

COLLABORATORS

	<i>TITLE :</i> SAMBA Developers Guide	<i>REFERENCE :</i>

Contents

Abstract

Last Update : Fri Oct 10 00:59:58 CEST 2003

This book is a collection of documents that might be useful for people developing samba or those interested in doing so.
GPLtbe)18(v)15(rstio18)2lte.14(Atio18)cop10wo18Th318ludresmba

Chapter 1

NetBIOS in a Unix World

Andrew Tridgell April 1995

1.1 Introduction

This is a short document that describes some of the issues that confront a SMB implementation on unix, and how Samba copes with them. They may help people who are looking at unix<->PC interoperability.

It was written to help out a person who was writing a paper on unix to PC connectivity.

1.2 Usernames

The SMB protocol has only a loose username concept. Early SMB protocols (such as CORE and COREPLUS) have no username concept at all. Even in later protocols clients often attempt operations (particularly printer operations) without first validating a username on the server.

Unix security is based around username/password pairs. A unix box should not allow clients to do any substantive operation without some sort of validation.

The problem mostly manifests itself when the unix server is in "share level" security mode. This is the default mode as the alternative "user level" security mode usually forces a client to connect to the server as the same user for each connected share, which is inconvenient in many sites.

In "share level" security the client normally gives a username in the "session setup" protocol, but does not supply an accompanying password. The client then connects to resources using the "tree connect" protocol, and supplies a password. The problem is that the user on the PC types the username and the password in different contexts, unaware that they need to go together to give access to totha9r0(serv)15(er)55(. -3056(The)-9r0(username)-290(is)-9r0(normally)-238(totha9r0oney)-238(totha9r0(user)-238(typdr)-fokgaropns m. The password is the they wten connecting to the printer.

The usee oftenotmaly difference connection.-496(Ooften)-212(the)15(y)-212(Isos)-2-1(w)10(ent)3212(to)-212(access)]TJ 0 -11.955 Td [(username the serecee ewitper the use=e

132sign

This brings up all sorts of interesting problems. For example, when you copy a file to a unix drive, and the file is world writeable but owned by another user the file will transfer correctly but will receive the wrong date. This is because the `utime()` call under unix only succeeds for the owner of the file, or root, even if the file is world writeable. For security reasons Samba does all file operations as the validated user, not root, so the `utime()` fails. This can stuff up shared development directories as programs like

1.7 Trapdoor UIDs

A SMB session can run with several uids on the one socket. This happens when a user connects to two shares with different usernames. To cope with this the unix server needs to switch uids within the one process. On some unixes (such as SCO) this is

Chapter 2

NT Domain RPC's

Luke Leighton Paul Ashton Duncan Stansfield 01 November 97(version 0.0.24)

2.1 Introduction

This document contains information to provide an NT workstation with login services, without the need for an NT server. It is the sgml version of <http://mailhost.cb1.com/~lkcl/cifsntdomain.txt>, controlled by Luke.

It should be possible to select a domain instead of a workgroup (in the NT workstation's TCP/IP settings) and after the oblig-

2.2.2 Enumerations

2.2.2.1 MSRPC Header type

command number in the msrpc packet header

MSRPC_Request: 0x00

MSRPC_Response: 0x02

MSRPC_Bind: 0x0B

MSRPC_BindAck: 0x0C

2.2.2.2 MSRPC Packet info

The meaning of these flags is undocumented

FirstFrag: 0x01

LastFrag: 0x02

NotaFrag: 0x04

RecRespond: 0x08

NoMultiplex: 0x10

NotForIdemp: 0x20

NotforBcast: 0x40

NoUuid: 0x80

2.2.3 Structures

2.2.3.1 VOID *

sizeof VOID* is 32 bits.

2.2.3.2 char

sizeof char is 8 bits.

2.2.3.3 UTIME

UTIME is 32 bits, indicating time in seconds since 01jan1970. documented in cifs6.txt (section 3.5 page, page 30).

2.2.3.4 NTTIME

NTTIME is 64 bits. documented in cifs6.txt (section 3.5 page, page 30).

2.2.3.17 CLNT_SRV (server, client names structure)

Note: logon server name starts with two '\ ' characters and is upper case.

VOID*

2.2.3.30 SERVER_INFO_101

Note: see cifs6.txt section 6.4 - the fields described therein will be of assistance here. for example, the type listed below is the same as fServerType, which is described in 6.4.1.

SV_TYPE_WORKSTATION 0x00000001 All workstations

SV_TYPE_SERVER 0x00000002 All servers

SV_TYPE_SQLSERVER 0x00000004 Any server running with SQL server

SV_TYPE_DOMAIN_CTRL 0x00000008 Primary domain controller

SV_TYPE_DOMAIN_BAKCTRL 0x00000010 Backup domain controller

SV_TYPE_TIME_SOURCE 0x00000020 Server running the timesource service

SV_TYPE_AFP 0x00000040 Apple File Protocol servers

SV_TYPE_NOVELL 0x00000080 Novell servers

SV_TYPE_DOMAIN_MEMBER 0x00000100 Domain Member

SV_TYPE_PRINTQ_SERVER 0x00000200 Server sharing print queue

SV_TYPE_DIALIN_SERVER 0x00000400 Server running dialin service.

SV_TYPE_XENIX_SERVER 0x00000800 Xenix server

SV_TYPE_NT 0x00001000 NT server

SV_TYPE_WFW 0x00002000 Server running Windows for

SV_TYPE_SERVER_NT 0x00008000 Windows NT non DC server

2.3.2.1

2.3.2.6 RPC_ResBind RW

the response to place after the header in the reply packet

Part II

Samba Basics

Chapter 3

Samba Architecture

Dan Shearer November 1997

3.1 Introduction

4.5.2 `dbghdr()`

This is the function that writes a debug message header. Headers are not processed via the format buffer. Also note that if the format buffer is not empty, a call to `dbghdr()` will not produce any output. See the comments in `dbghdr()` for more info.

It is not likely that this function will be called directly. It is used by `DEBUG()` and `DEBUGADD()`.

4.5.3 `format_debug_text()`

You may also call the `pull_ascii/pull_ucs2` or `push_ascii/push_ucs2` functions if you know that a particular string is ascii or unicode. There are also a number of other convenience functions in `chrcnv.c` that call the `pull_/push_` functions with particularly common arguments, such as `pull_ascii_pstring()`

5. The biggest thing to remember is that internal (unix) strings in Samba may now contain multi-byte characters. This means you cannot assume that characters are always 1 byte long. Often this means that you will have to convert strings to ucs2 and back again in order to do some (seemingly) simple task. For examples of how to do this see functions like `strchr_m()`. I know this is very slow, and we will eventually speed it up but right now we want this stuff correct not fast.
6. all `lp_` functions now return unix strings. The magic "DOS" flag on parameters is gone.
7. all `vfs` functions take unix strings. Don't convert when passing to them

5.3

Chapter 6

Coding Suggestions

Steve French Simo Sorce Andrew Bartlett Tim Potter Martin Pool So you want to add code to Samba ...

One of the daunting tasks facing a programmer attempting to write code for Samba is understanding the various coding conventions used by those most active in the project. These conventions were mostly unwritten and helped improve either the portability, stability or consistency of the code. This document will attempt to document a few of the more important coding practices used at this time on the Samba project. The coding practices are expected to change slightly over time, and even to grow

a 0 0 Td8oreano32237s15(xpet)1g

Chapter 7

Contributing code

Jelmer R. Vernooij Here are a few tips and notes that might be useful if you are interested in modifying samba source code and getting it into samba's main branch.

Retrieving the source In order to contribute code to samba, make sure you have the latest source. Retrieving the samba source code from CVS is documented in the appendix of the Samba HOWTO Collection.

Discuss large modifications with team members Please discuss large modifications you are going to make with members of

Chapter 9

RPC Pluggable Modules

Anthony Liguori Jelmer Vernooij January 2003

9.1 About

Chapter 10

VFS Modules

Alexander Bokovoy Stefan Metzmacher 27 May 2003

10.1 The Samba (Posix) VFS layer

While most of Samba deployments are done using POSIX-compatible operating systems, there is clearly more to a file system than what is required by POSIX when it comes to adopting semantics of NT file system. Since Samba 2.2 all file-system related operations go through an abstraction layer for virtual file system (VFS) that is modelled after both POSIX and additional functions needed to transform NTFS semantics.

...

Part IV

Debugging and tracing

15.3 Print Queue TDB's

Samba provides periodic caching of the output from the "lpq command" for performance reasons. This cache time is configurable

Part V

Appendices
