MOAUB

Abysssec Research

1) Advisory information

Title	:	Mozilla Firefox XSLT Sort Remote Code Execution Vulnerability
Version	:	Firefox 3.6.3
Analysis	:	http://www.abysssec.com
Vendor	:	http://www.mozilla.com
Impact	:	High/Critial
Contact	:	shahin [at] abysssec.com , info [at] abysssec.com
Twitter	:	@abysssec
CVE	:	CVE-2010-1199

2) Vulnerable version

Ubuntu Ubuntu Linux 9.10 sparc
Ubuntu Ubuntu Linux 9.10 powerpc
Ubuntu Ubuntu Linux 9.10 lpia
Ubuntu Ubuntu Linux 9.10 i386
Ubuntu Ubuntu Linux 9.10 amd64
Ubuntu Ubuntu Linux 9.04 sparc
Ubuntu Ubuntu Linux 9.04 powerpc
Ubuntu Ubuntu Linux 9.04 lpia
Ubuntu Ubuntu Linux 9.04 i386
Ubuntu Ubuntu Linux 9.04 amd64
Ubuntu Ubuntu Linux 8.04 LTS sparc
Ubuntu Ubuntu Linux 8.04 LTS powerpc
Ubuntu Ubuntu Linux 8.04 LTS lpia
Ubuntu Ubuntu Linux 8.04 LTS i386
Ubuntu Ubuntu Linux 8.04 LTS amd64
Ubuntu Ubuntu Linux 10.04 sparc
Ubuntu Ubuntu Linux 10.04 powerpc
Ubuntu Ubuntu Linux 10.04 i386
Ubuntu Ubuntu Linux 10.04 amd64
SuSE SUSE Linux Enterprise SDK 11 SP1

SuSE SUSE Linux Enterprise SDK 11
SuSE SUSE Linux Enterprise SDK 10 SP3
Slackware Linux x86_64 -current
Slackware Linux 13.1 x86_64
Slackware Linux 13.1
Slackware Linux 13.0 x86_64
Slackware Linux 13.0
Slackware Linux 12.2
Slackware Linux -current
S.u.S.E. SUSE Linux Enterprise Server 11 SP1
S.u.S.E. SUSE Linux Enterprise Server 11
+ Linux kernel 2.6.5
S.u.S.E. SUSE Linux Enterprise Server 10 SP3
S.u.S.E. SUSE Linux Enterprise Desktop 11 SP1
+ Linux kernel 2.6.5
S.u.S.E. SUSE Linux Enterprise Desktop 11
S.u.S.E. SUSE Linux Enterprise Desktop 10 SP3
S.u.S.E. openSUSE 11.2
S.u.S.E. openSUSE 11.1
S.u.S.E. openSUSE 11.0
RedHat Fedora 13
RedHat Fedora 12
RedHat Enterprise Linux WS 4
RedHat Enterprise Linux WS 3
RedHat Enterprise Linux Optional Productivity Application 5 server
RedHat Enterprise Linux ES 4
RedHat Enterprise Linux ES 3
RedHat Enterprise Linux Desktop Workstation 5 client
RedHat Enterprise Linux Desktop 5 client
RedHat Enterprise Linux AS 4
RedHat Enterprise Linux AS 3
RedHat Enterprise Linux Desktop version 4
RedHat Enterprise Linux 5 server
RedHat Desktop 3.0
Pardus Linux 2009 0
Mozilla Thunderbird 3.0.4
Mozilla Thunderbird 3.0.2
Mozilla Thunderbird 3.0.1
Mozilla Thunderbird 2.0 24
Mozilla Thunderbird 2.0 .9
Mozilla Thunderbird 2.0 .8
Mozilla Thunderbird 2.0 .6
Mozilla Thunderbird 2.0 .5
Mozilla Thunderbird 2.0 .4
Mozilla Thunderbird 2.0 .19
Mozilla Thunderbird 2.0 .17
Mozilla Thunderbird 2.0 .16
Mozilla Thunderbird 2.0 .15

Mozilla Thunderbird 2.0 .14
Mozilla Thunderbird 2.0 .13
Mozilla Thunderbird 2.0 .12
Mozilla Thunderbird 3.0
Mozilla Thunderbird 2.0.0.23
Mozilla Thunderbird 2.0.0.22
Mozilla Thunderbird 2.0.0.21
Mozilla Thunderbird 2.0.0.18
Mozilla SeaMonkey 2.0.4
Mozilla SeaMonkey 2.0.3
Mozilla SeaMonkey 2.0.2
Mozilla SeaMonkey 2.0.1
Mozilla SeaMonkey 1.1.19
Mozilla SeaMonkey 1.1.18
Mozilla SeaMonkey 1.1.17
Mozilla SeaMonkey 1.1.16
Mozilla SeaMonkey 1.1.15
Mozilla SeaMonkey 1.1.14
Mozilla SeaMonkey 1.1.13
Mozilla SeaMonkey 1.1.12
Mozilla SeaMonkey 1.1.11
Mozilla SeaMonkey 1.1.10
Mozilla SeaMonkey 1.1.9
Mozilla SeaMonkey 1.1.8
Mozilla SeaMonkey 1.1.7
Mozilla SeaMonkey 1.1.6
Mozilla SeaMonkey 1.1.5
Mozilla SeaMonkey 1.1.4
Mozilla SeaMonkey 1.1.3
Mozilla SeaMonkey 1.1.2
Mozilla SeaMonkey 1.1.1
Mozilla SeaMonkey 1.0.99
Mozilla SeaMonkey 1.0.9
Mozilla SeaMonkey 1.0.8
Mozilla SeaMonkey 1.0.7
Mozilla SeaMonkey 1.0.6
Mozilla SeaMonkey 1.0.5
Mozilla SeaMonkey 1.0.3
Mozilla SeaMonkey 1.0.2
Mozilla SeaMonkey 1.0.1
Mozilla SeaMonkey 2.0
Mozilla SeaMonkey 1.1 beta
Mozilla SeaMonkey 1.0 dev
Mozilla SeaMonkey 1.0
Mozilla Firefox 3.6.3
Mozilla Firefox 3.6.2
Mozilla Firefox 3.6.2
Mozilla Firefox 3.5.9

Mozilla Firefox 3.5.8

Mozilla Firefox 3.5.7

Mozilla Firefox 3.5.6

Mozilla Firefox 3.5.5

Mozilla Firefox 3.5.4

Mozilla Firefox 3.5.3

Mozilla Firefox 3.5.2

Mozilla Firefox 3.5.1

Mozilla Firefox 3.5

Mozilla Firefox 3.0.19

Mozilla Firefox 3.0.18

Mozilla Firefox 3.0.17

Mozilla Firefox 3.0.16

Mozilla Firefox 3.0.15

Mozilla Firefox 3.0.14

Mozilla Firefox 3.0.13

Mozilla Firefox 3.0.12

Mozilla Firefox 3.0.11

Mozilla Firefox 3.0.10

Mozilla Firefox 3.0.9

Mozilla Firefox 3.0.8

Mozilla Firefox 3.0.7 Beta

Mozilla Firefox 3.0.7

Mozilla Firefox 3.0.6

Mozilla Firefox 3.0.5

Mozilla Firefox 3.0.4

Mozilla Firefox 3.0.3

Mozilla Firefox 3.0.2

Mozilla Firefox 3.0.1

Mozilla Firefox 3.6

Mozilla Firefox 3.1 Beta 3

Mozilla Firefox 3.1 Beta 2

Mozilla Firefox 3.1 Beta 1

Mozilla Firefox 3.0 Beta 5

Mozilla Firefox 3.0

MandrakeSoft Linux Mandrake 2010.0 x86_64

MandrakeSoft Linux Mandrake 2010.0

MandrakeSoft Linux Mandrake 2009.1 x86_64

MandrakeSoft Linux Mandrake 2009.1

MandrakeSoft Linux Mandrake 2009.0 x86_64

MandrakeSoft Linux Mandrake 2009.0

MandrakeSoft Linux Mandrake 2008.0 x86_64

MandrakeSoft Linux Mandrake 2008.0

MandrakeSoft Enterprise Server 5 x86_64

MandrakeSoft Enterprise Server 5

Debian Linux 5.0 sparc

Debian Linux 5.0 s/390

Debian Linux 5.0 powerpc

Debian Linux 5.0 mipsel
Debian Linux 5.0 mips
Debian Linux 5.0 m68k
Debian Linux 5.0 ia-64
Debian Linux 5.0 ia-32
Debian Linux 5.0 hppa
Debian Linux 5.0 armel
Debian Linux 5.0 arm
Debian Linux 5.0 amd64
Debian Linux 5.0 alpha
Debian Linux 5.0
Avaya Messaging Storage Server 5.2
Avaya Messaging Storage Server 5.1
Avaya Messaging Storage Server 5.0
Avaya Messaging Storage Server 4.0
Avaya Message Networking 5.2
Avaya Message Networking 3.1
Avaya Intuity AUDIX LX 2.0 SP2
Avaya Intuity AUDIX LX 2.0 SP1
Avaya Intuity AUDIX LX 2.0

3) Vulnerability information

Class

1- Integer overflow

Impact

An attacker can exploit this issue to execute arbitrary code in the context of the user running the affected application. Failed exploit attempts will likely result in denial-of-service conditions.

Remotely Exploitable

Yes

Locally Exploitable

Yes

4) Vulnerabilities detail

This vulnerabilty exist in the SORT function related to XSLT. For the purpos of sorting on XML file, the execute function of txPushNewContext class is called. This function is in the source path content/xslt/src/xslt/txInstructions.cpp:646 and part of it demontrated below:

```
UnPatch FireFox 3.6.6:

nsresult
txPushNewContext::execute(txExecutionState& aEs)
{
    nsRefPtr<txAExprResult> exprRes;
    nsresult rv = mSelect->evaluate(aEs.getEvalContext(),
                                    getter_AddRefs(exprRes));   // XXX
    NS_ENSURE_SUCCESS(rv, rv);

    if (exprRes->getResultType() != txAExprResult::NODESET) {
        // XXX ErrorReport: nodeset expected
        return NS_ERROR_XSLT_NODESET_EXPECTED;
    }

    txNodeSet* nodes = static_cast<txNodeSet*>
                              (static_cast<txAExprResult*>
                                          (exprRes));   // XXX:
nodes to sort

    if (nodes->isEmpty()) {
        aEs.gotoInstruction(mBailTarget);

        return NS_OK;
    }

    txNodeSorter sorter;
    PRUint32 i, count = mSortKeys.Length();
    for (i = 0; i < count; ++i) {
        SortKey& sort = mSortKeys[i];
        rv = sorter.addSortElement(sort.mSelectExpr, sort.mLangExpr,
   // XXX: number of sort keys
                                   sort.mDataTypeExpr, sort.mOrderExpr,
                                   sort.mCaseOrderExpr,
                                   aEs.getEvalContext());
        NS_ENSURE_SUCCESS(rv, rv);
    }
    nsRefPtr<txNodeSet> sortedNodes;
    rv = sorter.sortNodeSet(nodes, &aEs, getter_AddRefs(sortedNodes));
// XXX
    NS_ENSURE_SUCCESS(rv, rv);
```

In the execute function of sorter object, it collect all of the keys that based on it XML file going to be sorted in to an aray. You see the code of the addSortElement of txNodeSorter class that keys are collected. The path of the code in the source code is content/xslt/src/xslt/txNodeSorter.cpp:68 :

```
nsresult
txNodeSorter::addSortElement(Expr* aSelectExpr, Expr* aLangExpr,
                             Expr* aDataTypeExpr, Expr* aOrderExpr,
                             Expr* aCaseOrderExpr, txIEvalContext*
aContext)
{
...
   // mSortKeys owns key now.
   rv = mSortKeys.add(key);
   NS_ENSURE_SUCCESS(rv, rv);

   key.forget();
   mNKeys++;   // XXX

   return NS_OK;
}
```

After storing the keys in the mSortKeys and their count to the mNkeys the following function is called to allocate required memory for sort operation. Path of the file relating to the class is content/xslt/src/xslt/txNodeSorter.cpp:157:

```
nsresult
txNodeSorter::sortNodeSet(txNodeSet* aNodes, txExecutionState* aEs,
                          txNodeSet** aResult)
{
   if (mNKeys == 0 || aNodes->isEmpty()) {
      NS_ADDREF(*aResult = aNodes);

      return NS_OK;
   }

   *aResult = nsnull;

   nsRefPtr<txNodeSet> sortedNodes;
   nsresult rv =
aEs->recycler()->getNodeSet(getter_AddRefs(sortedNodes));
   NS_ENSURE_SUCCESS(rv, rv);

   txNodeSetContext* evalContext = new txNodeSetContext(aNodes, aEs);
   NS_ENSURE_TRUE(evalContext, NS_ERROR_OUT_OF_MEMORY);
```

```
    rv = aEs->pushEvalContext(evalContext);
    NS_ENSURE_SUCCESS(rv, rv);

    // Create and set up memoryblock for sort-values and indexarray
    PRUint32 len = static_cast<PRUint32>(aNodes->size());   // XXX
    void* mem = PR_Malloc(len * (sizeof(PRUint32) + mNKeys *
sizeof(TxObject*)));   // XXX
    NS_ENSURE_TRUE(mem, NS_ERROR_OUT_OF_MEMORY);

    PRUint32* indexes = static_cast<PRUint32*>(mem);    // XXX
    TxObject** sortValues = reinterpret_cast<TxObject**>(indexes +
len);

    PRUint32 i;
    for (i = 0; i < len; ++i) {
        indexes[i] = i;
    }
    memset(sortValues, 0, len * mNKeys * sizeof(TxObject*));
```

The flaw exists in the following line:

```
void* mem = PR_Malloc(len * (sizeof(PRUint32) + mNKeys *sizeof(TxObject*)));
```

In this function mNkeys variable which mentioned earlier indicate number of sorting keys. In this line of code some memory are allocated by PR_Mallloc function.   If the number of sorting keys is greater than 2byes memory space less space will be allocated in this call. And when using this space in the following lines it cause Access Violation exception. The following code is the fixed part of the software:

```
// Don't overflow when calculating the length of the sort buffer.
PRUint32 itemSize = sizeof(PRUint32) + mNKeys * sizeof(TxObject*);
if (mNKeys > (PR_UINT32_MAX - sizeof(PRUint32)) / sizeof(TxObject*) ||
    len >= PR_UINT32_MAX / itemSize) {
    return NS_ERROR_OUT_OF_MEMORY;
}

void* mem = PR_Malloc(len * itemSize);
```

The attached XMLGenerator.py and XSLGenerator.py generate our POC XML and XSLT files. After opening the file in the software the following details of the exception is as follow:

```
 (d24.df0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=4012d118 ebx=4835de58 ecx=00000000 edx=0012f970 esi=0012faac edi=08230d40
eip=10497a1b esp=0012f9fc ebp=0012fa24 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000            efl=00010202
xul!txNodeSorter::compareNodes+0x88:
10497a1b 833b00          cmp     dword ptr [ebx],0    ds:0023:4835de58=????????
```