KSMBD - SMB3 Kernel Server

KSMBD is a linux kernel server which implements SMB3 protocol in kernel space for sharing files over network.

KSMBD architecture

The subset of performance related operations belong in kernelspace and the other subset which belong to operations which are not really related with performance in userspace. So, DCE/RPC management that has historically resulted into number of buffer overflow issues and dangerous security bugs and user account management are implemented in user space as ksmbd.mountd. File operations that are related with performance (open/read/write/close etc.) in kernel space (ksmbd). This also allows for easier integration with VFS interface for all file operations.

ksmbd (kernel daemon)

When the server daemon is started, It starts up a forker thread (ksmbd/interface name) at initialization time and open a dedicated port 445 for listening to SMB requests. Whenever new clients make request, Forker thread will accept the client connection and fork a new thread for dedicated communication channel between the client and the server. It allows for parallel processing of SMB requests(commands) from clients as well as allowing for new clients to make new connections. Each instance is named ksmbd/1~n(port number) to indicate connected clients. Depending on the SMB request types, each new thread can decide to pass through the commands to the user space (ksmbd.mountd), currently DCE/RPC commands are identified to be handled through the user space. To further utilize the linux kernel, it has been chosen to process the commands as workitems and to be executed in the handlers of the ksmbd-io kworker threads. It allows for multiplexing of the handlers as the kernel take care of initiating extra worker threads if the load is increased and vice versa, if the load is decreased it destroys the extra worker threads. So, after connection is established with client. Dedicated ksmbd/1..n(port number) takes complete ownership of receiving/parsing of SMB commands. Each received command is worked in parallel i.e., There can be multiple clients commands which are worked in parallel. After receiving each command a separated kernel workitem is prepared for each command which is further queued to be handled by ksmbd-io kworkers. So, each SMB workitem is queued to the kworkers. This allows the benefit of load sharing to be managed optimally by the default kernel and optimizing client performance by handling client commands in parallel.

ksmbd.mountd (user space daemon)

ksmbd.mountd is userspace process to, transfer user account and password that are registered using ksmbd.adduser (part of utils for user space). Further it allows sharing information parameters that parsed from smb.conf to ksmbd in kernel. For the execution part it has a daemon which is continuously running and connected to the kernel interface using netlink socket, it waits for the requests (dcerpc and share/user info). It handles RPC calls (at a minimum few dozen) that are most important for file server from NetShareEnum and NetServerGetInfo. Complete DCE/RPC response is prepared from the user space and passed over to the associated kernel thread for the client.

KSMBD Feature Status

Feature name	Status
Dialects	Supported. SMB2.1 SMB3.0, SMB3.1.1 dialects (intentionally excludes security vulnerable SMB1 dialect).
Auto Negotiation	Supported.
Compound Request	Supported.
Oplock Cache Mechanism	Supported.
SMB2 leases(v1 lease)	Supported.
Directory leases(v2 lease)	Planned for future.
Multi-credits	Supported.
NTLM/NTLMv2	Supported.
HMAC-SHA256 Signing	Supported.
Secure negotiate	Supported.
Signing Update	Supported.
Pre-authentication integrity	Supported.
SMB3 encryption(CCM, GCM)	Supported. (CCM and GCM128 supported, GCM256 in progress)
SMB direct(RDMA)	Supported.
SMB3 Multi-channel	Partially Supported. Planned to implement replay/retry mechanisms for future.
Receive Side Scaling mode	Supported.
SMB3.1.1 POSIX extension	Supported.

Feature name	Status
ACLs	Partially Supported. only DACLs available, SACLs (auditing) is planned for the future. For ownership (SIDs) ksmbd generates random subauth values(then store it to disk) and use uid/gid get from inode as RID for local domain SID. The current acl implementation is limited to standalone server, not a domain member. Integration with Samba tools is being worked on to allow future support for running as a domain member.
Kerberos	Supported.
Durable handle v1,v2	Planned for future.
Persistent handle	Planned for future.
SMB2 notify	Planned for future.
Sparse file support	Supported.
DCE/RPC support	Partially Supported. a few calls(NetShareEnumAll, NetServerGetInfo, SAMR, LSARPC) that are needed for file server handled via netlink interface from ksmbd.mountd. Additional integration with Samba tools and libraries via upcall is being investigated to allow support for additional DCE/RPC management calls (and future support for Witness protocol e.g.)
ksmbd/nfsd interoperability	Planned for future. The features that ksmbd support are Leases, Notify, ACLs and Share modes.

How to run

1. Download ksmbd-tools and compile them.
    - https://github.com/cifsd-team/ksmbd-tools
2. Create user/password for SMB share.

    # mkdir /etc/ksmbd/ # ksmbd.adduser -a <Enter USERNAME for SMB share access>

3. Create /etc/ksmbd/smb.conf file, add SMB share in smb.conf file
    - Refer smb.conf.example and https://github.com/cifsd-team/ksmbd-tools/blob/master/Documentation/configuration.txt
4. Insert ksmbd.ko module

    # insmod ksmbd.ko

5. Start ksmbd user space daemon
    # ksmbd.mountd
6. Access share from Windows or Linux using CIFS

Shutdown KSMBD

1. kill user and kernel space daemon
    # sudo ksmbd.control -s

How to turn debug print on

Each layer /sys/class/ksmbd-control/debug

1. Enable all component prints
    # sudo ksmbd.control -d "all"
2. Enable one of components (smb, auth, vfs, oplock, ipc, conn, rdma)
    # sudo ksmbd.control -d "smb"
3. Show what prints are enabled.

    # cat /sys/class/ksmbd-control/debug
        [smb] auth vfs oplock ipc conn [rdma]

4. Disable prints:
    If you try the selected component once more, It is disabled without brackets.