim Workaround

The interim work around for this issue will require the affected drivers to be removed from OpenSC and this to be recompiled.

Solution

The vendor has implemented a fix. Users should upgrade to the latest version of OpenSC. https://www.opensc-project.org/opensc/changeset/4913

Dependencies

In order to successfully exploit the vulnerability described in this advisory, an attacker would

need to have physical access to the affected system in order to be able to plug in a malicious smart card.

Detailed Vulnerability Description

The issue is a stack-based buffer overflow affecting the following drivers, in the "Get Serial Number" function.

- card-atrust-acos.c
- card-acos5.c
- card-starcos.c

The affected code is included here. The vulnerability is in the memcpy shown below, as the serial number that the card sends can be larger (up to 258 bytes) than the buffer where the data is being copied to (32 bytes).

```
#define SC_MAX_SERIALNR         32

#define SC_MAX_APDU_BUFFER_SIZE         258

u8  rbuf[SC_MAX_APDU_BUFFER_SIZE];
...
apdu.resp = rbuf;
apdu.resplen = sizeof(rbuf);
...
memcpy(card->serialnr.value, apdu.resp, apdu.resplen);
```

During the investigation of this vulnerability a Proof-of-Concept smart-card was created. The malicious smart card was specially developed to trigger this issue and overwrite the value of the instruction pointer.

```
(gdb) r
Starting program: /usr/local/bin/opensc-tool --serial
[Thread debugging using libthread_db enabled]
41 31 41 32 41 33 41 34 42 31 42 32 42 33 42 34 A1A2A3A4B1B2B3B4
43 31 43 32 43 33 43 34 44 31 44 32 44 33 44 34 C1C2C3C4D1D2D3D4
DA 00 00 00 A0 12 E8 B7 D2 89 04 08 20 4C D1 B7 ............ L..
54 84 04 08 01 00 00 00 F4 6F F5 B7 2E 4E 3D F6 T........o...N=.
28 78 F5 B7 30 6F F5 BF BF 32 F4 B7 20 6F F5 BF (x..0o...2.. o..
54 84 04 08 14 6F F5 BF CC 77 F5 B7 00 00 00 00 T....o...w......
88 C7 B6 B7 01 00 00 00 00 00 00 00 01 00 00 00 ................
58 76 F5 B7 00 00 00 00 20 67 E8 B7 00 00 00 80 Xv...... g......
20 10 D1 B7 A0 12 E8 B7 20 6F F5 BF 14 6F F5 BF  ....... o...o..
F4 6F F5 B7 C8 8A B7 B7 C8 1A E8 B7 60 6F F5 BF .o..........`o..
70 76 F5 B7 D2 89 04 08 D8 CD B6 B7 00 00 00 00 pv..............
00 00 00 00 00 00 00 00 AF 5F B8 B7 15 00 00 00 ........._......
00 D0 B6 B7 48 B0 13 00 01 00 00 00 F5 62 D3 B7 ....H........b..
16 27 C8 B7 C8 1A E8 B7 07 00                    .'........
[V4W1W2-tool] reader-pcsc.c:678:pcsc_unlock: SCardEndTransaction failed: Reader is
unavailable.
[New Thread 0xb7b53a00 (LWP 4126)]

Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0xb7b53a00 (LWP 4126)]
0x41414141 in ?? ()
(gdb) i r
```

```
eax            0x41414141    1094795585
ecx            0x0    0
edx            0x8c93008     147402760
ebx            0xb7f25ff4    -1208852492
esp            0xbff56b1c    0xbff56b1c
ebp            0xbff56b48    0xbff56b48
esi            0x8c93138     147403064
edi            0x8c93008     147402760
eip            0x41414141    0x41414141
eflags         0x210206      [ PF IF RF ID ]
cs             0x73   115
ss             0x7b   123
ds             0x7b   123
es             0x7b   123
fs             0x0    0
gs             0x33   51
```

Acknowledgement

Thanks to Nils for the support and guidance on this research.

Thanks to Martin Paljak of OpenSC for his co-operation in working with the author in regards to this matter and acknowledge his prompt response in implementing a fix.

MWR InfoSecurity
St. Clement House
1-3 Alencon Link
Basingstoke, RG21 7SB
Tel: +44 (0)1256 300920
Fax: +44 (0)1256 844083
mwrinfosecurity.com