Future Directions for the AFS Client on Windows

Jeffrey Eric Altman

jaltman *at* secure-endpoints *dot* com

Why am I here at SLAC?

- Introduce myself to the community
- Describe the state of OpenAFS on Windows today
- Describe the issues which must be solved
- Offer proposals for future directions
- Obtain your feedback

Who am I and what do I do?

- OpenAFS Gatekeeper for Windows
  - Audit code submissions
  - Manage Bug Requests
  - Build releases
  - Fix things
  - Plan for the future
- MIT Kerberos for Windows maintainer
- Project JXTA Board Member

What else have I done?

- The Kermit Project
  - Cross platform (Unix, OS/2, Windows)
- Internet Access Methods
  - Java based Person to Person collaboration software
- Miscellaneous Network Security stuff
  - OpenSSL, Secure Remote Password, TELNET START_TLS, FTP AUTH TLS, SSH
- Internet Engineering Task Force (IETF)

I have no AFS experience
Why am I a Gatekeeper?

- Windows development background
- Networking experience
- Security experience
- Reputation from other projects
- Volunteer

How bad things were ...

- OpenAFS on Windows was under supported
- Other than the work added in 1.2.8 there have been close to zero changes since 1.0
- Submitted patches could not be applied as there was no one to audit them
- Bugs placed in RT could not be responded to.

securendpoints

There is a new sheriff in town

- All items in RT queue have at least been responded to if not fixed
- Outstanding patches have been applied
- Code submissions obtained and integrated
- Resource leaks plugged
- “Stable” OpenAFS 1.3.61 announced March 22

1.2.11 vs. 1.3.61: Which definition of "stable" do we mean?

1. "Stable" meaning that the code does not change very much from release to release providing predictability
2. "Stable" meaning that the code performs reliably without crashing unexpectedly or adversely impacting the performance of the system

Reasons 1.2.x is Not a Stable Release

- Un-initialized variables
- Memory leaks due to reference count management errors
- Kernel object leaks due to reference count and usage errors
- Thread deadlocks due to recursive use of single use lock implementation

More reasons 1.2.x is Not Stable

- Memory allocated in one DLL is de-allocated in another
- Operations which require both a pioctl and a RPC to send private data (ktc_GetToken and ktc_SetToken) are not atomic

Even more reasons ...

- The number of NetBIOS control blocks used in protocol operations (100) exceeds the number of objects which Windows can wait on simultaneously (64).
- SMB messages with the “extended” bit set were not supported preventing file operations from being performed on a subset of files.

What is new in 1.3.61?

- Code Donations from:
  - Rob Murawski
  - Joe Beuhler
  - MIT
  - Morgan Stanley
  - Secure Endpoints
  - Sine Nomine
  - Skyrope
  - others
- New functionality
- Improved Performance
- Improved Reliability
- New Installer
- Improved Developer experience

New Build System

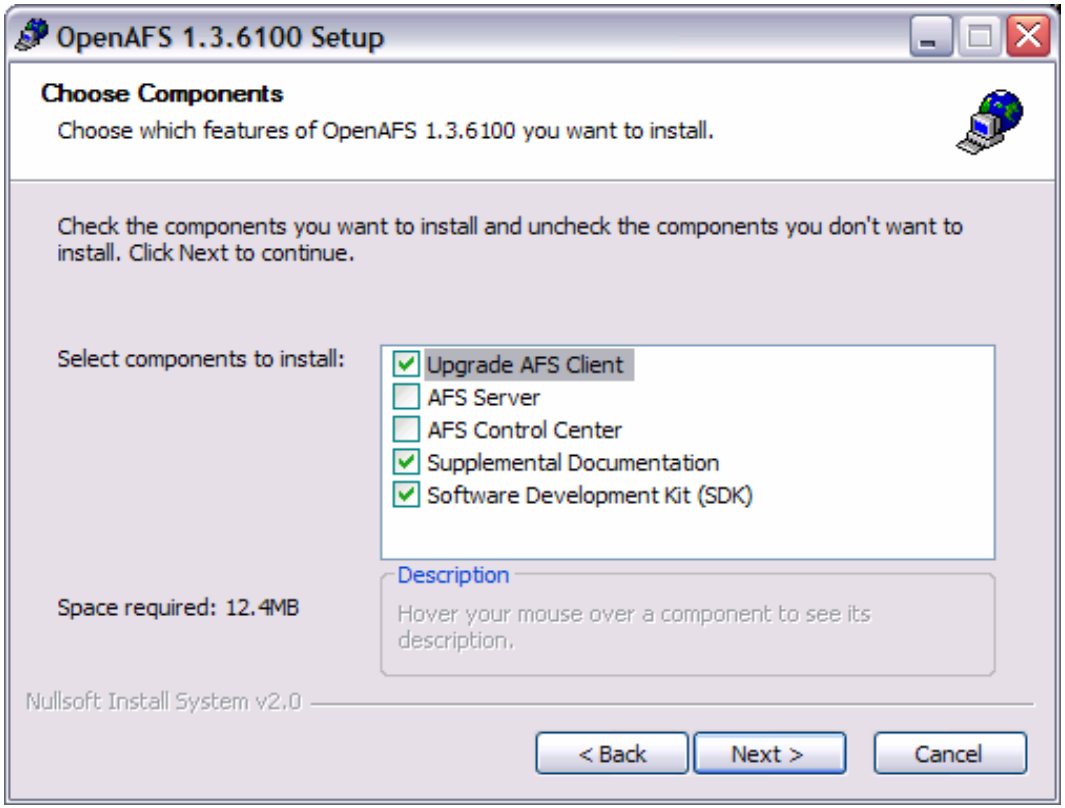
- Supports
  - Microsoft Visual C++ 6.0;
  - Visual Studio .NET; and
  - Visual Studio .NET 2003 (release builds)
- Only Windows 2000 and above
- Windows 9X did not compile and there is no desire to fix it.

New NSIS Installer

- Rob Murawski implemented a new installer using the Open Source Nullsoft Scriptable Installer Framework 2.0
- Supports new installs, uninstalls and upgrades from previous releases
- Designed for interactive installs (not an MSI)

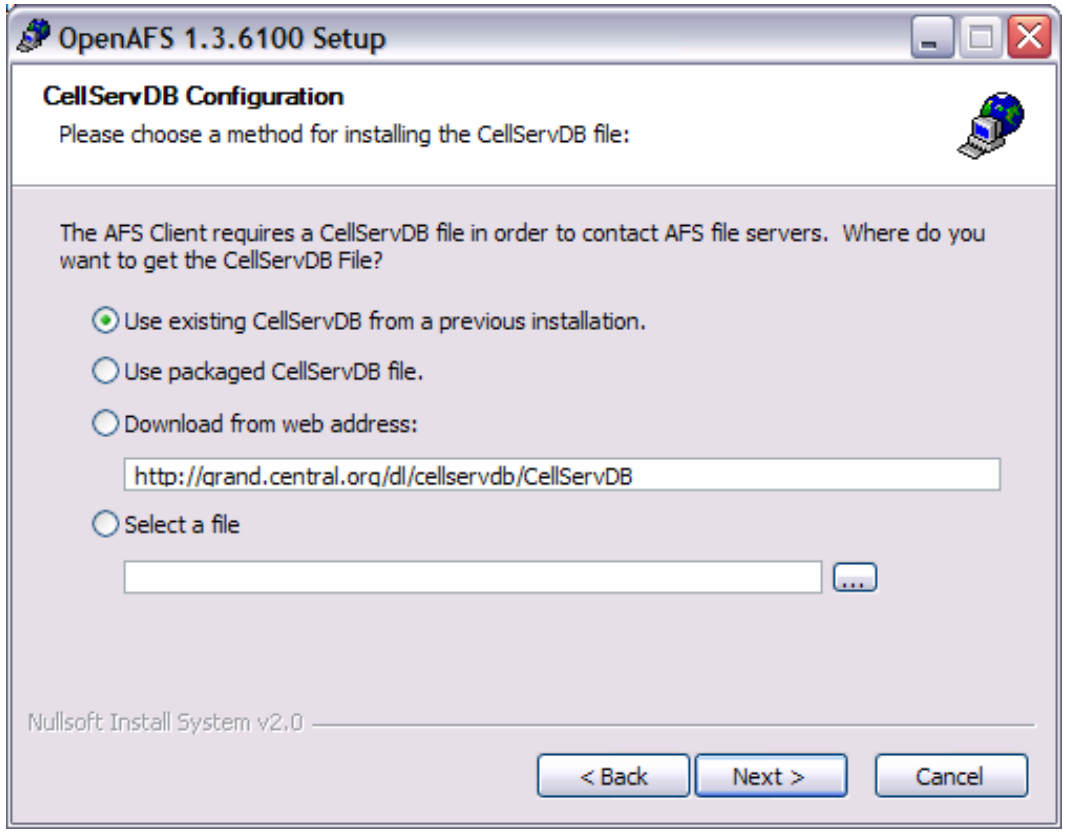
securendpoints

NSIS Installer: Selecting Components

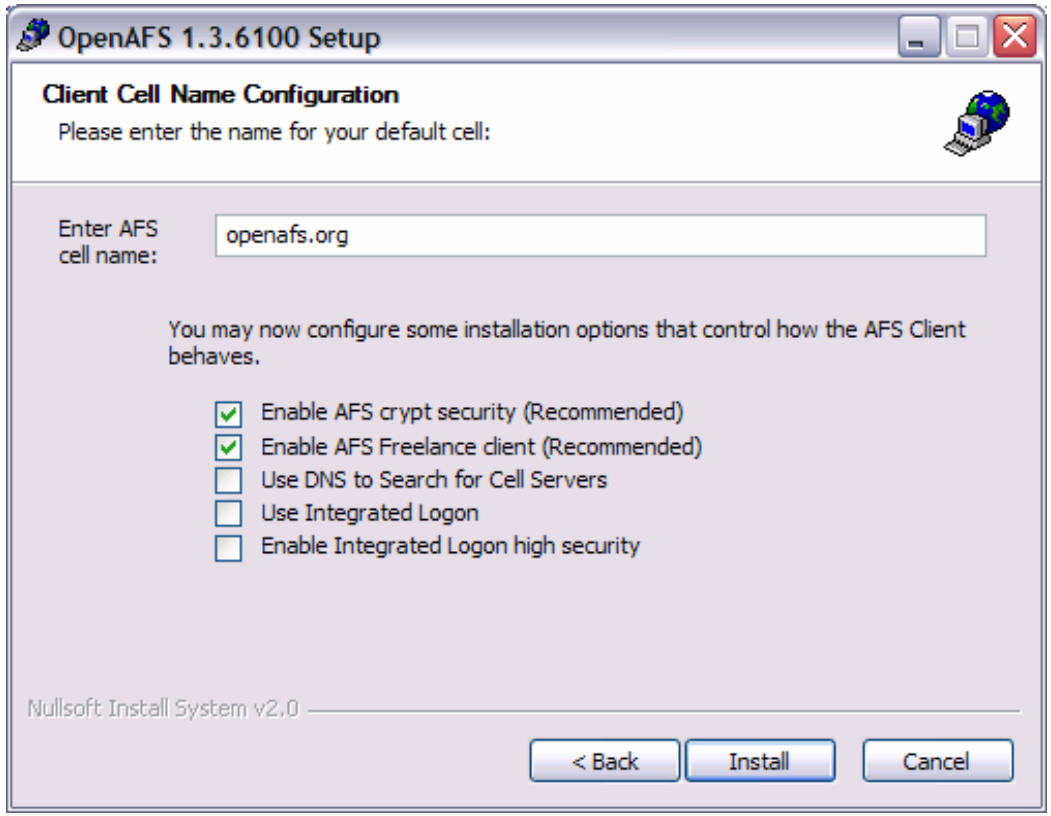


securendpoints

NSIS Installer: CellServDB



NSIS Installer: Client Configuration



securendpoints

\\afs\cellname\

- UNC paths of the form \\afs\cellname are now supported when using the MS Loopback adapter
- The “NetbiosName” registry value can be used to specify alternatives to “afs”
- No longer need to use \\afs\all\cellname

MIT Kerberos for Windows 2.6 Integration

- Obtain tokens using Kerberos 5 and krb524d
- Imports credentials from both the MSLSA and CCAPI credential caches
- Automatically renews tokens and tickets as they approach expiration
- Architecture supports obtaining tokens for multiple cells from a single krb5 tgt (no UI)
- Not yet supported by Integrated Logon
- Can be disabled on a per user basis (no UI)

Using DNS to resolve Cells (not new just not used)

- Cells not specified in the %WINDIR%\afsdcell.ini (aka CellServDB) may be discovered via DNS
- Windows DNS Query API now used instead of home grown implementation
- No longer a need to configure DNS servers with %WINDIR%\afsdns.ini
- Controlled by “UseDNS” registry value

Freelance mode (not new just not used)

- No need for a home cell to provide mount points for other cells
- Dynamically mounts cells upon first use
- Stores local mount points in %WINDIR%\afs_freelance.ini
- “fs mkmount” and “fs rmmount” may be used to configure mount lists
- Controlled by “FreelanceClient” registry value
- Provides for better disconnected user experience

Select Lan Adapter by Name

- The display name of the LAN Adapters can be used as a means of specifying which LAN adapter should be used by the AFS Client Service.
- Simply name the desired LAN Adapter “AFS”
- This functionality may be disabled using the “NoFindLanaByName” registry value
- This functionality is disable by default by the 1.3.61 NSIS installer.

Hidden Dot Files

- Following Unix tradition, files/directories whose names begin with a period are given the Hidden attribute when the “HideDotFiles” registry value is set

Power Management Support

- Automatic Flushing of Volume data upon receipt of Standby or Suspend Notifications

securendpoints

Compatibility with Cisco IPSec VPN Client

- The maximum size of Rx packets must be kept no larger than 1292 bytes in order to pass through the Cisco IPSec VPN Client
- Installer sets the “RxMaxMTU” registry value to 1260 to provide compatibility

securendpoints

Logging Changes

- afsd_init.log and afsd.log moved to the %TEMP% directory (usually %WINDIR%\TEMP for the SYSTEM account)
- Stack Trace data logged to afsd_init.log during assertion failure or unhandled exception

The Beginning of Per User Profile Information

- HKLM\Software\OpenAFS\Client key used to set system default values
- HKCU\Software\OpenAFS\Client key used to store user configuration data
- Currently used for:
  - Token Expiration Reminders
  - Use of Kerberos for Windows
  - Show Tray Icon (afscreds.exe auto start)
  - afscreds.exe shortcut parameters

New afscreds.exe functionality

- -A = if needed, obtain tokens automatically using available Kerberos credentials or display an obtain token dialog to the user
- -M = renew drive mapping
- -N = activate IP Address Change monitor. If new address is discovered and no tokens are present query KDC; if found present token dialog to the user
- -Z = remove all drive mappings

Many other changes

- Performance optimizations
- Additional runtime configuration via the registry
- Added instrumentation
- Fixed “vos listaddrs” and “fs setserverprefs”
- See the release notes for details

Known Issues: Multi-user support

- “Cell” registry value serves two orthogonal purposes
  - Specifies home cell for the AFS Client Service
  - Specifies the default cell to use when obtaining tokens
- Drive mapping data is stored globally although drive maps are actually maintained by the shell per user
- Mount points are global allowing users to alter the environment for others
- Token leakage occurs when tokens are obtained via afscreds.exe, aklog.exe, or KfW’s Leash32.exe

Known Issues: AFS Client Service

- AFSD Client Service unable to handle dynamic changes to network configuration when MS Loopback Adapter is not installed
- SMB redirector overhead imposes performance restrictions
- Large File (> 2GB) support not yet implemented

Known Issues: Cache Management

- Cache is memory based resulting in a loss of cache data upon AFS Client Service shutdown
- Each cached file is stored multiple times by the system:
  - Once in the AFSCache file
  - Once in the memory mapped to the AFSCache file
- Maximum Cache size restricted by resource utilization

Known Issues: Integrated Logon

- Does not yet work with Kerberos for Windows
- Needs a better method of storing tokens on a per session basis than generating random SMB user names
- Needs a method of delayed token acquisition if the network is disconnected at logon time

Known Issues: User Interface

- afscreds.exe provides no method for obtaining tokens for multiple cells from a single Kerberos principal
- Drive mapping dialogs do not make it clear that submounts are being created
- AFS Shell Extensions do not work for UNC paths such as \\AFS\openafs.org\

Known Issues: More User Interface

- afs_config.exe is really an Administrator tool:
  - should be removed from afscreds.exe
  - control panel should be moved to Administration folder
  - access should be restricted based upon ACLs
- New registry values must be added
- When modifying UI, must support all languages not just English

securendpoints

Known Issues: Miscellaneous

- Need to synchronize with Unix sources
- No support for Named Pipes
- Need to migrate away from INI files to use of the registry
  - afsdsbmt.ini
  - afs_freelance.ini
- Storage of Unicode file names
- The AFS Server does not work!!!!
- Documentation !!!!

Limitations of the existing architecture

- Original design by Mike Kazar meant to be a prototype
- SMB redirector slowing us down
- Cache size severely restricted
- AFS volumes available late in the boot cycle

securendpoints

Using a Network IFS as the basis for an alternative architecture

- An IFS increases throughput by removing the delays imposed by the Ack/Nak SMB protocol and the overhead of protocol translation
- Once an IFS has been implemented, cache management can be performed by manipulating the contents of the Windows File System Cache instead of storing our own. Thereby reducing resource overhead.
- Use of an IFS would allow tighter integration with the Windows Security model. Tokens would be associated with user SIDs instead of SMB names.

All IFS Kits are not created equal

- There are several issues to be concerned with when selecting an IFS Kit upon which to base development:
  - Licensing requirements of the kit. The license must be compatible with the OpenAFS.ORG license used for the existing code base.
  - Compatibility with existing and future releases of Microsoft Windows operating systems.
  - Cost of the kit. This higher the cost reduces the number of OpenAFS.org users who are capable of experimenting and producing binaries on their own. Fewer developers with access to the kit mean fewer developers who can contribute to the project simply by donating their time.

Comparison of Available IFS Kits

IFS Kit	License Requirements	Windows Compatibility	Retail Pricing
Microsoft IFS Kit	Royalty free. 60 day notice to Microsoft before shipping binaries. All binaries must be digitally signed by Microsoft.	API known to be compatible with shipping operating systems. MS reserves the right to alter the API in future operating systems.	$995 per developer per version (no support)
GNU IFS Kit	GNU Public License Version 2	Unknown	Free (no support)
OSR File System Framework	Royalty free.	OSR will upgrade their product to maintain compatibility with future releases.	$95,000 per commercial product (includes support)

- the Microsoft IFS Kit and the GNU IFS Kit contain licensing terms which are incompatible with the existing OpenAFS.ORG license
- Binaries produced with the OSR FSF are redistributable without royalty payments. OSR has in the past provided open source projects, CODA, with a modified version of their libraries at reduced pricing and may be willing to do so for OpenAFS.ORG.

The Future of OpenAFS for Windows Rests in the Hands of the Community

- The work to be done is significant in scope
- How much can be done and in what time frame will be determined by the resources the Community can donate to the project.
- Donations can be in the form of person hours (programming, UI design, documentation) or money (support and work for hire contracts or general use grants).

securendpoints

Q & A

You ask, I answer

Contact Information

Jeffrey Eric Altman

jaltman *at* secure-endpoints *dot* com