



ORACLE[®]



Linux libtirpc BoF

Chuck Lever
Linux Kernel Architect



Topics

- New GSSAPI support
- Getting to a single user space RPC implementation
- Others?

New GSSAPI Support

- Legacy U-Mich GSSAPI support is incomplete
- New `rpc_gss` API
 - Same API as FreeBSD and Solaris
 - Built into `libtirpc`
 - `RPCAUTH_GSS` now a pre-registered flavor
 - Kerberos only
 - No `/etc/gss/{mech, qop}` files
 - Support for other GSS mechanisms?
- Incomplete legacy U-Mich API remains for now

rpc_gss External API

Mechanism Discovery API

- `void rpc_gss_get_error(rpc_gss_error_t *error)`
- `char **rpc_gss_get_mechanisms(void)`
- `char **rpc_gss_get_mech_info(char *mechanism, rpc_gss_service_t *service)`
- `bool_t rpc_gss_get_versions(u_int *high, u_int *low)`
- `bool_t rpc_gss_is_installed(char *mechanism)`
- `bool_t rpc_gss_mech_to_oid(char *mechanism, rpc_gss_OID *oid)`
- `bool_t rpc_gss_qop_to_num(char *qop, char *mechanism, u_int *num)`

rpc_gss External API

RPC Client API

- AUTH *rpc_gss_seccreate(CLIENT *clnt,
char *principal, char *mechanism,
rpc_gss_service_t service, char *qop,
rpc_gss_options_req_t *req,
rpc_gss_options_ret_t *ret)
- bool_t rpc_gss_set_defaults(AUTH *auth,
rpc_gss_service_t service, char *qop)
- int rpc_gss_max_data_length(AUTH *auth,
int maxlen)

rpc_gss External API

RPC Server API

- `int rpc_gss_svc_max_data_length(struct svc_req *rqst, int maxlen)`
- `bool_t rpc_gss_set_svc_name(char *principal, char *mechanism, u_int req_time, u_int program, u_int version)`
- `bool_t rpc_gss_getcred(struct svc_req *rqst, rpc_gss_rawcred_t **rcred, rpc_gss_ucred_t **ucred, void **cookie)`
- `bool_t rpc_gss_get_principal_name(rpc_gss_principal_t *principal, char *mechanism, char *user_name, char *node, char *secdomain)`

rpc_gss External API

RPC Server Callback API

- ```
typedef struct {
 u_int program;
 u_int version;
 bool_t (*callback)(struct svc_req *,
 gss_cred_id_t, gss_ctx_id_t,
 rpc_gss_lock_t *, void **);
} rpc_gss_callback_t;
```
- ```
bool_t rpc_gss_set_callback(rpc_gss_callback_t  
*callback)
```

GSSAPI Support: Next Steps

- Finish rpcsec_gss API
 - Testing
 - Code review
 - Man page review
- GSS credential cache
 - Support for more than one client credential
 - Manage credential expiry
 - Detect replays

Transition From glibc RPC

- One RPC implementation to rule them
 - man pages
 - Testing
 - Removing headers
 - Glibc surgery

MIT gssrpc Implementation

- Similar to glibc, but has working U-Mich GSS API
 - Not compatible with FreeBSD and Solaris
- No support for TI-RPC
- Collisions with headers, library linking

Other Topics?

