

MWR InfoSecurity Security
Advisory

PCSC-Lite:
pcscd ATR Handler
Buffer Overflow

13th December 2010

MWR  INFOSECURITY

PCSC-Lite: pcscd ATR Handler Buffer Overflow Vulnerability

Package Name:	PCSC-Lite
Date Reported	3 rd November 2010
Affected Versions:	Confirmed in Version 1.5.3

CVE Reference	Not Yet Assigned
Author	Rafael Dominguez Vega
Severity	Medium Risk
Vulnerability Class	Buffer overflow
Vendor	PCSC-Lite - http://pcsc-lite.alieth.debian.org/
Vendor Response	The vendor has implemented a fix. http://lists.alieth.debian.org/pipermail/pcsc-lite-cvs-commit/2010-November/004923.html
Exploit Details Included	No

Overview

MWR InfoSecurity identified a vulnerability in PCSC-Lite's pcscd daemon. The vulnerability can be triggered using a malicious smart card.

Impact

An attacker could use this vulnerability to trigger a denial of service condition or potentially 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 ATR (atrhandler.c). The vulnerability occurs as the value of the length supplied in the affected memcopy can be incremented to be larger than the destination buffer.

Interim Workaround

Disabling the pcscd daemon will prevent users from exploiting the vulnerability.

Solution

The vendor has implemented a fix. Users should upgrade to the latest version of PCSC-Lite. <http://lists.alieth.debian.org/pipermail/pcsc-lite-cvs-commit/2010-November/004923.html>

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 buffer overflow affecting the code responsible for handling the ATR coming from the card (atrhandler.c).

The affected code is included here. The vulnerability is in the memcpy shown below, as the value of "p" which is used as the length in the memcpy can be incremented to be larger (up to 45 bytes) than the destination buffer.

```
#define MAX_ATR_SIZE          33

UCHAR Value[MAX_ATR_SIZE];

p = K = TCK = Yli = T = 0;

Yli = pucAtr[1] >> 4;
K = pucAtr[1] & 0x0F;
p = 2;
int i = 1;

do {
    short TAi, TBi, TCi, TDi;
    TAi = (Yli & 0x01) ? pucAtr[p++] : -1;
    TBi = (Yli & 0x02) ? pucAtr[p++] : -1;
    TCi = (Yli & 0x04) ? pucAtr[p++] : -1;
    TDi = (Yli & 0x08) ? pucAtr[p++] : -1;
    ...
    if (p > MAX_ATR_SIZE) return 0;

    i++;
}
while (Yli != 0);

psExtension->ATR.HistoryLength = K;
memcpy(psExtension->ATR.HistoryValue, &pucAtr[p], K);
p = p + K;

memcpy(psExtension->ATR.Value, pucAtr, p);
```

Source Code from atrhandler.c

Acknowledgement

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

Thanks to Ludovic Rousseau 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