

Security and Integrity in Linux Filesystems.



Alejandro Sanchez Acosta
asanchez@gnu.org

Linux Introduction.

Introduction.

- What's free software?
- What is the Linux kernel?
- A little story about the Linux kernel.
- Architecture portability.
- Linux maintainment.
- General ideas.

Filesystem Sources.

Filesystem sources.

- Fs/
- Superblock, aio, acls, file, file_table, inode, attr, quota..
- Binfmt*
- Adfs, affs, afs, autofs, befs, bfs, coda, cramfs, ramfs, devfs, devpts, hfs, hpfs, qnx4, umsdos, vfat, xfs, jfs, isofs, hugetlbfs, minix, romfs,

Filesystem Introduction.

Filesystem Introduction.

- What's a Filesystem?
- Management with VFS layer.
- Proc fs.
- Sysfs.
- Relayfs.
- Udev y hotplugging.

What's a Filesystem?

- A place to storage data on disk.
- Superblock.
- Inodes.
- Directory entries.
- Files.

Filesystem Form with VFS

- Superblock and sb_ops.
- Inode and inode_ops.
- File and file_operations.
- Register_filesystem
- Mounting a filesystem
- Accessing data filesystem via defined syscalls.

More Known Filesystems.

- ext2/ext3
- Jfs
- Reiser3 y reiser4.
- XFS
- NTFS
- UDF
- Distributed filesystems: NFS, Coda, SMB, AFS.

The future of filesystems.

- More oriented-object or more oo.
- Modularity via plugins.
- Faster searching data.
- Encryption and compression support.
- More robusted used algorithms.
- Better storage.

Reiserfs4 Overview.

Basic semantics.

- Files.
- Names and objects.
- Namespaces and interfaces.
- Directories.
- Security attributes.

Trees concepts.

- Set of nodes.
- Fanout.
- Finited and infinited trees.
- Keys to identify objects.
- Node structure.
- Items structure.

Trees design.

- Height or space balanced.
- B and b+ trees.
- Htrees.
- Positional trees.
- Dancing trees.
- Cache design.

Nodes.

- Identified by a key.
- Formatted and unformatted.
- Leaf and twig nodes.
- Items: nodes collection to storage data.
- Units: data that we put in the whole item.

Storing Data.

- Graphs and dancing trees.
- Separate layers: semantic and storage.
- BLOB's and extents.

Atomic filesystem

- Brief history about fs crashing.
- Filesystem checkers.
- Reducing the damage with atomic op.
- Journalled location.
- Committing allocation.

Repacker.

- 80% remain unchanged on disk.
- Ordering the tree.
- Sort the tree and pack perfectly.
- Eliminates possible fragmentation.

Journaling.

- Location on disk: journal/log.
- Committed area.
- Problem: twice write data.
- Metadata journaling.
- Solution: Wandering logging.
- Committing and transactional layer.
- Copy-on-capture and steal-on-capture.

Distributed Filesystem.

WAFL.

- Distributed Filesystem
- Used in network appliances.
- Snapshots.
- Copy-on-write.
- Large files, NFS, high performance and a quickly restart.

Plugins design.

- File, directory and hash.
- Security.
- Item
- Key assignment.
- Node and item search.
- Still not dinamically loaded.

Reiser future.

- Cryptography and compression.
- Quotas support.
- Dynamic plugins.
- Distributed filesystem.
- Encryption on commit.

Seguridad en sistemas de
ficheros.

Basic Polices.

- Credentials.
- Capabilities.
- ACL's
- Attributes.
- Metadata.

Security in filesystems.

- Filesystem and swap crypto.
- CryptoAPI support.
- LSM hooks for the file access.
- File capabilities.

CryptoAPI.

- Criptografía en kernel space.
- Uso de scatterlists.
- Implementación de criptografía de clave privada y hashing (ciphers y digests)
- Ejemplos: MD4, MD5, DES, AES, Blowfish, Twofish, ..
- Patent-free (IDEA en el 2011? :-) y estandarizados.
- Necesidad por ipv6, packet encryption.
- Firma de módulos.

```
#include <linux/crypto.h>

struct scatterlist sg[2];
char result[128];
struct crypto_tfm *tfm;

tfm = crypto_alloc_tfm("md5", 0);
if (tfm == NULL)
    fail();

/* Rellenar scatterlists */

crypto_digest_init(tfm);
crypto_digest_update(tfm, &sg, 2);
crypto_digest_final(tfm, result);

crypto_free_tfm(tfm);
```

Cryptoloop.

- Inicializamos pool con dd.
- Cargar cipher.
- Losetup -e twofish /dev/loop0 /pool
- Keysize and password.
- Crear sistema de ficheros para loop.
- Montamos sobre loop.
- Desmontamos loop y filesystem.

Benchmarking.

- Contest.
- LTT.
- Linux Test Project.
- Classics benchmarks.

Linux Security Modules.

LSM.

- NSA, SELinux, SGI, Immunix y Janus.
- Capabilities.
- `sys_security` y `security_operations`.
- `register_security`
- `selinux_plug_init`
- `netfilter`.

Referencias.

- Nucleo desarrollo: listas.hispalinux.es
- Kernelnewbies-es y [kernelnewbies](http://kernelnewbies.org).
- Kerneljanitors.
- LKML.
- Posthalloween 2.5.x
- Artículos en www.lwn.net sobre Drivers Porting.
- Traducciones en es.gnu.org/~alejandro.

¿¿¿Preguntas???

Security and Integrity in Linux Filesystems.

Alejandro Sanchez Acosta

asanchez@gnu.org