

Intro To Linux Filesystems

By Xe1phix

<https://twitter.com/xe1phix>

<https://archive.org/details/@xe1phix>

<https://www.bitchute.com/channel/xe1phix/>

<https://www.youtube.com/channel/UC4rzx4VToyHJDWbAEJ5cMxQ/videos>

Who Am I?

Linux Systems Administrator (LPIC-1, Linux+)

Just passed my LPIC-201 Linux Engineer Exam

Studying:

Linux Malware Analysis

Linux Memory Forensics

Linux Kernel Hardening

<https://gitlab.com>





Talk Layout

- [+] **ZFS Filesystem:**

- **ZFS Checksums** - SHA256 Checksum Integrity Verification.
- **ZFS XATTR** – Setting and listing Posix ACLs.
- **ZFS Snapshots** - Create, List, rolling back, delete, etc.
- **ZFS Clones** - Clone an individual dataset.
- **ZFS Compression** - Compress individual datasets (lzb | gzip | gzip-N | zle).

- [+] **Btrfs Filesystem:**

- **Btrfs Creation** – Create a Btrfs partition.
- **Btrfs Compression (zlib, LZO)** – Mount a Btrfs partition, using compression.
- **Btrfs Subvolumes** - Create, delete, list, snapshot, compress, etc.
- **Btrfs Snapshots** - Create, delete, list, mount read-only, etc.
- **Btrfs Dump-Super** - Show btrfs superblock information.
- **Btrfs dump-tree** - Dump tree structures from a given device

- [+] **XFS Filesystem:**

- **XFS Repair** - Repairs corrupt or damaged XFS filesystems.
- **XFS Metadump** - Copy XFS filesystem metadata to a file.
- **XFS Restore** - Restores filesystems from dumps produced by xfsdump.

- [+] **Filesystem Tools**

- **SMARTCTL** - Self-Monitoring Analysis and Reporting Technology
- **Fdisk, SFDisk, Parted** – Listing disks, disk statistics, inode information, etc.
- **HDDTemp** – Checking the temperature of disks.

- [+] **CFP Layout:**

- <https://gitlab.com/xe1phix/ParrotSecWiki/blob/InfoSecTalk/InfoSec-CFP-Submissions/Xe1phix-IntroToLinuxFilesystems-SecDSM-2019.txt>

Talk Resources:

- Talk Overview:
 - <https://gitlab.com/xe1phix/ParrotSecWiki/raw/InfoSecTalk/InfoSec-CFP-Submissions/Xe1phix-IntroToLinuxFilesystems-CornCon-2019.txt>
- Talk Cheatsheets:
 - <https://gitlab.com/xe1phix/ParrotSecWiki/tree/InfoSecTalk/Xe1phix-InfoSec-Talk-Materials/IntroToLinuxFilesystems-CornCon-September2019/Xe1phix-IntroToLinuxFileSystems-ShellScripts+CheatSheets>
- Talk Slides:
 - <https://gitlab.com/xe1phix/ParrotSecWiki/tree/InfoSecTalk/Xe1phix-InfoSec-Talk-Materials/IntroToLinuxFilesystems-CornCon-September2019/%20Xe1phix-Linux-FileSystems-Slides>

List of Filesystems Covered:

- ZFS
(OpenZFS)
- Btrfs
- XFS

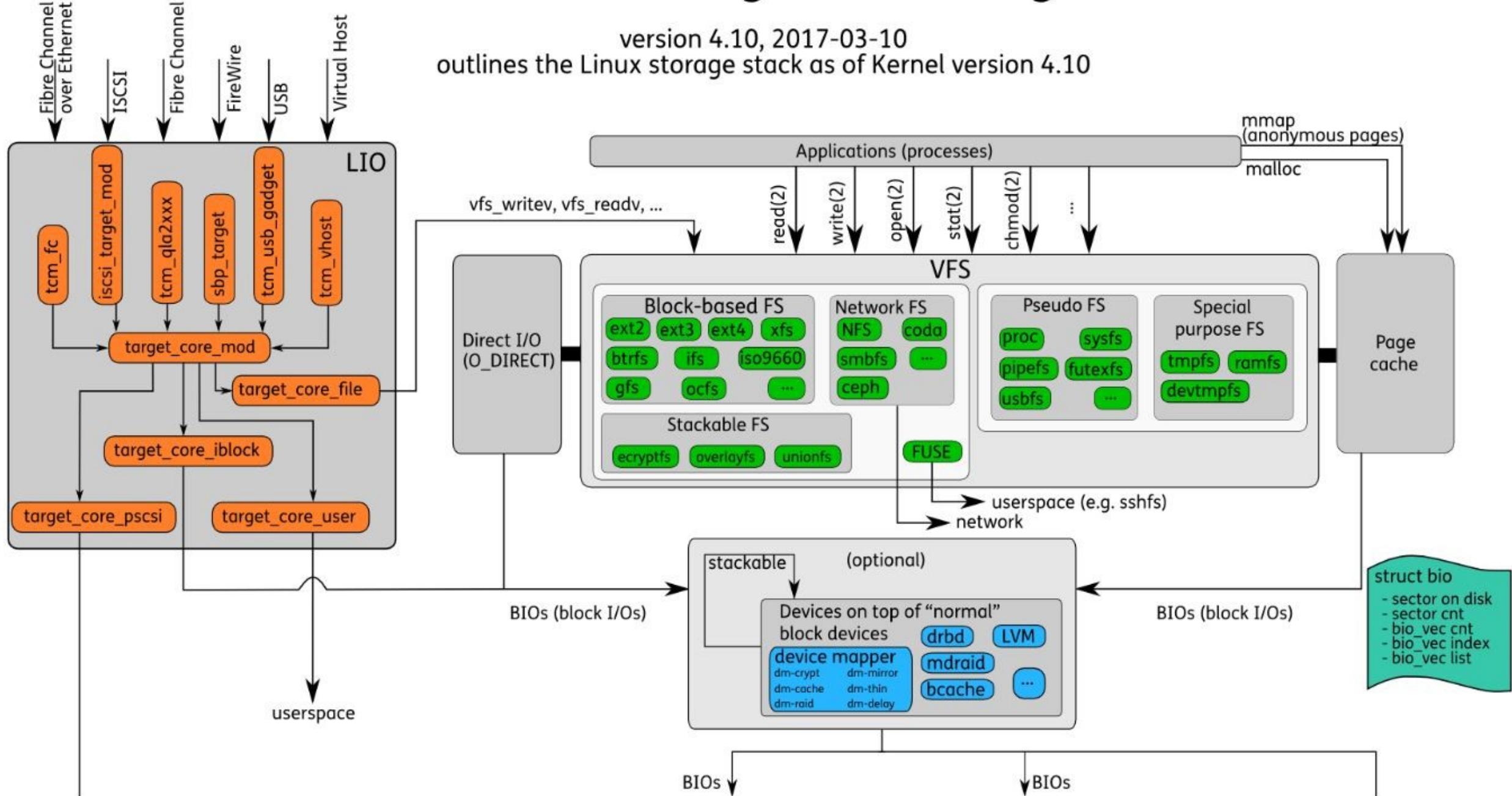
Other Filesystems Not covered:

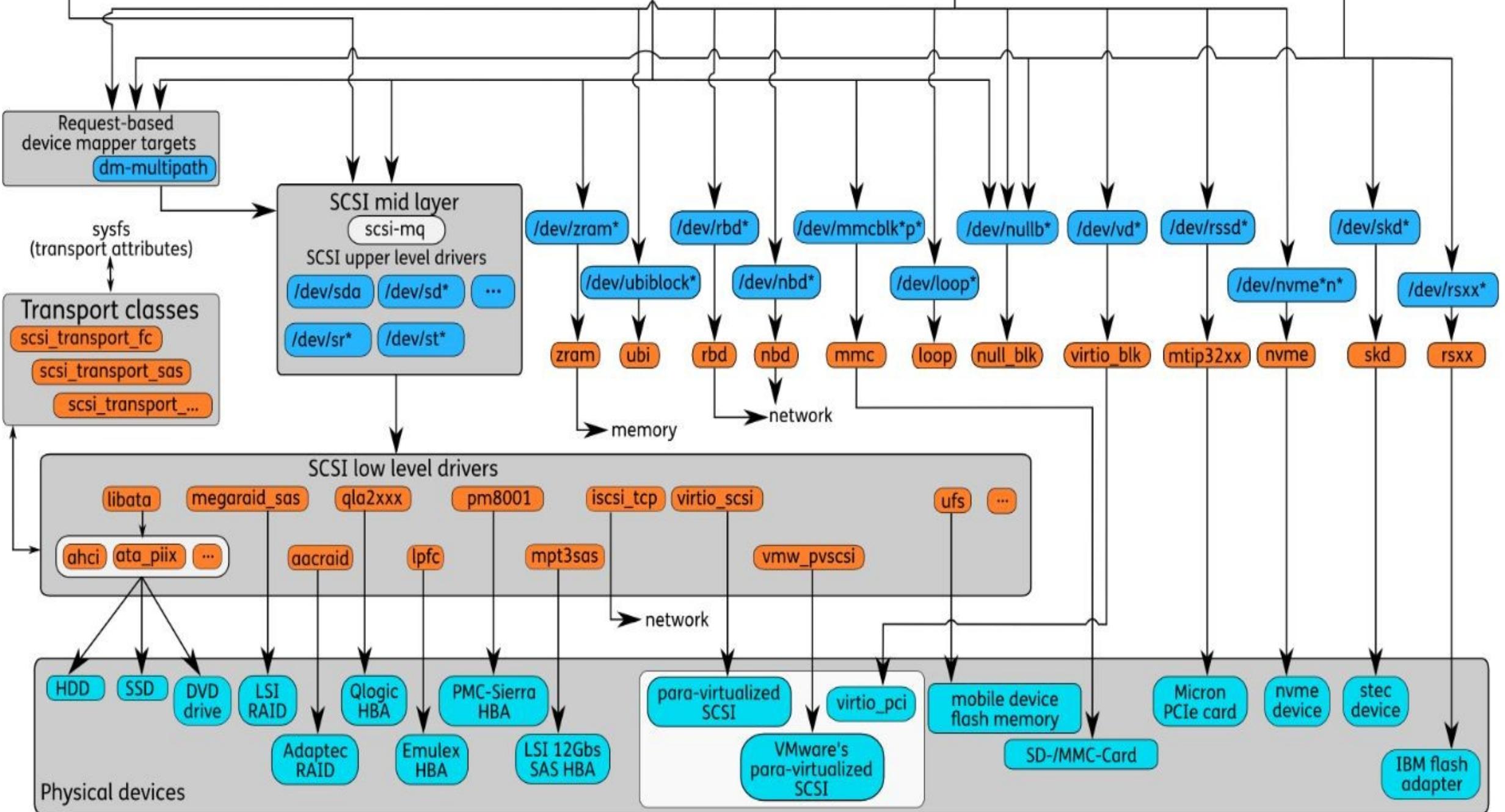
- EXT4
- LVM
- Loop Filesystems
- Fuse Filesystems
- EWF
- AFF
- eCryptFS
- Qcow2

The Linux Storage Stack Diagram

version 4.10, 2017-03-10

outlines the Linux storage stack as of Kernel version 4.10





OpenZFS FileSystem

- Xe1phix's ZFS Youtube Video:
 - <https://www.youtube.com/watch?v=hGhyIXYW9d0>
- Xe1phix's ZFS Archive.org Video:
 - https://archive.org/details/ModifyingZFSFilesystemFeaturesxattrvscanzon_edchecksumetc
- Xe1phix's ZFS Bitchute Video:
 - <https://www.bitchute.com/video/5KO3vEZBmVrC/>

<https://gitlab.com/xe1phix/Xe1phix-ZFS-Notes>

ZFS – Features 1

```
'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-#~'#
[+] ACL Inheritance
    • ACL Inheritance Flags
[+] Access Control Entries (ACEs)
    • Trivial ACLs
    • Non-Trivial ACLs
        ♦ ACL Entry Types
        ♦ ACL Access Privileges
[+] Extended Attributes
[+] ZFS User Delegating Permissions
-----
[+] Transaction Group Number
[+] Trusted Extension on Datasets
[+] Virus Scan Service
[+] ZFS Zone Restriction
[+] Temporary Mount Points
[+] Blocked Processes Execution
[+] Future Device Node Blocking
-----
[+] Read-Only Datasets Modification
[+] Enable The Set-UID Bit
'##~~-~-~-~-~-~-~-~-~-~-~-~-~-#~'#
[+] Log - ZFS Intent Log (ZFS ZIL)
    • Mirrored Log Devices
    • Separate Intent Log Devices
[+] ZFS Data Scrubbing and Resilvering
    • Automatic Repair (scrubbing)
'##~~-~-~-~-~-~-~-~-~-#~'
```

ZFS - Features 2

```
'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~##'
[+] ZFS VDEVs RAIDs:
    ▪ Striped VDEVs
    ▪ Mirrored VDEVs
    ▪ Striped Mirrored VDEVs

[+] RAID-Z Storage Pool:
    ▪ RAIDZ (4 Disks)
    ▪ Double-Parity RAID-Z | RAIDZ2 (5 Disks)
    ▪ Triple-Parity RAID-Z | RAIDZ3 (6 Disks - 3/6 Parity)
    ▪ Nested RAIDZ (8 Disks)

[+] RAID-Z Virtual Devices (Loop)

'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~##'
[+] Large ZFS Dataset Blocks | 128KB                                ## alloc
[+] ZFS - Hybrid Block Pointer | Embedded Data Feature
[+] Maximum 16 Exabyte File Size
[+] Maximum 256 Quadrillion Zettabytes
[+] Unlimited Dataset Creation

'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~##'
[+] Compression (Applied to Individual Datasets)
    ▪ LZ4      (The latest and greatest - recommended)
    ▪ gzip-N   (Where N is 1 (fastest) - 9 (best compression ratio) - not recommended)
    ▪ LZJB     (Provides a good trade-off between speed and space)

'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~##'
[+] I/O Statistics Display
[+] ZFS Admin Web Console
[+] Zpool History

'##~~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~-~##'
```

<https://gitlab.com/xelphix/Xelphix-ZFS-Notes>

ZFS - Features 3

```
'##~==~==~==~==~==~==~==~==~==~==~==~==~==~==~##'  
[+] SHA256 Checksum Integrity Verification  
[+] Deduplication  
[+] Ditto Blocks (Replicated Metadata)  
  
'##~==~==~==~==~==~==~==~==~==~==~==~==~==~##'  
[+] ZFS Snapshots

- Sending a ZFS Snapshot
- Receiving a ZFS Snapshot
- Rolling Back a ZFS Snapshot
- Snapshot User Holds
- ZFS Snapshot Diff Parser
- Rolling Snapshots
- Incremental Sending of Snapshot Streams

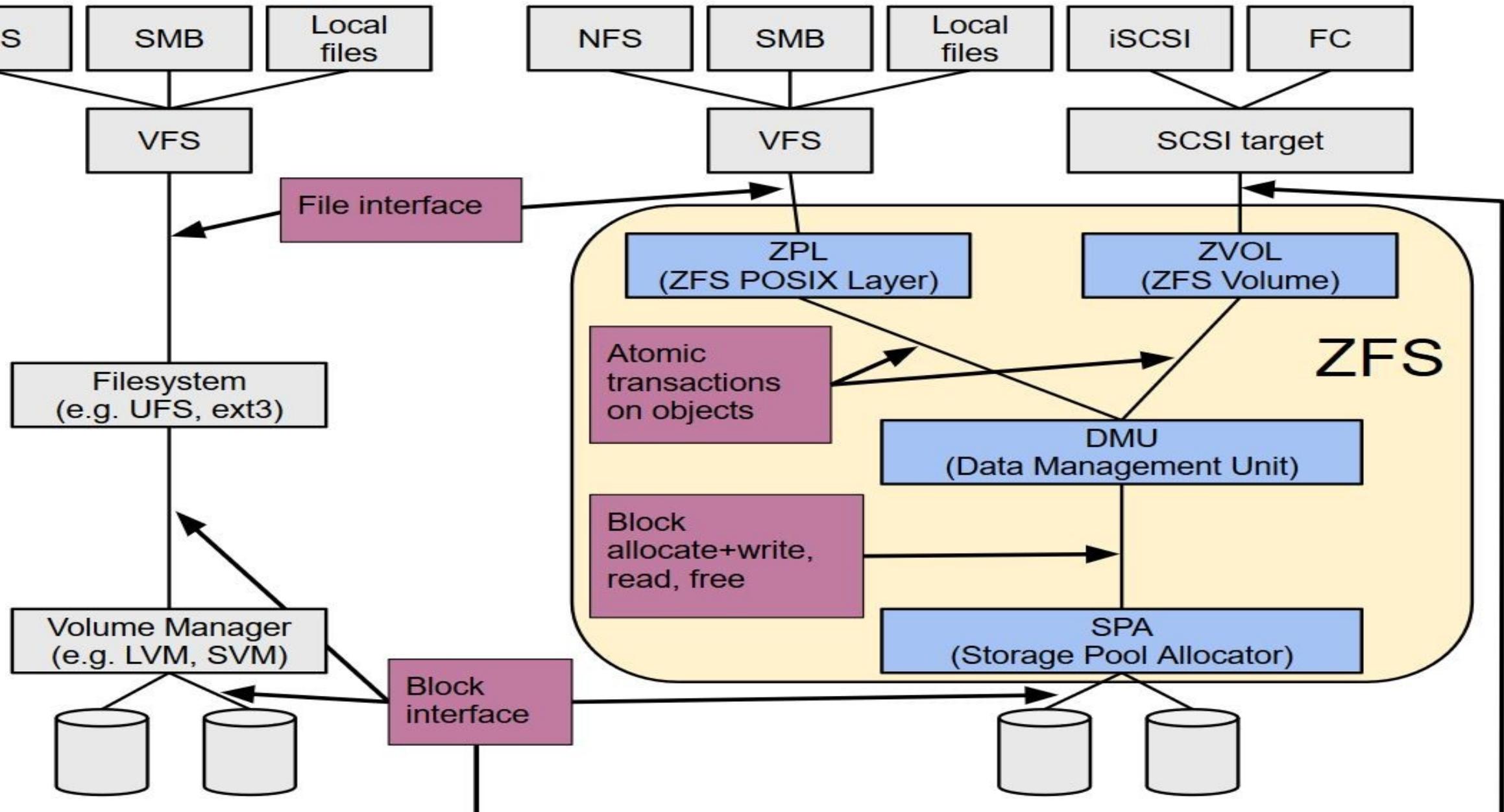


## @yesterday @today @friday @thursday @wednesday @tuesday  
## ZFS hole_birth Feature

[+] ZFS Clones  
  
'##~==~==~==~==~==~==~==~==~==~==~==~##'  
[+] ZFS Quotas (User & group)

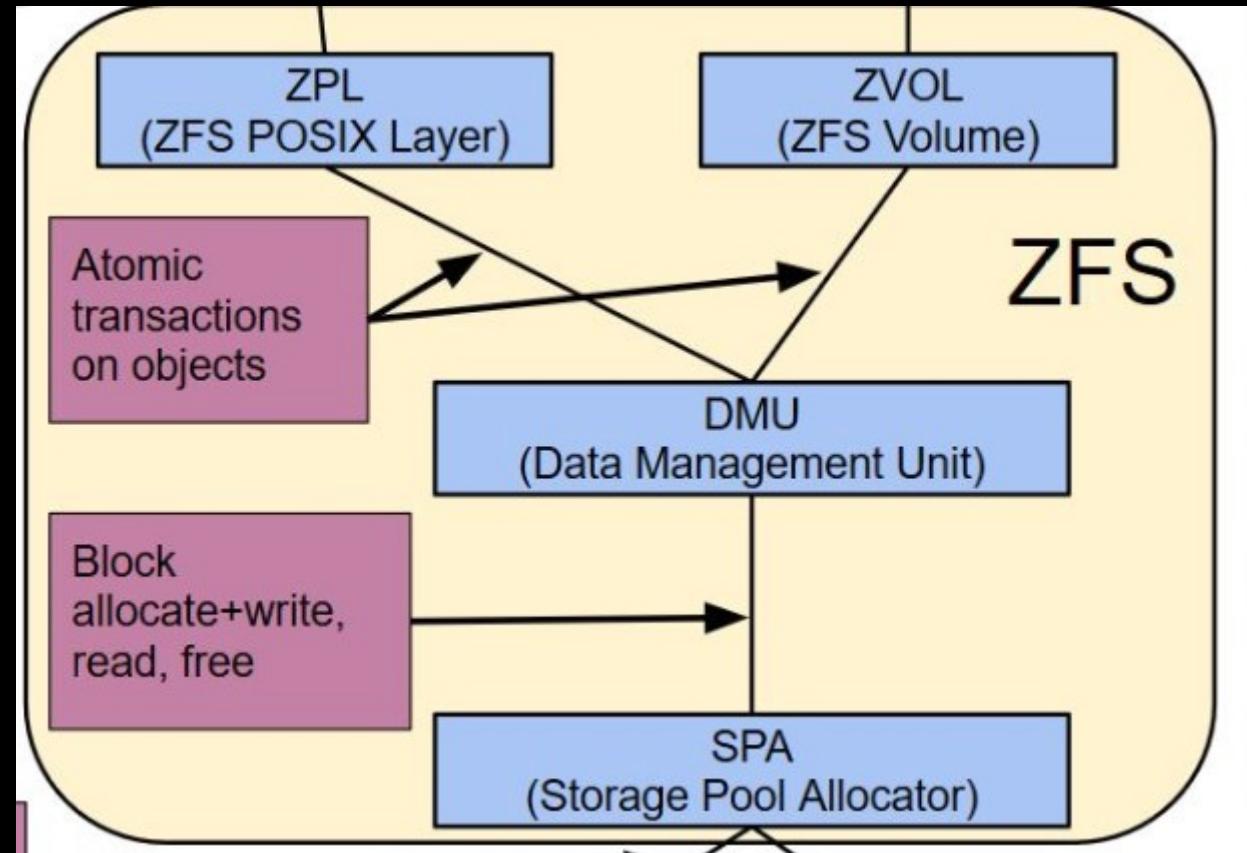
- quota=
- refquota=
- userquota=
- groupquota=

[+] Dataset Reservations  
  
'##~==~==~==~==~==~==~==~==~==~==~##'
```



ZFS – Overview of ZFS Layers

- **Storage Pool Allocator (SPA)**
 - Handles organizing the physical disks into storage.
 - Creates the block pointer
- **Data Management Unit (DMU)**
 - Uses the storage provided by the SPA by reading and writing to it transactionally in an atomic manner.
- **ZFS POSIX Layer (ZPL)**
 - Translates between operations on the filesystems and block devices (zvols) provided by the pool into operations in the DMU.



ZFS - Acronyms

- **Datasets** (Mount Points)
- **ZVols** (Block Devices)
- **Ditto Blocks** (Replicated Metadata)
- **ZLE** (Zero-Length Encoding)
- **Copy-On-Write** (CoW)
- **Adaptive Replacement Cache** (ARC)
- **Access Control Lists** (ACL)
- **ZFS Event Daemon** (ZED)
- **Layer 2 Adaptive Replacement Cache** (L2ARC)

ZFS – Definitions

ZFS File System Hierarchy

A ZFS storage pool is a logical collection of devices that provide space for datasets. A storage pool is also the root of the ZFS file system hierarchy.

The root of the pool can be accessed as a file system, such as mounting and unmounting, taking snapshots, and setting properties. The physical storage characteristics, however, are managed by the zpool(8) command.

Deduplication

Deduplication is the process for removing redundant data at the block level, reducing the total amount of data stored. If a file system has the dedup property enabled, duplicate data blocks are removed synchronously. The result is that only unique data is stored and common components are shared among files.

Snapshots

A snapshot is a read-only copy of a file system or volume. Snapshots can be created extremely quickly, and initially consume no additional space within the pool. As data within the active dataset changes, the snapshot consumes more data than would otherwise be shared with the active dataset.

Clones

A clone is a writable volume or file system whose initial contents are the same as another dataset. As with snapshots, creating a clone is nearly instantaneous, and initially consumes no additional space.

ZFS – VDev Types

There are seven types of VDEVs in ZFS:

1. disk (default)- The physical hard drives in your system.
2. file- The absolute path of pre-allocated files/images.
3. mirror- Standard software RAID-1 mirror.
4. raidz1/2/3- Non-standard distributed parity-based software RAID levels.
5. spare- Hard drives marked as a "hot spare" for ZFS software RAID.
6. cache- Device used for a level 2 adaptive read cache (L2ARC).
7. log- A separate log (SLOG) called the "ZFS Intent Log" or ZIL.

ZFS – The Self Healing Filesystem

ZFS was designed to be a self-healing filesystem. As ZFS writes data, it creates a checksum for each disk block it writes. As ZFS reads data, it validates the checksum for each disk block it reads. Media errors or “bit rot” can cause data to change, and the checksum no longer matches. When ZFS identifies a disk block checksum error on a pool that is mirrored or uses RAIDZ, it replaces the corrupted data with the correct data. Since some disk blocks are rarely read, regular scrubs should be scheduled so that ZFS can read all of the data blocks to validate their checksums and correct any corrupted blocks. While multiple disks are required in order to provide redundancy and data correction, ZFS will still provide data corruption detection to a system with one disk. TrueNAS® automatically schedules a monthly scrub for each ZFS pool and the results of the scrub are displayed by selecting the **Volume** and clicking **Volume Status**. Checking scrub results provides an early indication of potential disk problems.

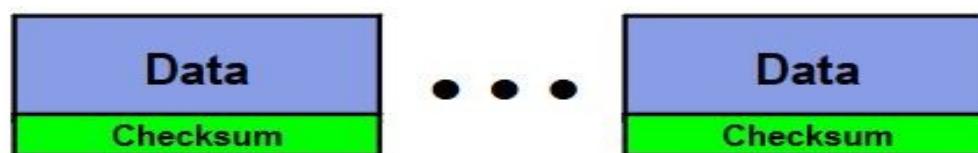
ZFS - Merkle Trees (SHA-256 hash Checksums)

- ZFS uses a Merkle tree to verify the integrity of the entire filesystem and all of the data stored in it.
- If a data block changes, all the parent hash blocks should change as well, including the uber block.
- Each block is cryptographically hashed, and their hash is stored in a node, or "hash block".
- This uber block is responsible for verifying the integrity of the entire Merkle tree.

End-to-End Data Integrity in ZFS

Disk Block Checksums

- Checksum stored with data block
- Any self-consistent block will pass
- Can't detect stray writes
- Inherent FS/volume interface limitation

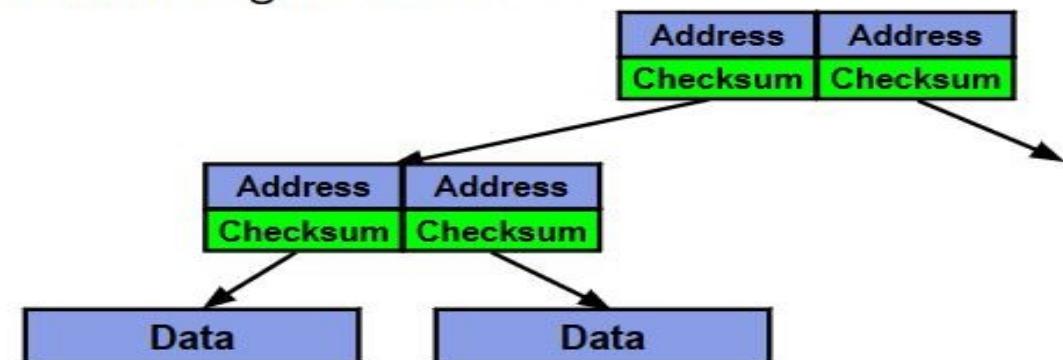


Disk checksum only validates media

- ✓ Bit rot
- ✓ Phantom writes
- ✓ Misdirected reads and writes
- ✓ DMA parity errors
- ✓ Driver bugs
- ✓ Accidental overwrite

ZFS Data Authentication

- Checksum stored in parent block pointer
- Fault isolation between data and checksum
- Checksum hierarchy forms self-validating Merkle tree



ZFS validates the entire I/O path

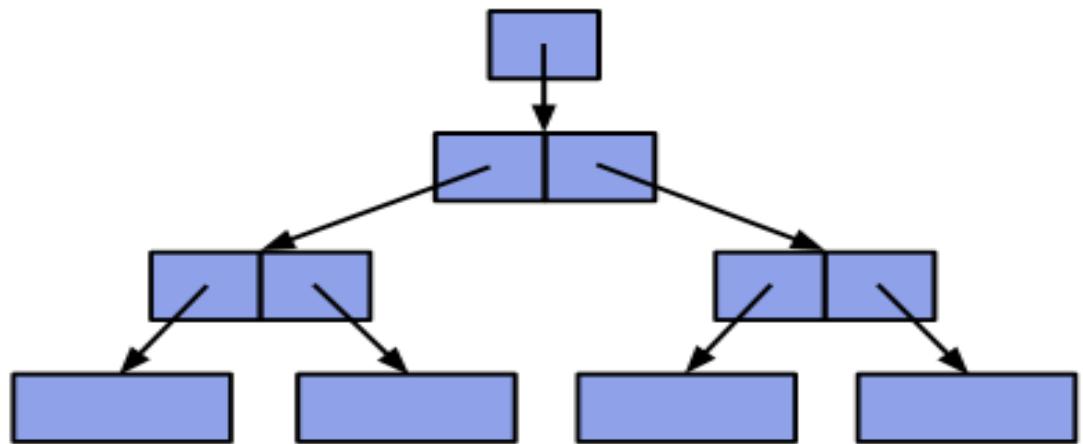
- ✓ Bit rot
- ✓ Phantom writes
- ✓ Misdirected reads and writes
- ✓ DMA parity errors
- ✓ Driver bugs
- ✓ Accidental overwrite

ZFS – Dataset Validation

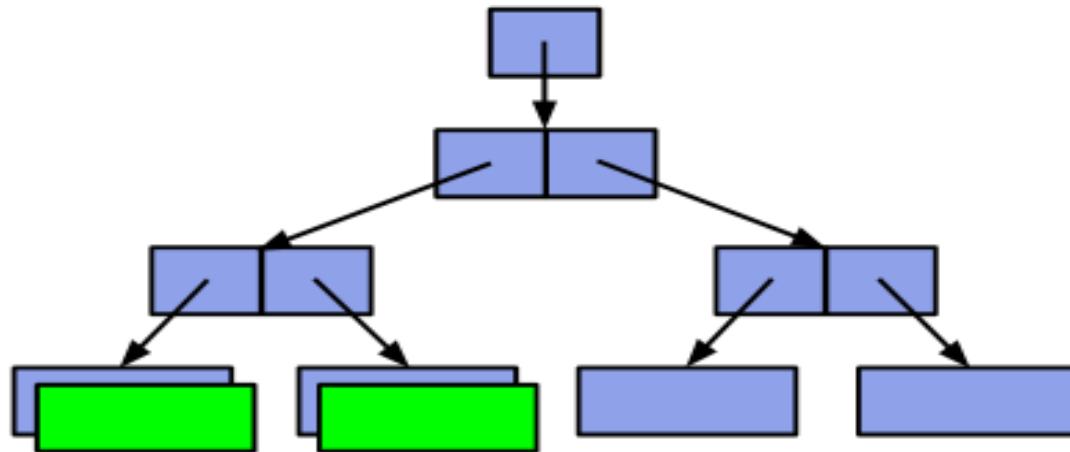
- The merkle tree uses 256-bit checksums stored in the block pointers to protect against misdirected writes.
- Committed data is stored in a merkle tree that is updated atomically on each transaction group commit.
- All metadata is stored twice by default.
- ZIL records storing changes to be made for synchronous IO are self checksumming blocks

Copy-On-Write Transactions

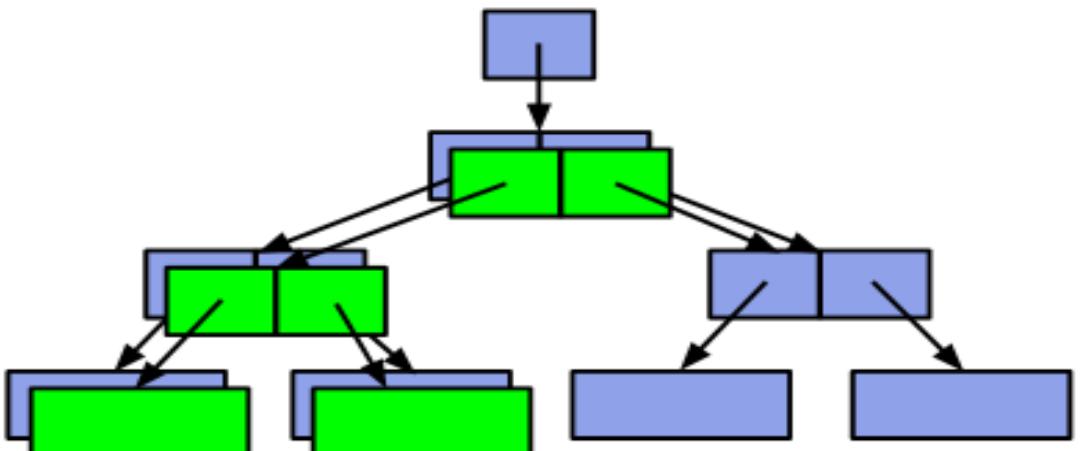
1. Initial block tree



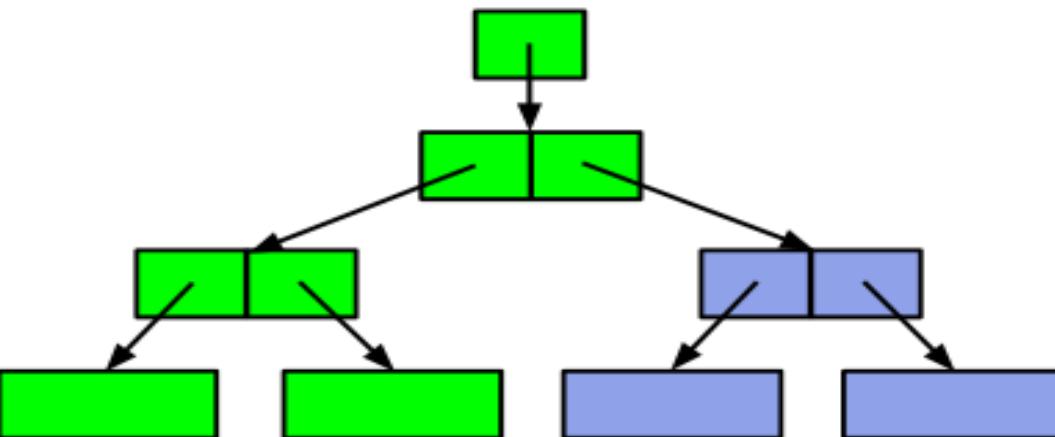
2. COW some blocks



3. COW indirect blocks



4. Rewrite uberblock (atomic)



ZFS - Copy-On-Write (COW)

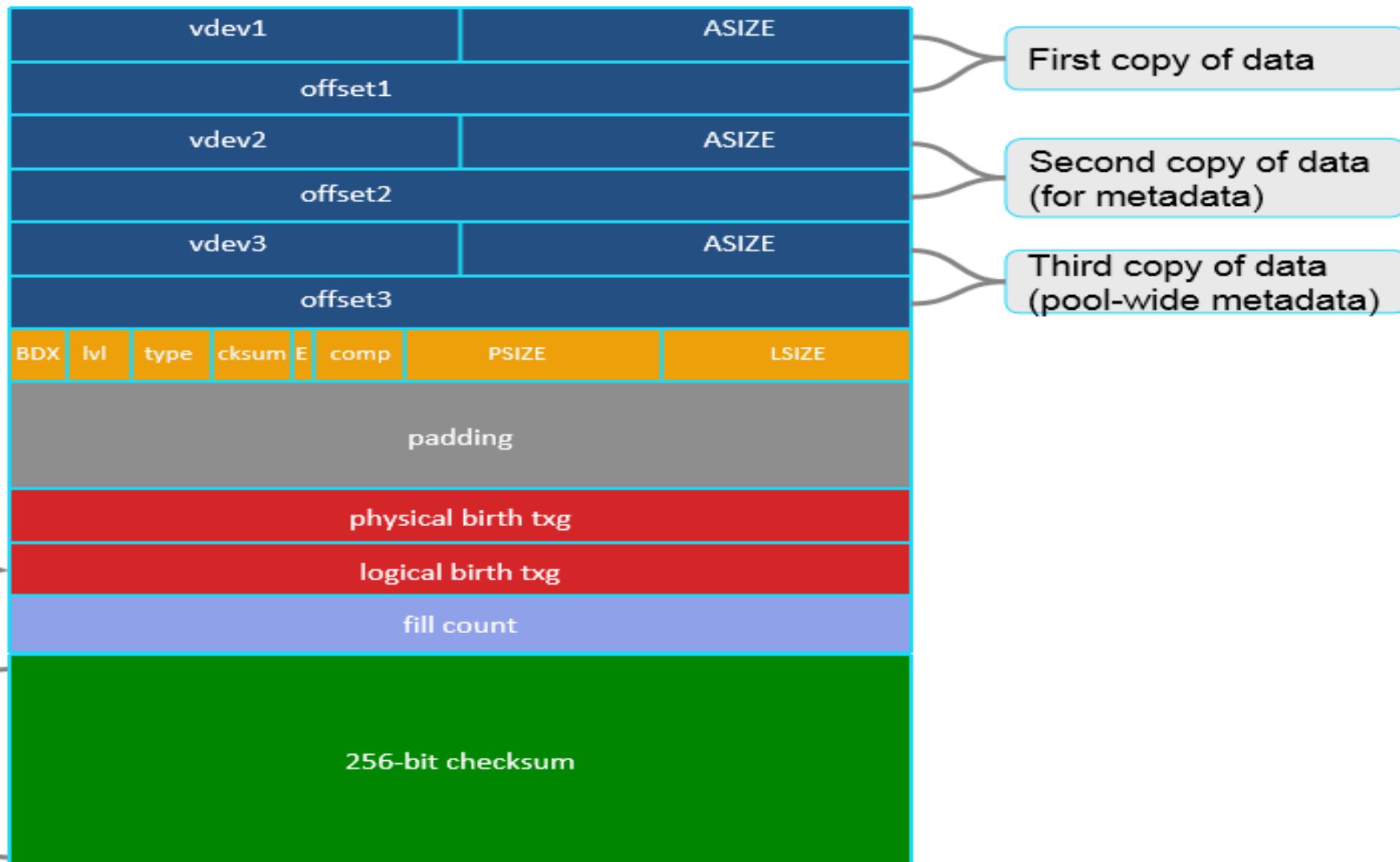
- Instead of modifying a file, you make a copy of the data block that is going to be modified, rather than modify the data block directly.
- You then update your pointers to look at the new block location, rather than the old.
- If you just write a file that didn't exist before, then the data is written to empty space.
- The data is either written or it isn't!
- This means, there's no reason to fsck. If something crashes mid write, the data hasn't changed.

<https://gitlab.com/xelphix/Xelphix-ZFS-Notes>

OpenZFS – Block Pointers

- Normally, filesystems write data in 4 KB blocks.
- ZFS writes data in 128 KB blocks.
- This minimizes the fragmentation by an order of 32.
- Logical Tag – When the block was written.
- 256-Bit Checksum – checksum of data the block points to.
- 3 copies of the data

128-byte Block pointers



ZFS - Attributes

<code>copies=1 2 3</code>	ZFS-aclinherit-definition.png	ZFS-ACLtype-definition.png	ZFS-Checksum-Definition.png	ZFS-Checksums-Definition.png	zfs-clones-definition.png	ZFS-Compression-Definition.png	ZFS-Contexts-Definition.png
<code>Controls the number of copies of data stored for this dataset. These copies are in addition to any redundancy provided by the pool, for example, mirroring or RAID-Z. The copies are stored on different disks, if possible. The space used by multiple copies is charged to the associated file and dataset, changing the used property and counting against quotas and reservations.</code>							
<code>exec=on off</code>							
<code>Controls whether processes can be executed from within this file system. The default value is roToLon.</code>	<code>The values on and off are equivalent to the exec and noexec mount options.</code>						
<code>overlay=off on</code>							
<code>Allows mounting on a busy directory or a directory which already contains files or directories. This is the default mount behavior for Linux file systems. For consistency with OpenZFS on other platforms overlay mounts are off by default. Set to on to enable overlay mounts.</code>							
<code>vscan=on off</code>							
<code>Controls whether regular files should be scanned for viruses when a file is opened and closed. In addition to enabling this property, the virus scan service must also be enabled for virus scanning to occur. The default value is off. This property is not used on Linux.</code>							
<code>xattr=on off sa</code>							
<code>Controls whether extended attributes are enabled for this file system. Two styles of extended attributes are supported either directory based or system attribute based.</code>							
<code>zoned=on off</code>							
<code>Controls whether the dataset is managed from a non-global zone. Zones are a Solaris feature and are not relevant on Linux. The default value is off.</code>							

ZFS – Attributes Continued

```
ut compression=on|off|gzip|gzip-N|lz4|lzjb|zle
Controls the compression algorithm used for this dataset.

est context=none|SELinux_User:SELinux_Role:Selinux_Type:Sensitivity_Level
This flag sets the SELinux context for all files in the file system under a mount point for
that file system. See selinux(8) for more information.

lus fscontext=none|SELinux_User:SELinux_Role:Selinux_Type:Sensitivity_Level
This flag sets the SELinux context for the file system file system being mounted. See
ideos selinux(8) for more information.

rash defcontext=none|SELinux_User:SELinux_Role:Selinux_Type:Sensitivity_Level
This flag sets the SELinux default context for unlabeled files. See selinux(8) for more infor-
e1phix mation.

ref rootcontext=none|SELinux_User:SELinux_Role:Selinux_Type:Sensitivity_Level
This flag sets the SELinux context for the root inode of the file system. See selinux(8) for
itLab- more information.

filesystem_limit=count|none
Limits the number of filesystems and volumes that can exist under this point in the dataset
tree. The limit is not enforced if the user is allowed to change the limit. Setting a
filesystem_limit to on a descendent of a filesystem that already has a filesystem_limit does
not override the ancestor's filesystem_limit, but rather imposes an additional limit. This
feature must be enabled to be used (see zpool-features(5)).

readonly=on|off
Controls whether this dataset can be modified. The default value is off. The values on and
off are equivalent to the ro and rw mount options.
```

ZFS - Attributes Continued

`redundant_metadata=all|most`

Controls what types of metadata are stored redundantly. ZFS stores an extra copy of metadata, so that if a single block is corrupted, the amount of user data lost is limited. This extra copy is in addition to any redundancy provided at the pool level (e.g. by mirroring or RAID-Z), and is in addition to an extra copy specified by the `copies` property (up to a total of 3 copies). For example if the pool is mirrored, `copies=2`, and `redundant_metadata=most`, then ZFS stores 6 copies of most metadata, and 4 copies of data and some metadata.

`primarycache=all|none|metadata`

Controls what is cached in the primary cache (ARC). If this property is set to `all`, then both user data and metadata is cached. If this property is set to `none`, then neither user data nor metadata is cached. If this property is set to `metadata`, then only metadata is cached. The default value is `all`.

`secondarycache=all|none|metadata`

Controls what is cached in the secondary cache (L2ARC). If this property is set to `all`, then both user data and metadata is cached. If this property is set to `none`, then neither user data nor metadata is cached. If this property is set to `metadata`, then only metadata is cached. The default value is `all`.

`nbmand=on|off`

Controls whether the file system should be mounted with `nbmand` (Non Blocking mandatory locks). This is used for SMB clients. Changes to this property only take effect when the file system is unmounted and remounted. See `mount(8)` for more information on `nbmand` mounts. This property is not used on Linux.

If that address is correct, here are three other things you can try:

- If you are connected but behind a firewall, check that

ZFS – Attributes – ACL Type + Checksum

acltype=off|noacl|posixacl

Controls whether ACLs are enabled and if so what type of ACL to use.

off default, when a file system has the acltype property set to off then ACLs are disabled.

noacl an alias for off

posixacl indicates posix ACLs should be used. Posix ACLs are specific to Linux and are not functional on other platforms. Posix ACLs are stored as an extended attribute and therefore will not overwrite any existing NFSv4 ACLs which may be set.

To obtain the best performance when setting posixacl users are strongly encouraged to set the xattr=sa property. This will result in the posix ACL being stored more efficiently on disk. But as a consequence of this all new extended attributes will only be accessible from OpenZFS implementations which support the xattr=sa property. See the xattr property for more details.

checksum=on|off|fletcher2|fletcher4|sha256|noparity|sha512|skein|edonr

Controls the checksum used to verify data integrity. The default value is on, which automatically selects an appropriate algorithm (currently, fletcher4, but this may change in future releases). The value off disables integrity checking on user data. The value noparity not only disables integrity but also disables maintaining parity for user data. This setting is used internally by a dump device residing on a RAID-Z pool and should not be used by any other dataset. Disabling checksums is NOT a recommended practice.

ZFS – Get Options

```
[root@parrot]#[/mnt/ZPool-ZFS]ty was left unspecified, the utf8only property is automatically set to
[ ]# zfs get the normalization property is none. This property cannot be changed after the file system
aclinherit dedup normalization secondarycache userrefs
aclmode utf8only=on | defer_destroy origin setuid userused@...
all devices -p sharenfs utf8only
atime Indicates whether the file system primarycache sharesmb include characters that are
available UTF-8 chargroupquota@...t. If the quota property is explicitly set to off, the no
canmount be explicit group used@...set to none. The default value snapdir to off, the no
casesensitivity is readonly used vscan
checksum logbias recordsize usedbychildren xattr
compression casesensit mlabel normalization referencedbonly proper usedbydataset new permissions that can
compressratio eged user mounted the ZFS delrefquota administration usedbyreservation
copies mountpoint refreservation usedbysnapshots
creationary Mount Point bman properties , reservation userquota@...
```

ZFS – Set Options

```
[root@parrot]~[/mnt/ZPool-ZFS]se-insensitive matching behavior. Currently, case-insensitive matching behavior is limited to the Solaris CIFS server product. For more information, see the ZFS Administration Guide.  
#zfs set checksum=se-insensitive  
aclinherit=mixed compression=or, sgroupquota@.vscan=primarycache=ion refreservation= sharesmb= vscan=  
aclmode= copies= logbias= quota= reservation= snapdir= xattr=  
atime= normalization dedup=none | form mlabel= | formk readonly=d secondarycache= userquota@...= zoned=  
canmount= devices= mountpoint= recordsize= setuid= version=  
checksum= exec= nbmand= refquota=a unicod sharenfs=zation o volsize=ames whenever tv
```

ZFS - Importing A Storage Pool

Example 9 Importing a ZFS Storage Pool

The following command displays available pools, and then imports the pool `tank` for use on the system. The results from this command are similar to the following:

```
# zpool import
  pool: tank
    id: 15451357997522795478
    state: ONLINE
    action: The pool can be imported using its name or numeric identifier.
  config:
    tank      ONLINE
    mirror    ONLINE
      sda     ONLINE
      sdb     ONLINE
# zpool import tank
```

ZFS – List

```
[root@parrot]#/mnt/ZPool-ZFS] Parrot Terminal          x Parrot Terminal          x Parrot Terminal
└─#zfs list -o mounted,name,used,avail,copies,readonly,mountpoint,type of a shape
MOUNTED NAME USED AVAIL COPIES RONLY MOUNTPOINT          TYPE
yes   ZPool-ZFS (3) - Free geometry 301G 3.27T 1 off /mnt/ZPool-ZFS filesystem
no    ZPool-ZFS/Audio (3) - set event 25.1G 3.27T 1 off /mnt/ZPool-ZFS/Audio filesystem
yes   ZPool-ZFS/BrowntownAlpha (3) - nonrectangular 43.6G 3.27T 1 off /mnt/ZPool-ZFS/BrowntownAlpha filesystem
yes   ZPool-ZFS/OSes (3) - X nonrectangular 53.7G 3.27T 1 off /mnt/ZPool-ZFS/OS filesystem
yes   ZPool-ZFS/Scripts (3) - nonrectangular 112K 3.27T 1 off /mnt/ZPool-ZFS/Scripts filesystem
no    ZPool-ZFS/Temp (3) - X nonrectangular 28.8G 3.27T 1 off /mnt/ZPool-ZFS(Temp filesystem
yes   ZPool-ZFS/Wordlists (3) - nonrectangular 54.2G 3.27T 1 off /mnt/ZPool-ZFS/Wordlists filesystem
yes   ZPool-ZFS/Xe1phixGitLab (3) - 33.9G 3.27T 2 off /mnt/ZPool-ZFS/Xe1phixGitLab filesystem
yes   ZPool-ZFS/infosec (3) - nonrectangular 47.1G 3.27T 1 off /mnt/ZPool-ZFS/infosec filesystem
[root@parrot]#/mnt/ZPool-ZFS] rectangular shape functions
└─#zfs list -o name,used,avail,aclmode,aclinherit,zoned,xattr,copies,checksum,compress,readonly
NAME USED AVAIL ACLMODE ACЛИNHERIT ZONED XATTR COPIES CHECKSUM COMPRESS RONLY
beQueryVersion (3) - X 301G 3.27T ar sha256 off off
ZPool-ZFSectInput (3) - X 301G 3.27T ar sha256 off off
ZPool-ZFS/ Audio (3) - X 25.1G 3.27T extgroupmasknctio restricted on on 1 sha256 off off
ZPool-ZFS/BrowntownAlpha (3) - X 43.6G 3.27T extgroupmasknctio restricted off on 1 sha256 off off
ZPool-ZFS/OS (3) - X 53.7G 3.27T extgroupmasknctio restricted off on 1 sha256 off off
ZPool-ZFS/Scripts (3) - X 112K 3.27T extgroupmasknctio restricted off on 1 sha256 off off
ZPool-ZFS/Temp (3) - X 28.8G 3.27T extgroupmasknctio restricted off on 1 sha256 off off
ZPool-ZFS/Wordlists (3) - X 54.2G 3.27T extgroupmasknctio restricted off on 1 sha256 off off
ZPool-ZFS/Xe1phixGitLab (3) - X 33.9G 3.27T extgroupmasknctio restricted off on 2 sha256 off off
ZPool-ZFS/infosec (3) - X 47.1G 3.27T extgroupmasknctio restricted off on 1 sha256 off off
```

ZFS – Get XATTR

```
[root@parrot]#[/mnt/ZPool-ZFS]o not represent the mode of the file.  
[ ]#zfs get xattr  
NAME          bits, unless PROPERTY uVALUE SOURCE  
ZPool-ZFS    ACL permissions xattr reduced so tlocal they are no greater th  
ZPool-ZFS/Audio property of pxattrroughon indic inherited from ZPool-ZFS  
ZPool-ZFS/BrowntownAlpha prxattr the onw mod inherited from ZPool-ZFS  
ZPool-ZFS/OS           xattr      on      inherited from ZPool-ZFS  
ZPool-ZFS/Scripts off       xattr      on      inherited from ZPool-ZFS  
ZPool-ZFS/Temp           xattr      on      inherited from ZPool-ZFS  
ZPool-ZFS/Wordlists whethexattr acceon time inherited from ZPool-ZFS  
ZPool-ZFS/Xe1phixGitLab txattrc wheon ready inherited from ZPool-ZFS  
ZPool-ZFS/infosec s and othxattrmilar on till it inherited from ZPool-ZFS
```

<https://gitlab.com/xelphix/Xelphix-ZFS-Notes>

ZFS – Set XATTR

```
[root@parrot]~[/mnt/ZPool-ZFS]
└─# zfs set xattr=off ZPool-ZFS/audiobyied during chmod(2). A file s
[root@parrot]~[/mnt/ZPool-ZFS]o not represent the mode of the file
└─# zfs get xattrgroup permissions. The permissions are reduced, s
NAME      bits, unless PROPERTY uVALUE SOURCE
ZPool-ZFS  ACL permissions xattr reduc on so t local they are no greater t
ZPool-ZFS/Audio  property of xattrrough off indicat local that no changes are
ZPool-ZFS/BrowntownAlpha pr xattr the on mod inherited from ZPool-ZFS
ZPool-ZFS/OS          xattr    on    inherited from ZPool-ZFS
ZPool-ZFS/Scripts off   xattr    on    inherited from ZPool-ZFS
ZPool-ZFS/Temp          xattr    on    inherited from ZPool-ZFS
ZPool-ZFS/Wordlists whethexattr acceon time inherited from ZPool-ZFS
ZPool-ZFS/Xe1phixGitLab xattrc wheon ready inherited from ZPool-ZFS
ZPool-ZFS/infosec s and othxattrmilar on will inherited from ZPool-ZFS
└─#
```

<https://gitlab.com/xelphix/Xe1phix-ZFS-Notes>

ZFS - Get/Set Compression

```
[root@parrot]# zfs get compression
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS     compression off     default
ZPool-ZFS/Alpha compression off     default
ZPool-ZFS/Brown compression off     default
ZPool-ZFS/OS   compression off     default
ZPool-ZFS/Scripts compression off     default
ZPool-ZFS/Tem compression off     default
ZPool-ZFS/Wordlists compression off     default
ZPool-ZFS/Xe1phixGitLab compression off     default
ZPool-ZFS/infosec compression off     default

[root@parrot]# zfs set compression=on ZPool-ZFS/Scripts
[root@parrot]# zfs get compression
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS     compression off     default
ZPool-ZFS/Alpha compression on     local
ZPool-ZFS/Brown compression off     default
ZPool-ZFS/OS   compression off     default
ZPool-ZFS/Scripts compression on     local
ZPool-ZFS/Tem compression off     default
ZPool-ZFS/Wordlists compression off     default
ZPool-ZFS/Xe1phixGitLab compression off     default
ZPool-ZFS/infosec compression off     default
```

ZFS - Get/Set - Exec/NoExec

```
[root@parrot]#[/mnt/ZPool-ZFS]lly, UNIX and POSIX
└─#zfs get exec
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS    exec      on      default
ZPool-ZFS/Alpha exec      on      default
ZPool-ZFS/BrownTown exec      on      default
ZPool-ZFS/OS     exec      on      default
ZPool-ZFS/Scripts exec      off     default
ZPool-ZFS/Tmp    exec      on      default
ZPool-ZFS/Wordlists exec     on     default
ZPool-ZFS/Xe1phixGitLab exec      on      default
ZPool-ZFS/infosec  exec      on      default
[root@parrot]#[/mnt/ZPool-ZFS]tly was left unspecified
└─#zfs set exec=off ZPool-ZFS/Scripts is none. The property was left unspecified.
[root@parrot]#[/mnt/ZPool-ZFS]
└─#zfs get exec | off
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS    exec      on      default
ZPool-ZFS/Alpha exec      on      default
ZPool-ZFS/BrownTown exec      on      default
ZPool-ZFS/OS     exec      on      default
ZPool-ZFS/Scripts exec      off     local
ZPool-ZFS/Tmp    exec      on      default
ZPool-ZFS/Wordlists exec     on     default
ZPool-ZFS/Xe1phixGitLab exec      on      default
ZPool-ZFS/infosec  exec      on      default
```

ZFS – Get/Set ReadOnly

```
[root@parrot]~[/mnt/ZPool-ZFS]
└─# zfs get readonly
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS    readonly=off      default
ZPool-ZFS/Alpha  readonly=off      default
ZPool-ZFS/BrowntownAlpha  readonly=off      default
ZPool-ZFS/OS      readonly=off      default
ZPool-ZFS/Scripts  readonly=off      default
ZPool-ZFS/Tmp      readonly=off      default
ZPool-ZFS/Wordlists  readonly=off      default
ZPool-ZFS/Xe1phixGitLab  readonly=off      default
ZPool-ZFS/infosec  readonly=off      default
[root@parrot]~[/mnt/ZPool-ZFS]
└─# zfs set readonly=on ZPool-ZFS/Wordlists
[root@parrot]~[/mnt/ZPool-ZFS]
└─# zfs get readonly
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS    readonly=off      default
ZPool-ZFS/Alpha  readonly=on      default
ZPool-ZFS/BrowntownAlpha  readonly=off      default
ZPool-ZFS/OS      readonly=off      default
ZPool-ZFS/Scripts  readonly=off      default
ZPool-ZFS/Tmp      readonly=off      default
ZPool-ZFS/Wordlists  readonly=on      default
ZPool-ZFS/Xe1phixGitLab  readonly=off      default
ZPool-ZFS/infosec  readonly=off      default
```

ZFS - Get/Set - SetUID Bit

```
[root@parrot]#[/mnt/ZPool-ZFS] Parrot Terminal
└─# zfs get setuid
  NAME      PROPERTY  VALUE  SOURCE
ZPool-ZFS  setuid    on     default
ZPool-ZFS/Alpha  setuid    on     default
ZPool-ZFS/OS   setuid    on     default
ZPool-ZFS/Scripts  setuid    on     default
ZPool-ZFS/Temp  setuid    on     default
ZPool-ZFS/Wordlists  setuid    on     default
ZPool-ZFS/Xe1phixGitLab  setuid    on     default
ZPool-ZFS/infosec  setuid    on     default

[root@parrot]#[/mnt/ZPool-ZFS]
└─# zfs set setuid=off ZPool-ZFS/Scripts
[root@parrot]#[/mnt/ZPool-ZFS]
└─# zfs get setuid
  NAME      PROPERTY  VALUE  SOURCE
ZPool-ZFS  setuid    off    local
ZPool-ZFS/Alpha  setuid    on     default
ZPool-ZFS/OS   setuid    on     default
ZPool-ZFS/Scripts  setuid    off    local
ZPool-ZFS/Temp  setuid    on     default
ZPool-ZFS/Wordlists  setuid    on     default
ZPool-ZFS/Xe1phixGitLab  setuid    on     default
ZPool-ZFS/infosec  setuid    on     default
```

<https://gitlab.com/xelphix/Xelphix-ZFS-Notes>

ZFS - Get/Set Zoned

```
[root@parrot]~[/mnt/ZPool-ZFS]
└─# zfs get zoned
  NAME      PROPERTY  VALUE  SOURCE
ZPool-ZFS  sensitive, or allo PROPERTY in VALUE o SOURCE styl
ZPool-ZFS  zoned      off    defaultx fi
ZPool-ZFS/Alpha   zoned      off    local
ZPool-ZFS/BrowntownAlpha zoned      off    defaultvity
ZPool-ZFS/OS       case-sensitive  zoned      off    defaultchin
ZPool-ZFS/Scripts  that sup zoned      mixeoffehav defaultlimi
ZPool-ZFS/Temp     d value beh zoned, seeoffie So defaultFS A
ZPool-ZFS/Wordlists zoned      off    default
ZPool-ZFS/Xe1phixGitLab nonzonedformC offformD defaultKC |
ZPool-ZFS/infosec  zoned      off    default

[root@parrot]~[/mnt/ZPool-ZFS] file system should pe
└─# zfs set zoned=on ZPool-ZFS/Scriptsrmalization al
[root@parrot]~[/mnt/ZPool-ZFS]t of any comparison pr
└─# zfs get zonedonly  property was left unspecifie
  NAME      PROPERTY  VALUE  SOURCE
ZPool-ZFS  sensitive, or allo PROPERTY in VALUE o SOURCE This
ZPool-ZFS  zoned      off    default
ZPool-ZFS/Alpha   zoned      off    local
ZPool-ZFS/BrowntownAlpha zoned      off    default
ZPool-ZFS/OS       indicates wheth zonedde filoffyste default re
ZPool-ZFS/Scripts  characteris zonedde seon If t localropert
ZPool-ZFS/Temp     explicitly s zoned be offto ndefault de
ZPool-ZFS/Wordlists after t zonedle syoffm is defaultd.
ZPool-ZFS/Xe1phixGitLab  zoned      off    default
ZPool-ZFS/infosec  sensitivity, zonedformalofftion default utf
```

ZFS - Checksums

```
[root@parrot]~[/mnt/ZPool-ZFS] Parrot Terminal
└─# zfs get checksum
  NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS      checksum  sha256  local
ZPool-ZFS/Alpha checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/BrownTownAlpha checksum sha256  inherited from ZPool-ZFS
ZPool-ZFS/OS    checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/Script  checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/Temps  checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/Wordlists  checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/Xe1phixGitLab  checksum  sha256  inherited from ZPool-ZFS
ZPool-ZFS/infosec  checksum  sha256  inherited from ZPool-ZFS
```

ZFS - Get/Set Copies

```
[root@parrot]~[/mnt/ZPool-ZFS]
[root@parrot ~]# zfs get copies
NAME          PROPERTY  VALUE   SOURCE
ZPool-ZFS    copies    1       system default
ZPool-ZFS/Audio8  copies    1       system default
ZPool-ZFS/BrowntownAlpha  copies    1       system default
ZPool-ZFS/OS      copies    1       system default
ZPool-ZFS/Scripts  copies    1       default
ZPool-ZFS/Temp     copies    1       system default
ZPool-ZFS/Wordlists  copies    1       ZFS default
ZPool-ZFS/XelphixGitLab  copies    2       local
ZPool-ZFS/infosec  copies    1       default
```

ZFS – Get/Set VScan

```
[root@parrot]#/mnt/ZPool-ZFS]
└─#zfs get vscan
  NAME          PROPERTY  VALUE   SOURCE
  ZPool-ZFS     vscan    off     default
  ZPool-ZFS/Audio vscan    off     default
  ZPool-ZFS/BrowntownAlpha vscan    off     default
  ZPool-ZFS/OS   vscan    off     default
  ZPool-ZFS/Scripts vscan    off     default
  ZPool-ZFS/Temp  vscan    off     default
  ZPool-ZFS/Wordlists vscan    off     default
  ZPool-ZFS/Xe1phixGitLab vscan    off     default
  ZPool-ZFS/infosec  vscan    off     default
[root@parrot]#/mnt/ZPool-ZFS]
└─#zfs set vscan=on ZPool-ZFS/Scripts
[root@parrot]#/mnt/ZPool-ZFS]
└─#zfs get vscan
  NAME          PROPERTY  VALUE   SOURCE
  ZPool-ZFS     vscan    off     default
  ZPool-ZFS/Audio vscan    off     default
  ZPool-ZFS/BrowntownAlpha vscan    off     default
  ZPool-ZFS/OS   vscan    off     default
  ZPool-ZFS/Scripts vscan    on     local
  ZPool-ZFS/Temp  vscan    off     default
  ZPool-ZFS/Wordlists vscan    off     default
  ZPool-ZFS/Xe1phixGitLab vscan    off     default
  ZPool-ZFS/infosec  vscan    off     default
```

<https://gitlab.com/xelphix/Xe1phix-ZFS-Notes>

ZFS – Get/Set DeDuplication

```
[root@parrot]~[/mnt/ZPool-ZFS] into two types, native prope
└─ #zfs get dedup internal statistics or control ZFS beha
  NAME          PROPERTY  VALUE   SOURCE
  ZPool-ZFS    dedup_onmenoffFor more inf
  ZPool-ZFS/  dedup      off     default
  ZPool-ZFS/Audio dedup      off     default
  ZPool-ZFS/BrownTownAlpha dedupof properties that default
  ZPool-ZFS/OS  dedup      off     default
  ZPool-ZFS/Scripts (file s  dedup      off     default
  ZPool-ZFS/Tem  dedup      off     default
  ZPool-ZFS/WordlistsF  dedup      off     can be spec
  ZPool-ZFS/XelphixGitLab dedup      off     the following default
  ZPool-ZFS/infosec dedup      off     default
[root@parrot]~[/mnt/ZPool-ZFS]
└─ #zfs set dedup=on ZPool-ZFS/Scripts
[root@parrot]~[/mnt/ZPool-ZFS]ic properties are case sensi
└─ #zfs get dedup
  NAME          PROPERTY  VALUE   SOURCE
  ZPool-ZFS    dedup_onpertioffconsist of r
  ZPool-ZFS/  dedup      off     defaultda
  ZPool-ZFS/BrownTownAlpha dedup      off     default
  ZPool-ZFS/OS  dedup      off     default
  ZPool-ZFS/Scripts      dedup      on      local
  ZPool-ZFS/Tem  dedup      off     default
  ZPool-ZFS/Wordlists      dedup      off     shared within default,
  ZPool-ZFS/XelphixGitLab dedup      off     reservations default
  ZPool-ZFS/infosec dedup      off     default
```

ZFS – History of commands:

```
zfs get xattr ZPool-ZFS/Scripts or the casesensitivity property in
zfs get xattr case-sensitive and case-insensitive matching behavior.
zfs set xattr=off ZPool-ZFS/Audio fixed behavior is limited to the S
zfs get xattr fixed value behavior, see the Solaris ZFS Administratio
zfs get vscan
zfs set vscan=on ZPool-ZFS/Scripts | formD | formKC | formKD
zfs get vscan
zfs get zoned indicates whether the file system should perform a unic
zfs set zoned=off ZPool-ZFS/Audio which normalization algorithm shou
zfs get zoned e normalized as part of any comparison process. If th
zfs set zoned=on ZPool-ZFS/Scripts was left unspecified, the utf8o
zfs get zoned the normalization property is none. This property ca
zfs get readonly
zfs set readonly=on ZPool-ZFS/Wordlists
zfs get readonly
zfs set readonly=on ZPool-ZFS/Wordlists system should reject file na
zfs get readonly character code set. If this property is explici
zfs list -o mounted,name,used,avail,copies,readonly,mountpoint,type
clear      changed after the file system is created.
zfs get compression
zfs set compression=on ZPool-ZFS/Scripts ion, and utf8only proper
zfs get compression sers by using the ZFS delegated administration
zfs get copies
```

Btrfs FileSystem

- Xe1phix's Btrfs Youtube Video:
 - <https://www.youtube.com/watch?v=V38etDSTtXg>
- Xe1phix's Archive.org Video:
 - <https://archive.org/details/IntroToBtrfsMkfsSubvolumesSnapshotsAndMore>
- Xe1phix's Btrfs Bitchute Video:
 - <https://www.bitchute.com/video/dN3psYmh7Dpd/>

Btrfs - Features

Major Features Currently Implemented

- Extent based file storage
- 2^{64} byte == 16 EiB maximum file size (practical limit is 8 EiB due to Linux VFS)
- Space-efficient packing of small files
- Space-efficient indexed directories
- Dynamic inode allocation
- Writable snapshots, read-only snapshots
- Subvolumes (separate internal filesystem roots)
- Checksums on data and metadata (crc32c)
- Compression (zlib, LZO, ZSTD), heuristics
- Integrated multiple device support
 - File Striping
 - File Mirroring
 - File Striping+Mirroring
 - Single and Dual Parity implementations (experimental, not production-ready)
- SSD (flash storage) awareness (TRIM/Discard for reporting free blocks for reuse) and optimizations (e.g. avoiding unnecessary seek optimizations, sending writes in clusters, even if they are from unrelated files. This results in larger write operations and faster write throughput)
- Efficient incremental backup
- Background scrub process for finding and repairing errors of files with redundant copies
- Online filesystem defragmentation
- Offline filesystem check
- In-place conversion of existing ext2/3/4 and reiserfs file systems
- Seed devices. Create a (readonly) filesystem that acts as a template to seed other Btrfs filesystems. The original filesystem and devices are included as a readonly starting point for the new filesystem. Using copy on write, all modifications are stored on different devices; the original is unchanged.
- Subvolume-aware quota support
- Send/receive of subvolume changes
 - Efficient incremental filesystem mirroring
- Batch, or out-of-band deduplication (happens after writes, not during)
- Swapfile support

Btrfs – Interesting Features

- Btrfs generates checksums for data and metadata blocks.
- Fault isolation is provided by storing metadata separately from user data.
- Data and metadata in Btrfs are protected with copy on write logging (COW).
- In Btrfs, checksums are verified each time a data block is read from disk.

Btrfs – mkfs.btrfs

```
[root@parrot]~[/mnt/ZPool-ZFS]
└─#mkfs.btrfs --data single --metadata single --label Btrfs /dev/sdc
btrfs-progs v4.15.1
See http://btrfs.wiki.kernel.org for more information.

ZPool-ZFS
Label:          Btrfs
UUID:          eafed205-e99b-4b12-b31b-57625b69c5c1
Node size:     16384
Sector size:   4096
Filesystem size: 14.84GiB
Block group profiles:
  Data:           single      8.00MiB
  Metadata:       single      8.00MiB
  System:         single      4.00MiB
SSD detected:  no
Incompat features: extref, skinny-metadata
Number of devices: 1
Devices:
  ID  SIZE  PATH
  1  14.84GiB /dev/sdc
```

Btrfs – Property List/Get

```
[root@parrot]~[/media/xelphix/Btrfs]
└─ #btrfs property list ../Btrfs/
    ro           Set/get read-only flag of subvolume.
    label        Set/get label of device.
    compression Set/get compression for a file or directory
[root@parrot]~[/media/xelphix/Btrfs]
└─ #btrfs property get ../Btrfs/
    ro=false      - Disable D-Bus. D-Bus has long been a huge security
    label=Btrfs   have no problems running Chromium or Firefox. This
```

Btrfs - Subvolumes

```
[root@parrot]~[/media/xelphix/Btrfs]
└─# btrfs subvolume create btrfs-test
Create subvolume './btrfs-test'
[root@parrot]~[/media/xelphix/Btrfs]
└─# ls
btrfs-test
└─audio
    └─audio
        └─audio
```

Btrfs – Subvolume Snapshots

```
[root@parrot]~[/media/xelphix/Btrfs]
└─#btrfs subvolume snapshot btrfs-test/ btrfs-test-snap/
Create a snapshot of 'btrfs-test/' in './btrfs-test-snap'
[root@parrot]~[/media/xelphix/Btrfs]
└─#ls
btrfs-test  btrfs-test-snap
```

Btrfs – Subvolume Snapshot ReadOnly

```
[root@parrot]~[/media/xe1phix/Btrfs]
└─#btrfs subvolume snapshot -r btrfs-test/ btrfs-test-snap-ro/
Create a readonly snapshot of 'btrfs-test/' in './btrfs-test-snap-ro'
[root@parrot]~[/media/xe1phix/Btrfs]
└─#ls
btrfs-test  btrfs-test-snap  btrfs-test-snap-ro
```

Btrfs – FileSystem Usage

```
[x]-[root@parrot]-[/media/xelphix/Btrfs]
└─#btrfs filesystem usage /media/xelphix/Btrfs
Overall:          $ firejail --private-bin=sh,bash,python*
Device size:firejail --blacklist 14.84GiB[1234]
Device allocated:all --read-only 276.00MiB[1-4]
Device unallocated:           14.57GiB
APPARMOR Device missing:      0.00B
Used:pArmor support is disabled at compile time. (240.00KiB)
Free (estimated):             14.58GiB      (min: 14.58GiB)
Data ratio: ./configure --prefix=/1.00 --enable-apparmor
Metadata ratio: ./configure 1.00
Global reserve:are install, a 16.00MiBAppArmor(used:0.00B)file,
The profile needs to be loaded into the kernel by running
Data,single: Size:8.00MiB, Used:64.00KiB
/dev/sdc      # aa-8.00MiB firejail-default
Metadata,single: Size:264.00MiB, Used:160.00KiB some advanced se
/dev/sdc      264.00MiB
System,single: Size:4.00MiB, Used:16.00KiBp" and "ps aux".
/dev/sdc      4.00MiB
Unallocated:   grams and scripts from user home or other director
/dev/sdc      14.57GiB
```

Btrfs – FileSystem Usage 2

```
[root@parrot]~[/media/xelphix/Btrfs]usr --enable-apparmor  
[ ]#btrfs filesystem usage -h -T --si btrfs-test  
Overall:uring software install, a generic AppArmor profile file,  
Device size:le needs to be loaded into 15.94GBnel by running  
Device allocated: 289.41MB  
Device unallocated:roce firejail-default 15.65GB  
Device missing: 0.00B  
Used:e installed profile tries to repl 245.76kBne advanced sec  
Free (estimated): 15.66GB (min: 15.66GB)  
Data ratio: Prevent information leakage 1.00 /proc and /sys  
Metadata ratio:ng commands such as "top" a1.00ps aux".  
Global reserve: 16.78MB (used: 0.00B)  
- Allow running programs only from well-known system  
Dataams MetadataapSystemm user home or other director:btrfs f  
Id Path single single single Unallocated  
-- -----  
1 /dev/sdc 8.39MB 276.82MBro4.19MBot, /15.65GBbin, /srv, /sys, /  
-- -----  
Total 8.39MBa276.82MBs.4.19MB has 15.65GBen a huge securit  
Used 65.54kBn163.84kBm16.38kBng Chromium or Firefox. This
```

Btrfs – Filesystem Stats – df, du

```
[x]-[root@parrot]-[/media/xelphix/Btrfs]
└ #btrfs filesystem df /media/xelphix/Btrfs
  Data, single: total=8.00MiB, used=64.00KiB home or ot
  System, single: total=4.00MiB, used=16.00KiB
  Metadata, single: total=264.00MiB, used=160.00KiB the
  GlobalReserve, single: total=16.00MiB, used=0.00B in,
```

```
[root@parrot]-[/media/xelphix/Btrfs]ser home or other direct
└ #btrfs fi show
Label: 'Btrfs'  uuid:weafed205-e99b-4b12-b31b-57625b69c5c1wing
  Total devices 1 FS bytes used 240.00KiBsbin, /srv, /sy
  devid      1 size 14.84GiB used 276.00MiB path /dev/sdc
  - Disable D-Bus. D-Bus has long been a huge secu
```

Btrfs – Dump SuperBlock

- Print full superblock information.
- Print the system chunk array.
- Print backup roots.
- Print superblock checksum status.
- Print device item and filesystem UUIDs.
- Print information about all present superblock copies

Btrfs – Dump SuperBlock

```
[root@parrot]~[/media/xelphix/Btrfs] into the kernel by running
└─#btrfs inspect-internal dump-super --full --all /dev/sdc
superblock: bytenr=65536, device=/dev/sdclt
-----
csum_type: installed pro01(crc32c) to replicate some advanced s
csum_size: 4
csum: - Prevent 0xe513be69 [match]kage in /proc and /s
bytenr: 65536s such as "top" and "ps aux".
flags: 0x1
        - Allow ru(WRITTEN)ams only from well-known sys
magic: grams and _BHRfS_MF [match]r home or other directo
fsid: eafed205-e99b-4b12-b31b-57625b69c5c1
label: - Allow Btrfss to files only in the following
generation: /mnt, /opt, /proc, /root, /run, /sbin, /srv, /sys
root: 22560768
sys_array_size: Disable 97Bus. D-Bus has long been a huge secur
chunk_root_generation: problems running Chromium or Firefox. Thi
root_level: 0
chunk_root_enable: AppArmor refinement on top of your current
```

Btrfs Superblock Information

- Superblock copies exist in the following offsets on the device:
 - primary: $64KiB$ (65536)
 - 1st copy: $64MiB$ (67108864)
 - 2nd copy: $256GiB$ (274877906944)
- A superblock size is $4KiB$ (4096).
- Use a different starting point (if the primary superblock is damaged)

```
## To copy a superblock:
btrfs check -s <Superblock>
```

Btrfs – Dump Tree

```
[root@parrot]#[/media/xelphix/Btrfs]# btrfs inspect-internal dump-tree --extents /dev/sdc  
btrfs-progs v4.15.1  
extent tree key (EXTENT_TREE_ROOT_ITEM 0) itemoff 16250 itemsize 33  
leaf 22544384 items 16 free space 15371 generation 12 owner 2  
leaf 22544384 flags 0x1(WRITTEN) backref revision 1e advanced sec  
fs uuid eafed205-e99b-4b12-b31b-517625b69c5c1  
chunk uuid e0882671-dfa1-4d4c-880c-fbb6553f01fd in /proc and /sys  
item 0 key (1048576 METADATA_ITEM 0) itemoff 16250 itemsize 33  
refs 1 gen 6 flags TREE_BLOCK  
tree block skinny level 0  
item 1 key (1048576 BLOCK_GROUP_ITEM 4194304) itemoff 16226 itemsize 24  
block group used 16384 chunk_objectid 256 flags SYSTEM  
item 2 key (5242880 BLOCK_GROUP_ITEM 8388608) itemoff 16202 itemsize 24  
block group used 16384 chunk_objectid 256 flags METADATA  
item 3 key (5390336 METADATA_ITEM 0) itemoff 16169 itemsize 33  
refs 1 gen 14 flags TREE_BLOCK  
tree block skinny level 0  
To enable tree block backref root 18446744073709551607 ent 1  
#man ssh-keygen  
#ssdeep cryptofile
```

Btrfs – Dump Tree

- Dump tree structures from a given device
- print only extent-related information: extent and device trees
- print only device-related information: tree root, chunk and device trees
- print only short root node information, ie. the root tree keys
- Print backup root info: The backup root keys and the respective tree root block offset
- analyzing filesystem state or inconsistencies
- Analyze the internal filesystem structure.

Btrfs – Tree Stats

Print sizes and statistics of trees.

```
# btrfs inspect-internal tree-stats /dev/sdc supported: -I  
WARNING: /dev/sdc is already mounted, results may be inaccurate  
Calculating size of root tree  
Total size: 16.00KiB  
    Inline data: 0.00B  
    Total seeks: 0 (private-bin=sh, bash, python*)  
        $ Forward seeks: 0 (list=~/.dir[1234])  
        $ Backward seeks: 0 (only=~/.dir[1-4])  
        Avg seek len: 0.00B  
APPARMOR Total clusters: 1  
    AppArmor Avg cluster size: 0.00B default at compile time  
        Min cluster size: 0.00B  
        $ Max cluster size: 16.00KiB --enable-apparmor  
    Total disk spread: 0.00B  
Total read time: 0/s 0.us generic AppArmor profile file  
    Levels: 1 needs to be loaded into the kernel by running  
Calculating size of extent tree  
Total size: 16.00KiB (rejail-default)  
    Inline data: 0.00B  
    Total seeks: 0 (profile tribe to replicate some advanced  
        Forward seeks: 0  
        Backward seeks: 0 (option leakage in /proc and /  
        Avg seek len: 0.00B as "top" and "ps aux".  
Total clusters: 1  
    Avg cluster size: 0.00B only from well-known sy  
    Min cluster size: 0.00Ber home or other direct  
    Max cluster size: 16.00KiB  
Total disk spread: 0.00B to files only in the following  
Total read time: 0/s 0.us /root, /run, /sbin, /srv, /sy  
    Levels: 1  
Calculating size of csum tree. D-Bus has long been a huge secu  
Total size: 16.00KiB ms running Chromium or Firefox. Th  
    Inline data: 0.00B  
    Total seeks: 0 (Armor confinement on top of your current
```

Btrfs – Mount Compression Flag

```
[root@parrot]~[/media/xelphix/Btrfs]
└─ #btrfs property list ..../Btrfs/
    ro                         Set/get read-only flag of subvolume.
    label                      Set/get label of device.
    compression                Set/get compression for a file or directory
[root@parrot]~[/media/xelphix]
└─ #mount -o compress=lzo /dev/sdb /mnt/
BTrees/Th  parrot_usb/ QemuKVM/    ZPool-ZFS/
[root@parrot]~[/media/xelphix]
└─ #mkdir /mnt/Btrfs && mount -o compress=lzo /dev/sdc /mnt/Btrfs
[root@parrot]~[/media/xelphix]
└─ #cd ../../mnt/Btrfs/
[root@parrot]~[/mnt/Btrfs]
└─ #ls
btrfs-test  btrfs-test-snap  btrfs-test-snap-ro
```

Btrfs – Btrfsck – Filesystem Checking

```
[x]-[root@parrot]-[~]
└─#btrfsck --check-data-csum --progress /dev/sdb
Checking filesystem on /dev/sdb
UUID: 9a65f64f-f2e2-44b3-a6ea-31f8dc9d846b
checking extents [o]
checking free space cache [.]
checking fs roots [o]
checking csums against data
checking root refs
found 15083016192 bytes used, also error found
total csum bytes: 14573044
total tree bytes: 157466624
total fs tree bytes: 135315456
total extent tree bytes: 5849088
btree space waste bytes: 23469580
file data blocks allocated: 14925582336
referenced 14925529088
```

Btrfs – Check – Filesystem Repairing

The checks detect: wrong reference counts of shared extents, backreferences, missing extents of inodes, directory and inode connectivity etc.

```
└─ #btrfs check --repair --check-data-csum --progress /dev/sdc
enabling repair mode
Checking filesystem on /dev/sdc
UUID: 79a525d1-d80e-4117-a427-ea8123bed437
Fixed 0 roots.
checking extents [o]
No device size related problem found
cache and super generation don't match, space cache will be invalidated
checking fs roots [0]
[?] checking csums against data
```

[?] Use a different starting point if the primary superblock is damaged.

```
## To copy a superblock:
btrfs check -s <Superblock>
```

<https://gitlab.com/xelphix/btrfs-notes>

Btrfs – Checks

Overview

- The checks detect:
 - Wrong reference counts of shared extents
 - Backreferences,
 - Missing extents of inodes,
 - Directory and inode connectivity
 - Etc.

XFS FileSystem

- Xe1phix's XFS Youtube Video:
 - To be announced
- Xe1phix's XFS Archive.org Video:
 - To be announced
- Xe1phix's XFS Bitchute Video:
 - To be announced

<https://gitlab.com/xe1phix/ParrotSecWiki/blob/InfoSecTalk/Xe1phix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xe1phix-Linux-FileSystem-ShellScripts+Cheatsheets/Xe1phix-XFS-Cheatsheet-v4.7.sh>

XFS – Mounting + Unmounting

```
[root@parrot]~[~/home/xelphix]
└─#mount -t xfs /dev/sdd /mnt/XFS -o rw
[root@parrot]~[~/home/xelphix]
└─#umount /mnt/XFS
```

<https://archive.org/details/mountingxfsfilesystemsusingthemountcommand>

<https://gitlab.com/xelphix/ParrotSecWiki/blob/InfoSecTalk/Xelphix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xelphix-Linux-FileSystem-ShellScripts+Cheatsheets/Xelphix-XFS-Cheatsheet-v4.7.sh>

XFS - XFSDump

```
[root@parrot]~[/home/xelphix]
└─#xfsdump -f /home/xelphix/XFSBackup /run/media/public/XFS
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.6 (dump format 3.0) - type ^C for status and control

----- dump label dialog -----

please enter label for this dump session (timeout in 300 sec)
-> XFSBackup
session label entered: "XFSBackup"

----- end dialog -----

xfsdump: level 0 dump of parrot:/run/media/public/XFS
xfsdump: dump date: Fri Jul  5 13:24:51 2019
xfsdump: session id: 86eedf00-a4a1-473c-8540-49256d931bcc
xfsdump: session label: "XFSBackup"
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 20800 bytes
xfsdump: /var/lib/xfsdump/inventory created
```

I

<https://archive.org/details/usingxfsdump todumpxfsmetadata>

<https://gitlab.com/xelphix/ParrotSecWiki/blob/InfoSecTalk/Xelphix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xelphix-Linux-FileSystem-ShellScripts+Cheatsheets/Xelphix-XFS-Cheatsheet-v4.7.sh>

XFS - XFSDump

```
===== media label dialog =====

please enter label for media in drive 0 (timeout in 300 sec)
-> XFSBackup
media label entered: "XFSBackup"

----- end dialog -----

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 21016 bytes
xfsdump: dump size (non-dir files) : 0 bytes
xfsdump: dump complete: 18 seconds elapsed
xfsdump: Dump Summary:
xfsdump:   stream 0 /home/xelphix/XFSBackup OK (success)
xfsdump: Dump Status: SUCCESS
```

<https://archive.org/details/usingxfsdumptodumpxfsmetadata>

<https://gitlab.com/xelphix/ParrotSecWiki/blob/InfoSecTalk/Xelphix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xelphix-Linux-FileSystem-ShellScripts+Cheatsheets/Xelphix-XFS-Cheatsheet-v4.7.sh>

XFS - Restore

```
[root@parrot]~|~/home/xelphix]
└ #xfsrestore -v trace -f /home/xelphix/XFSBackup /run/media/public/XFS
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.6 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: file 0 in object 0 of stream 0
xfsrestore: file 0 in stream, file 0 of dump 0 on object
xfsrestore: dump description:
xfsrestore: hostname: parrot
xfsrestore: mount point: /run/media/public/XFS
xfsrestore: volume: /dev/sd[0-9]
xfsrestore: session time: Fri Jul  5 13:24:51 2019
xfsrestore: level: 0
xfsrestore: session label: "XFSBackup"
xfsrestore: media label: "XFSBackup"
xfsrestore: file system id: 2d84979c-9481-46c8-ab4d-07d3ace11609
xfsrestore: session id: 86eedf00-a4a1-473c-8540-49256d931bcc
xfsrestore: media id: 5a8ee92f-16bf-4283-a652-91db9e13311f
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: dump session label: "XFSBackup"
xfsrestore: dump session id: 86eedf00-a4a1-473c-8540-49256d931bcc
```

<https://archive.org/details/restoringxfsfilesystemusingxfsrestore>

<https://gitlab.com/xelphix/ParrotSecWiki/blob/InfoSecTalk/Xelphix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xelphix-Linux-FileSystem-ShellScripts+CheatSheets/Xelphix-XFS-Cheatsheet-v4.7.sh>

XFS - Restore

```
xfsrestore: file system id: 2d84979c-948f-46c8-ab4d-07d3acell689
xfsrestore: session id: 86eedf00-a4a1-473c-8540-49256d931bcc
xfsrestore: media id: 5a8ee92f-16bf-4283-a652-91db9e13311f
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: dump session label: "XFSBackup"
xfsrestore: dump session id: 86eedf00-a4a1-473c-8540-49256d931bcc
xfsrestore: stream 0, object 0, file 0
xfsrestore: initializing directory attributes registry
xfsrestore: initializing directory entry name registry
xfsrestore: initializing directory hierarchy image
xfsrestore: reading directories
xfsrestore: reading the ino map
xfsrestore: reading the directories
xfsrestore: directory 128 0 (0): updating
xfsrestore: 1 directories and 0 entries processed
xfsrestore: number of mmap calls for windows = 1
xfsrestore: directory post-processing
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore:   stream 0 /home/xelphix/XFSBackup OK (success)
xfsrestore: Restore Status: SUCCESS
```

<https://archive.org/details/restoringxfsfilesystemusingxfsrestore>

<https://gitlab.com/xelphix/ParrotSecWiki/blob/InfoSecTalk/Xelphix-InfoSec-Talk-Materials/Intro-To-Linux-Filesystems-SecDSM-May-2019/Xelphix-Linux-FileSystem-ShellScripts+CheatSheets/Xelphix-XFS-Cheatsheet-v4.7.sh>



Filesystem Tools:

```
bool dead = false;  
  
void goDeeper()  
{  
    if (dead == true)  
        return;  
    goDeeper();  
}  
  
int main(void)  
{  
    goDeeper();  
}  
  
while (alive) {  
    eat();  
    sleep();  
    code();  
    repeat();  
}
```

std::cout << "In Code We Trust";

return 0;

538 DENIED

```
me = Person();  
while(me.awake()):  
    me.code();
```

List of Filesystem Tools Covered:

- Smartctl
 - GSmartControl
- Hdparm
- Hdtemp
- FDisk
- SFDisk
- LSBlk
- Fsck

SMART - Self-Monitoring Analysis and Reporting Technology

```
[root@parrot]~[/home/xelphix]
└─#smartctl -a /dev/sda | grep "SMART support"
SMART support is: Available - device has SMART capability.
SMART support is: Disabled
[root@parrot]~[/home/xelphix]
└─#smartctl -s on /dev/sda
smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.19.0-parrot1-13t-amd64] (local build)
Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org
466 smartctl -t long /dev/sda
*** START OF ENABLE/DISABLE COMMANDS SECTION ***
SMART Enabled.
```

Smartctl - XALL

```
[root@parrot]~[~/home/xelphix] [ trust { on | off } ]  
└─#smartctl --xall /dev/sda [ node guid { eui64 } ]  
smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.19.0-parrot1-13t-amd64] (local build)  
Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org  
object FILE [ section NAME ] [ verbose ] |  
==== START OF INFORMATION SECTION ====  
Device Model: TOSHIBA MQ01ABF032  
Serial Number: 24A2S76MS  
LU WWN Device Id: 5 000039 555683f47  
Firmware Version: AM002C  
User Capacity: 320,072,933,376 bytes [320 GB]  
Sector Sizes: 512 bytes logical, 4096 bytes physical  
Rotation Rate: 5400 rpm  
Form Factor: 2.5 inches  
Device is: Not in smartctl database [for details use: -P showall]  
ATA Version is: ATA8-ACS (minor revision not indicated)  
SATA Version is: SATA 2.6, 3.0 Gb/s (current: 3.0 Gb/s)  
Local Time is: Sat Jun 29 17:02:06 2019 CDT  
SMART support is: Available - device has SMART capability.  
SMART support is: Enabled  
AAM feature is: Unavailable  
APM level is: 254 (maximum performance)  
Rd look-ahead is: Enabled  
Write cache is: Enabled  
DSN feature is: Unavailable  
ATA Security is: Disabled, NOT FROZEN [SEC1]  
Wt Cache Reorder: Enabled
```

GSmartControl

Device Information - /dev/sda: TOSHIBA MQ01ABF032 - GSmartControl (sandboxed or root)

Device: /dev/sda Model: TOSHIBA MQ01ABF032

General Attributes Statistics Self-Tests Error Log Temperature Log Advanced

Device Model	TOSHIBA MQ01ABF032
Serial Number	24A2S76MS
World Wide Name	5 000039 555683f47
Firmware Version	AM002C
Capacity	320.07 GB [298.09 GiB, 320072933376 bytes]
Sector Sizes	512 bytes logical, 4096 bytes physical
Rotation Rate	5400 rpm
Form Factor	2.5 inches
In Smartctl Database	No
ATA Version	ATA8-ACS (minor revision not indicated)
SATA Version	SATA 2.6, 3.0 Gb/s (current: 3.0 Gb/s)
Scanned on	Sat Jun 29 16:53:54 2019 CDT
SMART Supported	Yes
SMART Enabled	Yes
AAM Feature	Unavailable
APM Level	254 (maximum performance)
Read Look-Ahead	Enabled
Write Cache	Enabled

Refresh View Output Save As Close

GSmartControl

Smartctl Output - GSmartControl (sandboxed or root)

```
smartctl 6.6 2017-11-05 r4594 [x86_64-linux-4.19.0-parrot1-13t-amd64] (local build)
Copyright (C) 2002-17, Bruce Allen, Christian Franke, www.smartmontools.org
```

```
==== START OF INFORMATION SECTION ====
Device Model: TOSHIBA MQ01ABF032
Serial Number: 24A2S76MS
LU WWN Device Id: 5 000039 555683f47
Firmware Version: AM002C
User Capacity: 320,072,933,376 bytes [320 GB]
Sector Sizes: 512 bytes logical, 4096 bytes physical
Rotation Rate: 5400 rpm
Form Factor: 2.5 inches
Device is: Not in smartctl database [for details use: -P showall]
ATA Version is: ATA8-ACS (minor revision not indicated)
SATA Version is: SATA 2.6, 3.0 Gb/s (current: 3.0 Gb/s)
Local Time is: Sat Jun 29 16:53:54 2019 CDT
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
AAM feature is: Unavailable
APM level is: 254 (maximum performance)
Rd look-ahead is: Enabled
Write cache is: Enabled
DSN feature is: Unavailable
ATA Security is: Disabled, NOT FROZEN [SEC1]
```

```
==== START OF READ SMART DATA SECTION ====
SMART overall-health self-assessment test result: PASSED
```

```
General SMART Values:
Offline data collection status: (0x82) Offline data collection activity
                                was completed without error.
                                Auto Offline Data Collection: Enabled.
```

Save As

Close

Hdparm - Requesting Drive Information

```
[x]-[root@parrot]-[/home/xelphix] #hdparm -I /dev/sda  
/usr/share/doc/wireguard-tools/examples/highlighter/gui/highlight.pro  
/usr/share/doc/wireguard-tools/examples/json/README  
/dev/sda:  
/usr/share/doc/wireguard-tools/examples/json/wg.json  
/usr/share/doc/wireguard-tools/examples/keygen-html/.gitignore  
/usr/share/doc/wireguard-tools/examples/keygen-html/README  
Model Number: TOSHIBA MQ01ABF032  
Serial Number: 24A2S76MS  
Firmware Revision: AM002C  
Transport: Serial, ATA8-AST, SATA 1.0a, SATA II Extensions, SATA Rev 2.5, SATA Rev 2.6  
Standards:  
Supported: 8 7 6 5  
Likely used: 8  
Configuration:  
Logical max current cylinders 16383 16383  
heads 16 16  
sectors/track 63 63  
--  
CHS current addressable sectors: 16514064  
LBA user addressable sectors: 268435455  
LBA48 user addressable sectors: 625142448  
Logical Sector size: 512 bytes  
Physical Sector size: 4096 bytes  
Logical Sector-0 offset: 0 bytes  
device size with M = 1024*1024: 305245 MBytes  
device size with M = 1000*1000: 320072 MBytes (320 GB)
```

Hdparm - Requesting Drive Information Part 2

```
device size with M = 1024*1024:      305245 MBytes
device size with M = 1000*1000:      320072 MBytes (320 GB)
cache/buffer size = 8192 KBytes
Form Factor: 2.5 inch
Nominal Media Rotation Rate: 5400
Capabilities:
    LBA, IORDY(can be disabled)
    Queue depth: 32
    Standby timer values: spec'd by Standard, no device specific minimum
    R/W multiple sector transfer: Max = 16 Current = 16
    Advanced power management level: 254
    DMA: sdma0 sdma1 sdma2 mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5
    Cycle time: min=120ns recommended=120ns
    PIO: pio0 pio1 pio2 pio3 pio4
    Cycle time: no flow control=120ns IORDY flow control=120ns
Commands/features:
    Enabled Supported:
        * SMART feature set
        * Security Mode feature set
        * Power Management feature set
        * Write cache
        * Look-ahead
        * WRITE_BUFFER command
        * READ_BUFFER command
        * NOP cmd
        * DOWNLOAD_MICROCODE
```

Hdparm - Requesting Drive Information Part 2

```
* Advanced Power Management feature set
* 48-bit Address feature set
* Device Configuration Overlay feature set
* Mandatory FLUSH_CACHE
* FLUSH_CACHE_EXT
* SMART error logging
* SMART self-test
* General Purpose Logging feature set
* 64-bit World wide name
* IDLE_IMMEDIATE with UNLOAD
* WRITE_UNCORRECTABLE EXT command
* {READ,WRITE}_DMA_EXT_GPL commands
* Segmented DOWNLOAD_MICROCODE
* Gen1 signaling speed (1.5Gb/s)
* Gen2 signaling speed (3.0Gb/s)
* Native Command Queueing (NCQ)
* Phy event counters
* Idle-Unload when NCQ is active
* DMA Setup Auto-Activate optimization
* Device-initiated interface power management
* Software settings preservation
* SMART Command Transport (SCT) feature set
* SCT Write Same (AC2)
* SCT Error Recovery Control (AC3)
* SCT Features Control (AC4)
* SCT Data Tables (AC5)
```

Hdparm - Requesting Drive Information Part 3

```
*      Native Command Queueing (NCQ)
*      Phy event counters
*      Idle-Unload when NCQ is active
*      DMA Setup Auto-Activate optimization
*      Device-initiated interface power management
*      Software settings preservation
*      SMART Command Transport (SCT) feature set
*      SCT Write Same (AC2)
*      SCT Error Recovery Control (AC3)
*      SCT Features Control (AC4)
*      SCT Data Tables (AC5)
Security:
Master password revision code = 65534
supported
not enabled
not locked
not frozen
not expired: security count
supported: enhanced erase
68min for SECURITY ERASE UNIT. 68min for ENHANCED SECURITY ERASE UNIT.
Logical Unit WWN Device Identifier: 5000039555683f47
NAA      : 5
IEEE OUI : 000039
Unique ID: 555683f47
Checksum: correct
```

Hdparm – Displaying Drive Geometry

```
[x]-[root@parrot]-[/home/xelphix]ff |  
└ #hdparm -g /dev/sda          object FILE [ section NAME ]  
                                pinned FILE ) ]  
/dev/sda:                      [ master DEVICE ] [ vrf NAME ]  
geometry      = 38913/255/63, sectors = 625142448, start = 0
```

Hdparm - Getting/Setting ReadOnly Flag

```
[root@parrot]~[/home/xelphix]
└─#hdparm -r /dev/mmcblk0
```

```
/dev/mmcblk0:
  readonly      =  0 (off)
[root@parrot]~[/home/xelphix]
└─#hdparm -r0 /dev/mmcblk0
```

```
/dev/mmcblk0:
  setting readonly to 0 (off)
  readonly      =  0 (off)
[root@parrot]~[/home/xelphix]
└─#hdparm -rl /dev/mmcblk0
```

```
/dev/mmcblk0:
  setting readonly to 1 (on)
  readonly      =  1 (on)
```

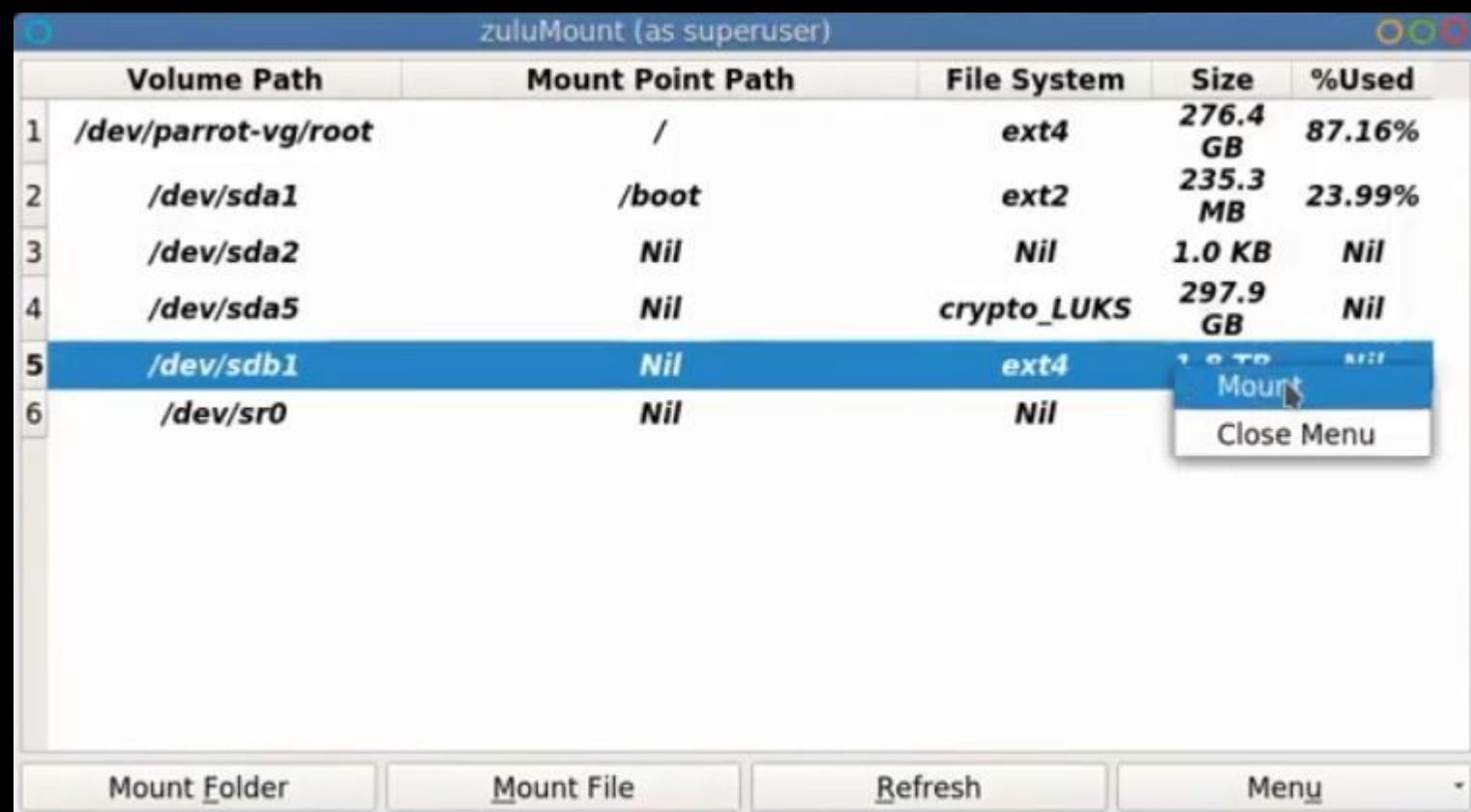
Mounting Using Mount + UDisks

```
[root@parrot]~[/home/xelphix]
└─#mount -t xfs /dev/sdd /mnt/XFS -o rw
[root@parrot]~[/home/xelphix]
└─#umount /mnt/XFS
```

```
mount -t ext2 /dev/fd0 /mnt/floppy
mount -t iso9660 /dev/hdb /mnt/cdrom
mount -t iso /tmp/image_file /mnt/iso_file/ -o loop
```

```
[root@parrot]~[/home/xelphix]
└─#udisksctl mount --block-device /dev/sdb
```

Mounting Devices Using A GUI - ZuluMount



Xelphix-ZuluMount-CLI-Cheatsheet.sh

```
## ----- ##  
zuluMount-cli -m -d /dev/sdc1      ## Mount A Volumes  
zuluMount-cli -u -d /dev/sdc1      ## Unmount A Volumes  
  
zuluMount-cli -d $VolumesPath     ## Volumes Path  
zuluMount-cli -z $MountPoint      ## Mount Point  
zuluMount-cli -e mode rw          ## Read-Write Mode  
zuluMount-cli -e mode ro          ## Read-Only Mode  
## ----- ##  
zuluMount-cli -u -d /dev/sdc1      ## Unmount A Volume  
zuluMount-cli -m -d /dev/sdc2 -p $Pass ## Unmount A Volume  
## ----- ##  
zuluMount-cli -l                  ## Print Expanded List of All Volumes  
zuluMount-cli -P                  ## Print A List of All Volumes  
zuluMount-cli -A                  ## Print A List of All Volumes  
zuluMount-cli -S                  ## Print A List of System Volumes  
zuluMount-cli -N                  ## Print A List of Non System Volumes  
zuluMount-cli -E                  ## Print A List of Mounted Volumes  
zuluMount-cli -D                  ## Get A Device Node Address From its Mapper Path  
## ----- ##  
zuluMount-cli -Y                  ## File System Options  
zuluMount-cli -e                  ## Mount Options  
## ----- ##  
zuluMount-cli -t                  ## Unlock a Volumes as VeraCrypt Volumes, Use "-t vera"  
zuluMount-cli -F                  ## Path To Truecrypt Multiple KeyFiles.  
## ----- ##  
zuluMount-cli -M                  ## Create A Mount Point In "/run/media/private/$USER"  
                                ## Publicly Accessible "mirror" in "/run/media/public/"  
## ----- ##
```

HDDTemp

```
[root@parrot]~[/home/xelphix] ~ i  
└─#hddtemp /dev/sda | team slave  
/dev/sda: TOSHIBA MQ01ABF032: 40°C
```

FDisk – Listing Devices

```
[root@parrot]~[/home/xelphix]
└─#fdisk -l

Disk /dev/sda: 465.8 GiB, 500107862016 bytes, 976773168 sectors
Disk model: WDC WD5000AAKX-0
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xddfcfcbb2



| Device    | Boot | Start   | End       | Sectors   | Size   | Id | Type     |
|-----------|------|---------|-----------|-----------|--------|----|----------|
| /dev/sda1 | *    | 2048    | 1249279   | 1247232   | 609M   | 83 | Linux    |
| /dev/sda2 |      | 1251326 | 976771071 | 975519746 | 465.2G | 5  | Extended |
| /dev/sda5 |      | 1251328 | 976771071 | 975519744 | 465.2G | 83 | Linux    |


```

FDisk – Listing Devices - Part 2

```
Disk /dev/mapper/sda5_crypt: 465.2 GiB, 499464011776 bytes, 975515648 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/mapper/parrot--vg-root: 449.2 GiB, 482282045440 bytes, 941957120 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/mapper/parrot--vg-swap_1: 16 GiB, 17179869184 bytes, 33554432 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/sdd: 14.4 GiB, 15483273216 bytes, 30240768 sectors
```

```
Disk model: DataTraveler 2.0
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Checking The Block Size of A Disk

```
[*]-[root@parrot]-[/home/xelphix]
└─#cat /sys/block/sda/queue/logical_block_size
512
[*]-[root@parrot]-[/home/xelphix]
└─#cat /sys/block/sda/queue/physical_block_size
4096
[*]-[root@parrot]-[/home/xelphix]
└─#blockdev --getpbsz /dev/sda
4096
[*]-[root@parrot]-[/home/xelphix]
└─#blockdev --getss /dev/sda
512
```

LSBlk – Bytes + Perms

```
[root@parrot]~[/home/xelphix]
└─#lsblk --bytes
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda      8:0    0 320072933376  0 disk
└─sda1   8:1    0 598736896   0 part /boot
└─sda2   8:2    0 1024   0 part
└─sda5   8:5    0 319471747072  0 part
  └─parrot--vg-root 253:0    0 311448043520  0 lvm /
  └─parrot--vg-swap_1 253:1    0 8019509248  0 lvm [SWAP]
mmcblk0 179:0   0 15854469120  1 disk
zram0   251:0   0 134217728   0 disk [SWAP]
zram1   251:1   0 134217728   0 disk [SWAP]
[root@parrot]~[/home/xelphix]
└─#lsblk --perms
NAME SIZE OWNER GROUP MODE
sda 298.1G brw-rw----
```

LSBlk – Topology + Filesystem Info

```
[root@parrot]~[/home/xelphix]
└─ #lsblk --topology
NAME          ALIGNMENT MIN-I/O OPT-I/O PHY-SEC LOG-SEC ROTA SCHED      RQ-SIZE RA WSAME
sda            0        4096    0        4096     512    1 mq-deadline   64 128 0B
└─sda1          0        4096    0        4096     512    1 mq-deadline   64 128 0B
└─sda2          1024    4096    0        4096     512    1 mq-deadline   64 128 0B
└─sda5          0        4096    0        4096     512    1 mq-deadline   64 128 0B
  └─parrot--vg-root 0        4096    0        4096     512    1
  └─parrot--vg-swap_1 0        4096    0        4096     512    1
mmcblk0
zram0
zram1

[root@parrot]~[/home/xelphix]
└─ #lsblk --fs
NAME          FSTYPE      LABEL UUID                                     FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1          ext4       f9f71f28-0cf2-43d7-873e-4a3f8d818795  428.9M   14% /boot
└─sda2          raw
└─sda5          LVM2_member hIt0Fe-fVFI-u9CV-Ddnh-WK9e-rD8z-c85GEp
  └─parrot--vg-root btrfs      cbd28195-9297-41c7-8def-1eae5671bca8  171.1G   40% /
  └─parrot--vg-swap_1 swap      3cf3af57-33de-42af-8fc5-1bf005f56c44
mmcblk0
zram0
zram1  ##### [SWAP] [SWAP]
```

Fsck (Fix Disk Errors)

```
[root@tcox1 mnt]# fsck /dev/xvdj1
fsck from util-linux-ng 2.17.2
e2fsck 1.41.12 (17-May-2010)
/dev/xvdj1: clean, 11/65280 files, 8528/261048 blocks
```

Stay Tuned For Part 2

- LVM
- EXT4
- LoopFS
- QCow2
- EWF
- OverlayFS