

Subject: [PATCH v3, RESEND 06/16] lockd: get rpc_pipefs mount point from callers
Posted by [Kirill A. Shutemov](#) on Tue, 08 Feb 2011 18:41:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Kirill A. Shutemov <kas@openvz.org>

Reviewed-by: Rob Landley <rlandley@parallels.com>

```
fs/lockd/clntlock.c      |  8 ++++++---
fs/lockd/host.c          | 17 ++++++++-----
fs/lockd/mon.c           | 15 ++++++-----
fs/lockd/svc.c           |  6 ++----
fs/nfs/client.c          |  1 +
fs/nfsd/nfssvc.c         |  2 +-
include/linux/lockd/bind.h |  3 ++-
include/linux/lockd/lockd.h |  4 +++-
8 files changed, 36 insertions(+), 20 deletions(-)
```

diff --git a/fs/lockd/clntlock.c b/fs/lockd/clntlock.c

index 8d4ea83..4664c56 100644

--- a/fs/lockd/clntlock.c

+++ b/fs/lockd/clntlock.c

```
@@ -56,13 +56,14 @@ struct nlm_host *nlmclnt_init(const struct nlmclnt_initdata *nlm_init)
    u32 nlm_version = (nlm_init->nfs_version == 2) ? 1 : 4;
    int status;
```

```
- status = lockd_up();
+ status = lockd_up(nlm_init->rpcmount);
    if (status < 0)
        return ERR_PTR(status);
```

```
    host = nlmclnt_lookup_host(nlm_init->address, nlm_init->addrlen,
                               nlm_init->protocol, nlm_version,
-                               nlm_init->hostname, nlm_init->noresvport);
+                               nlm_init->hostname, nlm_init->noresvport,
+                               nlm_init->rpcmount);
    if (host == NULL) {
        lockd_down();
        return ERR_PTR(-ENOLCK);
```

```
@@ -223,7 +224,8 @@ reclaim(void *ptr)
    allow_signal(SIGKILL);
```

```
    down_write(&host->h_rwsem);
- lockd_up(); /* note: this cannot fail as lockd is already running */
+ /* note: this cannot fail as lockd is already running */
+ lockd_up(host->h_rpcmount);
```

```
    dprintk("lockd: reclaiming locks for host %s\n", host->h_name);
```

```

diff --git a/fs/lockd/host.c b/fs/lockd/host.c
index 45e973f..b4cb391 100644
--- a/fs/lockd/host.c
+++ b/fs/lockd/host.c
@@ -14,9 +14,10 @@
#include <linux/in6.h>
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/svc.h>
-#include <linux/sunrpc/rpc_pipe_fs.h>
#include <linux/lockd/lockd.h>
#include <linux/mutex.h>
+#include <linux/mount.h>
+

#include <net/ipv6.h>

@@ -55,6 +56,7 @@ struct nlm_lookup_host_info {
    const char *hostname; /* remote's hostname */
    const size_t hostname_len; /* it's length */
    const int noresvport; /* use non-priv port */
+ struct vfsmount *rpcmount; /* rpc_pipefs mount point */
};

/*
@@ -134,6 +136,7 @@ static struct nlm_host *nlm_alloc_host(struct nlm_lookup_host_info *ni,
    host->h_srcaddrlen = 0;

    host->h_rpcclnt = NULL;
+ host->h_rpcmount = mntget(ni->rpcmount);
    host->h_name = nsm->sm_name;
    host->h_version = ni->version;
    host->h_proto = ni->protocol;
@@ -179,6 +182,7 @@ static void nlm_destroy_host_locked(struct nlm_host *host)

    nsm_unmonitor(host);
    nsm_release(host->h_nsmhandle);
+ mntput(host->h_rpcmount);

    clnt = host->h_rpcclnt;
    if (clnt != NULL)
@@ -207,7 +211,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
    const unsigned short protocol,
    const u32 version,
    const char *hostname,
-    int noresvport)
+    int noresvport,
+    struct vfsmount *rpcmount)
{

```

```

struct nlm_lookup_host_info ni = {
    .server = 0,
@@ -218,6 +223,7 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
    .hostname = hostname,
    .hostname_len = strlen(hostname),
    .noresvport = noresvport,
+ .rpcmount = rpcmount,
};
struct hlist_head *chain;
struct hlist_node *pos;
@@ -243,6 +249,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
    continue;
    if (host->h_version != version)
        continue;
+ if (host->h_rpcmount->mnt_sb != ni.rpcmount->mnt_sb)
+ continue;

    nlm_get_host(host);
    dprintk("lockd: %s found host %s (%s)\n", __func__,
@@ -333,6 +341,7 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
    .version = rqstp->rq_vers,
    .hostname = hostname,
    .hostname_len = hostname_len,
+ .rpcmount = rqstp->rq_server->sv_rpcmount,
};

    dprintk("lockd: %s(host='%s', vers=%u, proto=%s)\n", __func__,
@@ -374,6 +383,8 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
    continue;
    if (!rpc_cmp_addr(nlm_srcaddr(host), src_sap))
        continue;
+ if (host->h_rpcmount->mnt_sb != ni.rpcmount->mnt_sb)
+ continue;

    /* Move to head of hash chain. */
    hlist_del(&host->h_hash);
@@ -464,7 +475,7 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
    .authflavor = RPC_AUTH_UNIX,
    .flags = (RPC_CLNT_CREATE_NOPING |
              RPC_CLNT_CREATE_AUTOBIND),
- .rpcmount = init_rpc_pipefs,
+ .rpcmount = host->h_rpcmount,
};

/*
diff --git a/fs/lockd/mon.c b/fs/lockd/mon.c
index 6219026..a121f5e 100644
--- a/fs/lockd/mon.c

```

```

+++ b/fs/lockd/mon.c
@@ -15,7 +15,6 @@
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/xprtsock.h>
#include <linux/sunrpc/svc.h>
-#include <linux/sunrpc/rpc_pipe_fs.h>
#include <linux/lockd/lockd.h>

#include <asm/unaligned.h>
@@ -63,7 +62,7 @@ static inline struct sockaddr *nsm_addr(const struct nsm_handle *nsm)
    return (struct sockaddr *)&nsm->sm_addr;
}

-static struct rpc_clnt *nsm_create(void)
+static struct rpc_clnt *nsm_create(struct vfsmount *rpcmount)
{
    struct sockaddr_in sin = {
        .sin_family = AF_INET,
@@ -79,13 +78,14 @@ static struct rpc_clnt *nsm_create(void)
        .version = NSM_VERSION,
        .authflavor = RPC_AUTH_NULL,
        .flags = RPC_CLNT_CREATE_NOPING,
-    .rpcmount = init_rpc_pipefs,
+    .rpcmount = rpcmount,
    };

    return rpc_create(&args);
}

-static int nsm_mon_unmon(struct nsm_handle *nsm, u32 proc, struct nsm_res *res)
+static int nsm_mon_unmon(struct nsm_handle *nsm, u32 proc, struct nsm_res *res,
+ struct vfsmount *rpcmount)
{
    struct rpc_clnt *clnt;
    int status;
@@ -101,7 +101,7 @@ static int nsm_mon_unmon(struct nsm_handle *nsm, u32 proc, struct
nsm_res *res)
    .rpc_resp = res,
};

- clnt = nsm_create();
+ clnt = nsm_create(rpcmount);
if (IS_ERR(clnt)) {
    status = PTR_ERR(clnt);
    dprintk("lockd: failed to create NSM upcall transport, "
@@ -151,7 +151,7 @@ int nsm_monitor(const struct nlm_host *host)
    */
    nsm->sm_mon_name = nsm_use_hostnames ? nsm->sm_name : nsm->sm_addrbuf;

```

```

- status = nsm_mon_unmon(nsm, NSMPROC_MON, &res);
+ status = nsm_mon_unmon(nsm, NSMPROC_MON, &res, host->h_rpcmount);
  if (unlikely(res.status != 0))
    status = -EIO;
  if (unlikely(status < 0)) {
@@ -185,7 +185,8 @@ void nsm_unmonitor(const struct nlm_host *host)
    && nsm->sm_monitored && !nsm->sm_sticky) {
    dprintk("lockd: nsm_unmonitor(%s)\n", nsm->sm_name);

- status = nsm_mon_unmon(nsm, NSMPROC_UNMON, &res);
+ status = nsm_mon_unmon(nsm, NSMPROC_UNMON, &res,
+ host->h_rpcmount);
  if (res.status != 0)
    status = -EIO;
  if (status < 0)
diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c
index 32310b1..7387b04 100644
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -31,7 +31,6 @@
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/svc.h>
#include <linux/sunrpc/svcsock.h>
-#include <linux/sunrpc/rpc_pipe_fs.h>
#include <net/ip.h>
#include <linux/lockd/lockd.h>
#include <linux/nfs.h>
@@ -249,7 +248,7 @@ out_err:
/*
 * Bring up the lockd process if it's not already up.
 */
-int lockd_up(void)
+int lockd_up(struct vfsmount *rpcmount)
{
    struct svc_serv *serv;
    int error = 0;
@@ -270,8 +269,7 @@ int lockd_up(void)
    "lockd_up: no pid, %d users??\n", nlmsvc_users);

    error = -ENOMEM;
- serv = svc_create(&nlmsvc_program, init_rpc_pipefs, LOCKD_BUFSIZE,
- NULL);
+ serv = svc_create(&nlmsvc_program, rpcmount, LOCKD_BUFSIZE, NULL);
    if (!serv) {
        printk(KERN_WARNING "lockd_up: create service failed\n");
        goto out;
diff --git a/fs/nfs/client.c b/fs/nfs/client.c

```

```

index 0e0a952..79c01f8 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -675,6 +675,7 @@ static int nfs_start_lockd(struct nfs_server *server)
    .nfs_version = clp->rpc_ops->version,
    .noresvport = server->flags & NFS_MOUNT_NORESVPORT ?
        1 : 0,
+    .rpcmount = init_rpc_pipefs,
    };

    if (nlm_init.nfs_version > 3)
diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
index e4fc85d..dc11012 100644
--- a/fs/nfsd/nfssvc.c
+++ b/fs/nfsd/nfssvc.c
@@ -220,7 +220,7 @@ static int nfsd_startup(unsigned short port, int nrservs)
    ret = nfsd_init_socks(port);
    if (ret)
        goto out_racache;
-    ret = lockd_up();
+    ret = lockd_up(init_rpc_pipefs);
    if (ret)
        goto out_racache;
    ret = nfs4_state_start();
diff --git a/include/linux/lockd/bind.h b/include/linux/lockd/bind.h
index fbc48f8..97cd4bf 100644
--- a/include/linux/lockd/bind.h
+++ b/include/linux/lockd/bind.h
@@ -42,6 +42,7 @@ struct nlmclnt_initdata {
    unsigned short protocol;
    u32 nfs_version;
    int noresvport;
+    struct vfsmount *rpcmount;
    };

/*
@@ -53,7 +54,7 @@ extern void nlmclnt_done(struct nlm_host *host);

extern int nlmclnt_proc(struct nlm_host *host, int cmd,
    struct file_lock *fl);
-extern int lockd_up(void);
+extern int lockd_up(struct vfsmount *rpcmount);
extern void lockd_down(void);

#endif /* LINUX_LOCKD_BIND_H */
diff --git a/include/linux/lockd/lockd.h b/include/linux/lockd/lockd.h
index ff9abff..32dbb7f 100644
--- a/include/linux/lockd/lockd.h

```

```

+++ b/include/linux/lockd/lockd.h
@@ -44,6 +44,7 @@ struct nlm_host {
    size_t  h_addrlen;
    struct sockaddr_storage h_srcaddr; /* our address (optional) */
    size_t  h_srcaddrlen;
+ struct vfsmount *h_rpcmount; /* rpc_pipefs mount point */
    struct rpc_clnt *h_rpcclnt; /* RPC client to talk to peer */
    char *h_name; /* remote hostname */
    u32 h_version; /* interface version */
@@ -222,7 +223,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
    const unsigned short protocol,
    const u32 version,
    const char *hostname,
-   int noresvport);
+   int noresvport,
+   struct vfsmount *rpcmount);
void nlmclnt_release_host(struct nlm_host *);
struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
    const char *hostname,
--
1.7.4

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
