FORTCONSULT
Straight talk on IT security

SECURITY ADVISORY

December 2006

Lotus Notes Port 1352 Pre-login Information Leakage
Advisory Version 1.1

Discovered in May 2006 by the FortConsult A/SSecurity Research Team/Andrew Christensen

Table of Contents

Copyright and Disclaimer

It is provided so that our customers and others understand the risk they may be facing by running affected software on their systems.

In case you wish to copy information from this advisory, you must either copy all of it or refer to this document (ideally by the URL http://www.fortconsult.net/)

No guarantee is provided for the accuracy of this information, or damage you may cause your systems in testing.

The Security Research Team

This advisory has been discovered by FortConsults Security Research Team/Andrew Christensen.

FortConsult is a specialist in technical services within the field of IT security. We are vulnerability experts that help business enterprises to protect themselves against the numerous security threats that exist today – both as impartial consultants and with responsibility for specific tasks. Our primary services are security tests and practically-oriented security consultancy.

For more information: www.fortconsult.net.

Issue History

This document has been updated to the present version as information has been received from various external sources.

April 2006:

Issue discovered by FortConsults Security Research Team.

May 2006:

Issue reported to 3Com's Zero Day Initiative for indepedent verification. ZDI confirms the issue as a valid security issue, decides not to buy disclosure rights due to type of issue.

June 20th 2006:

Issue reported to security-alert@austin.ibm.com by PGP-encrypted and signed mail.

June 22nd 2006:

An IBM employee with the title of AIX Security Developer confirms receipt and decryption of mail, and states issue has been forewarded to Lotus Security team (security@notesdev.ibm.com).

June 28th 2006:

Katherine Emling from IBM's Lotus Security replies, requesting copies of Proof-of-Concept utilities, and providing information about possible workarounds to try. A reply was made to IBM on the next day, June 29th, with copies of the Proof-of-Concept utilities.

July 27th 2006:

IBM replies with more detailed analysis of issue, description of how roaming ID files are downloaded versus how initial ID's are downloaded. Updated version (1.1) of this advisory is created reflecting that this issue does not impact roaming users in a serious way.

November 7th 2006:

IBM releases versions of Lotus Notes containing a configuration switch that allows risky functionality to be disabled in favor of more secure key distribution.

Brief Issue Description

This issue affects servers which have the Lotus Notes port 1352 open to person testing / attacking the server.

It is possible to retrieve unencrypted data from the ”names.nsf” database (or encrypted, and therefore relevatively useless, data stored in roaming users' personal databases) on Lotus Notes servers, without being logged in:

- It is always possible to validate usernames that do / do not exist.
- It is possible to download new user's User.ID file which they have not yet downloaded, and any User.ID user key files which exist in the names.nsf database, and then to perform an offline attack on them to recover the passphrase.

Examples of Specific Issues

Notes on Proof-of-Concept Tools

The following examples were made using two programs coded by FortConsults Security Research Team, which may be provided upon request. User.ID keyfiles can also be retrieved on request for proof.

- lotus_username_finder.pl, which identifies valid usernames and downloads the associated User.ID keyfile if it is in names.nsf. If the user.id is for a roaming user, and is stored in the roaming user's personal database, a User.ID file which is partially encrypted and therefore relatively useless to an attacker will be downloaded.
- lotus_proto.pl, which identifies valid view names within names.nsf, and in some cases validates data within names.nsf, before being logged in.

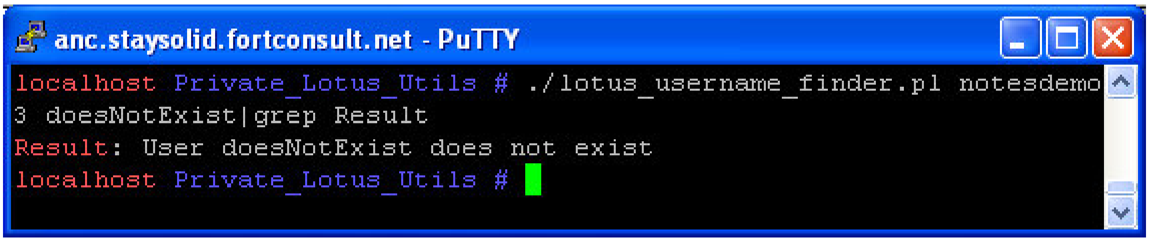
The programs are written in Perl, and should in theory work on nearly any common platform

(Windows, Linux, Solaris). They have been tested on Linux, and on Windows with ActiveState's Perl.

Username Validation

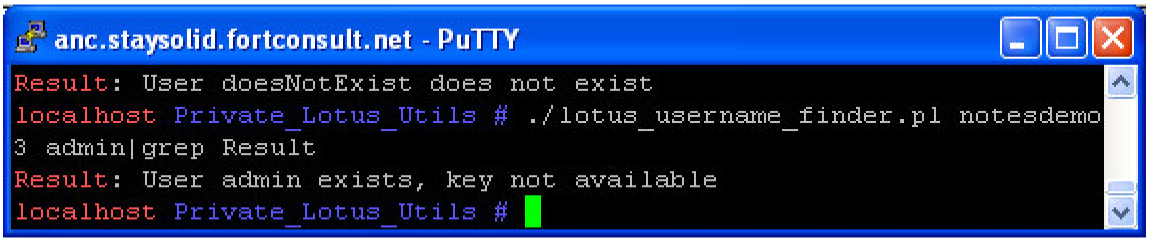
A test Lotus Notes system was used for demonstration purposes. The test system was installed using completely default settings, but in order to secure it in a sensible way as might be seen at (for example) a bank or at SWIFT, the Notes Port encryption and "Username & Passwords: less combinations for greater security" options were enabled. The standard "No" setting of "Allow anonymous Notes connections" was left untouched, as that is already the most secure setting by default.

First, it is possible to validate that the user DoesNotExist is not present on this system:



```
localhost Private_Lotus_Utils # ./lotus_username_finder.pl notesdemo
3 doesNotExist|grep Result
Result: User doesNotExist does not exist
localhost Private_Lotus_Utils #
```

Next, it was possible to validate the the user Admin is present on this system:



```
Result: User doesNotExist does not exist
localhost Private_Lotus_Utils # ./lotus_username_finder.pl notesdemo
3 admin|grep Result
Result: User admin exists, key not available
localhost Private_Lotus_Utils #
```

Naturally, this can be scripted in order to rapidly build a list of valid usernames on the system, which can then be fed into Hydra or Brutus in an online bruteforce attack on (for example) http://notesserver/names.nsf or http://notesserver/mail.nsf.

User key User.ID Download

Why this is an Issue

Many very security minded organizations consider the Lotus Notes key handling system to be secure. For example, in a post visible at http://www.derkeiler.com/Mailing-Lists/securityfocus/security-basics/2002-08/0242.html, an employee of SWIFT (the global electronic payment transfer network) states that:

”Notes is a true client server application, there is little you can do if you connect to TCP 1352 without using a notes client/server. The actual protocol is proprietary but similar to SSL in many aspects. I'd view the security requirements as similar to that. The only direct risk of having TCP 1352 open is DoS, but this is the same for any open port. Mitigation is by restricting which machines have access to each other.”

We don’t mean to hang SWIFT out to dry – the post above is (of course) now 4 years old, and even today doesn't seem completely inaccurate. However, a lot of companies and governments seem to share this view that Lotus Notes provides perfect key handling infrastructure, and have left port 1352 open to the Internet. This results in attackers being able to download key files from many places.

The keyfiles, of course, have passphrases on them. These can range from blank passphrases, up to the unbreakably long (barring any crypto backdoors in Lotus Notes, of course).

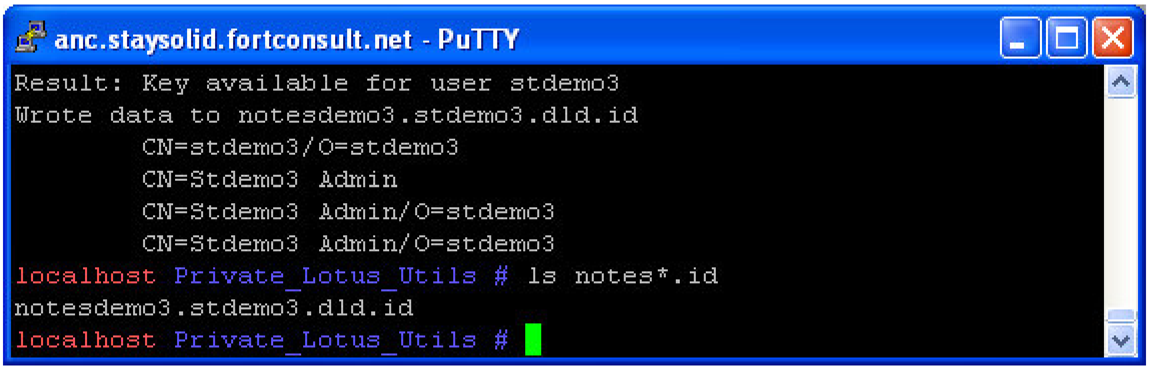
While the encryption used to protect User.ID keyfiles is (according to best available industry knowlege) quite good, it is possible to test approximately 40,000 passphrases per second using a standard Dell P3 machine and commercially-available key-cracking software.

This means that with a single machine, you can test:

- all-numeric phrases up to 8 characters in an hour or so
- alpha-numeric phrases up to 8 characters in a month or so.
- More complex phrases involving alphanumerics and punctuation / European characters (å, æ, ä, ü, etc.) within a reasonable period of time, especially if you build a distributed cracking framework.

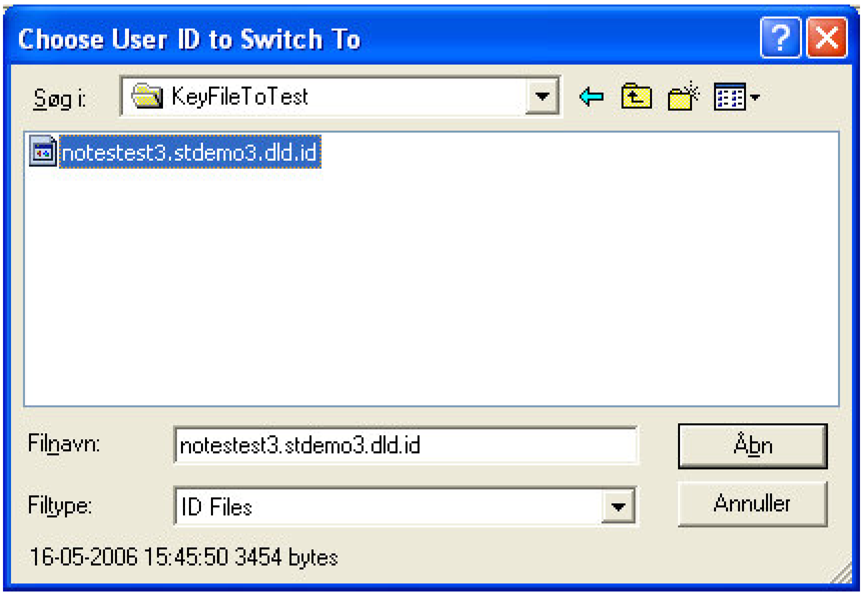
Demonstration of the Issue

An account where the key file is present was located, and the key was downloaded:



```
Result: Key available for user stdemo3
Wrote data to notesdemo3.stdemo3.dld.id
        CN=stdemo3/O=stdemo3
        CN=Stdemo3 Admin
        CN=Stdemo3 Admin/O=stdemo3
        CN=Stdemo3 Admin/O=stdemo3
localhost Private_Lotus_Utils # ls notes*.id
notesdemo3.stdemo3.dld.id
localhost Private_Lotus_Utils #
```

This file could be verified as legitimate by attempting to switch to this User Key in a Lotus Notes client installation:



Søg i: KeyFileToTest

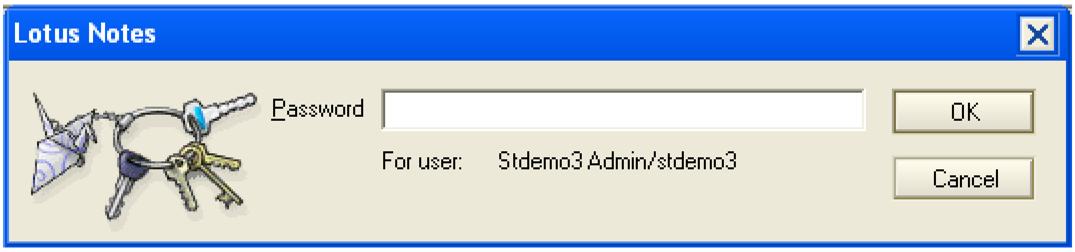
notestest3.stdemo3.dld.id

Filnavn: notestest3.stdemo3.dld.id

Filtype: ID Files

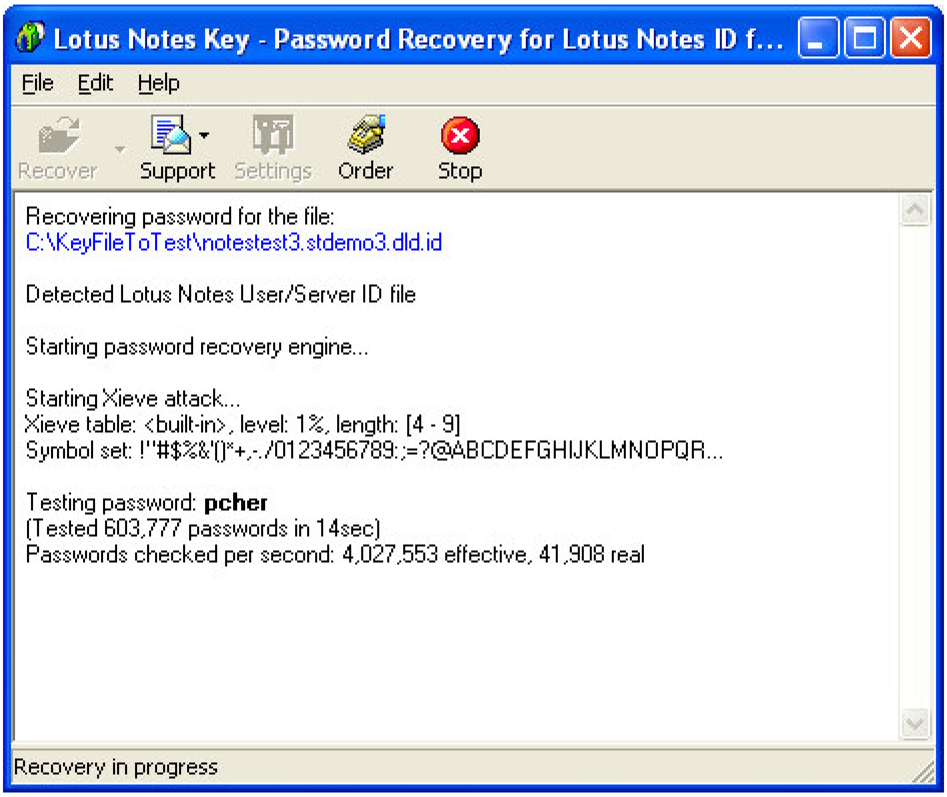
16-05-2006 15:45:50 3454 bytes

Here, we could manually enter the password if guessed or known:



For user: Stdemo3 Admin/stdemo3

As it's not a lot of fun to manually type in billions of passwords, you can at this point elect to use one of the several key passphrase crackers available both commercially and for free (a crippled demo version of the Lotus Notes keycracker from http://www.lostpassword.com/lotus-notes.htm is shown below):



```
Recovering password for the file:
C:\KeyFileToTest\notestest3.stdemo3.dld.id

Detected Lotus Notes User/Server ID file

Starting password recovery engine...

Starting Xieve attack...
Xieve table: <built-in>, level: 1%, length: [4 - 9]
Symbol set: !"#$%&'()*+,-./0123456789:;=?@ABCDEFGHIJKLMNOPQR...

Testing password: pcher
(Tested 603,777 passwords in 14sec)
Passwords checked per second: 4,027,553 effective, 41,908 real
```

Recovery in progress

Mitigation

For the new versions of Lotus Notes (6.5.5 FP2 and 7.0.2), it is now possible to set a configuration setting in the notes.ini file to disable unauthenticated ID file retrieval.

If you are unable to upgrade to a version where this can be configured, alternate mitigation strategies for this issue are:

- Running Notes behind a VPN / firewall; do not allow port 1352 to be visible on the Internet, and / or:
- Ensuring that no User.ID files are visible within names.nsf for extended periods of time and / or:
- At the end, this comes down to the age old issue of password quality. This can be avoided by ensuring that very strong passphrases are required for User.ID files. Alternately, Lotus Domino has features for smartcard-based auth which could probably be employed to avoid the issue of weak passwords completely.