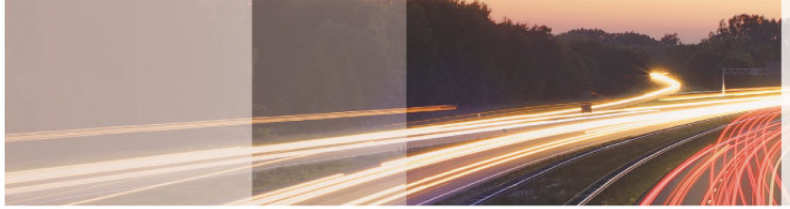


GARR

The Italian Academic & Research Network

www.garr.it

SER

Sip EXpress router

Pierpaolo Culurciello (GARR), Andrea De Vita (IIT – CNR PISA)

WS9, Roma, 15/06/2009

Consortium GARR

Agenda (1)

- Speaker: Pierpaolo Culurciello
  - Introduzione al SER
    - funzionalità (SIP proxy, registrar e redirect)
    - architettura (core e moduli)
    - messagge processing (stateful & stateless proxing, flags)
  - Gestione utenze
    - piano di indirizzamento
    - numerico vs letterale
    - utilizzo degli alias
  - AAA
    - metodi di autenticazione
      - MySQL
      - RADIUS (con credenziali su MySQL o LDAP)
    - metodi di autorizzazione e accounting

Agenda (2)

- Speaker: Andrea De Vita
  - Strumenti di management
    - Serweb
    - Serctl
    - intefaccia custom
  - Instradamento chiamata
    - posizionamento del SER all'interno dell'infrastruttura
    - prefissi di preselezione VoIP/PSTN o fallback automatico
    - Enum lookup - ISN
    - SipEdu
  - NAT traversal
    - problematiche generali
    - soluzione con Mediaproxy

Introduzione a SER – Funzionalità (1)

- SIP Express Router (SER) è un server VoIP:
  - gratuito e con licenza GPL
  - per piattaforme UN*X
  - basato su protocollo SIP.
- Integra tutte le funzioni in RFC3261, tra cui:
  - SIP Registrar
  - Proxy Server
  - Redirect Server
- Inoltre supporta:
  - RADIUS/DB/syslog accounting e authorization
  - ENUM query
- È realizzato da iptel.org: http://www.iptel.org

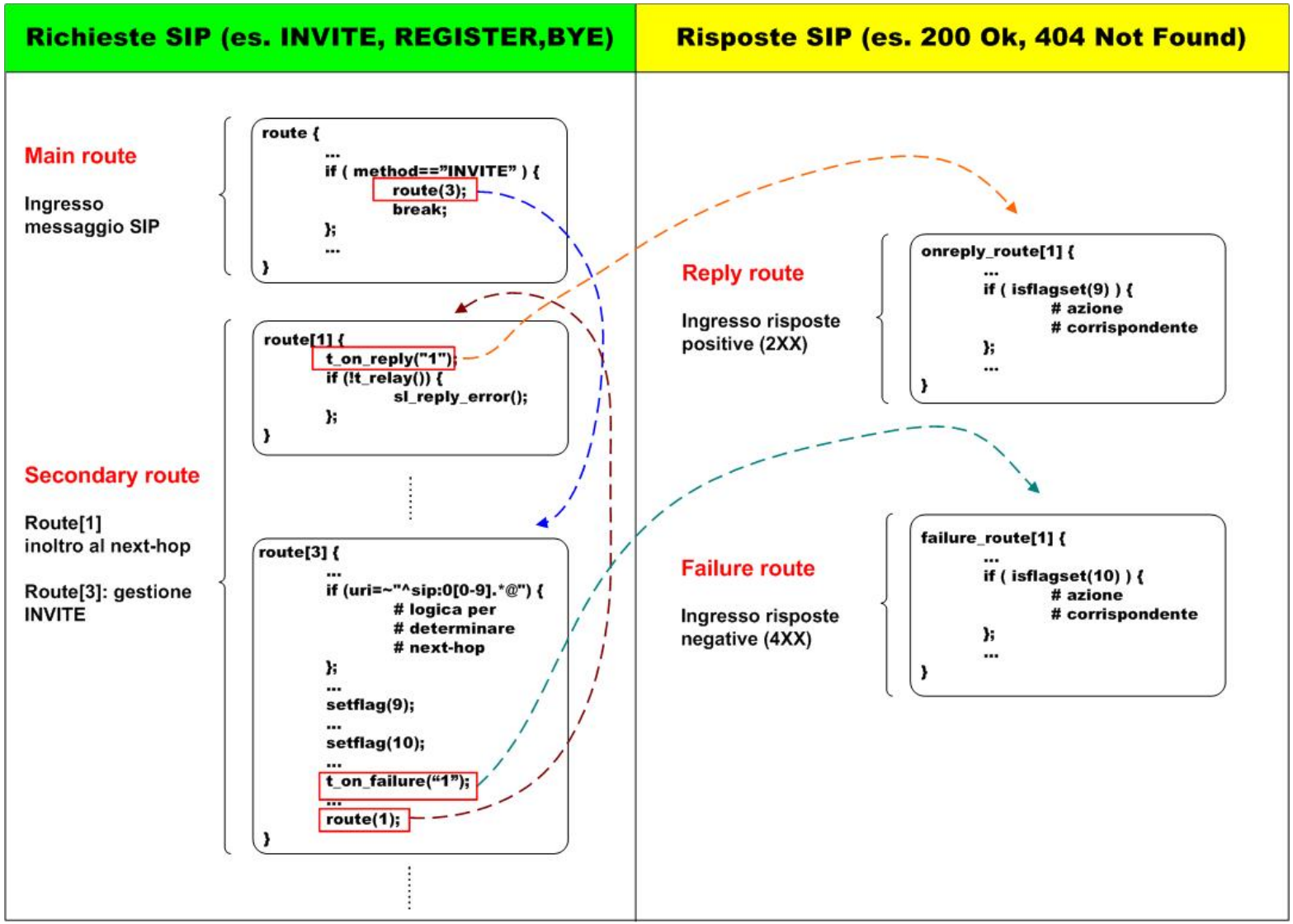
Introduzione a SER – Funzionalità (2)

- Funzione principale di un SIP server è l'instradamento delle richieste (determinarne il next-hop)
- La logica di instradamento può essere complessa
  - route statiche a PSTN gateway
  - route dinamiche verso utenti registrati
  - politiche di autenticazione
- Per questo SER utilizza un apposito linguaggio di routing
- La logica risiede nel file ser.cfg
  - Azioni
    - Integrate nel *core* di SER
    - Esportate da moduli esterni
  - Blocchi delle "route" (più azioni aggregate)
  - Espressioni condizionali
- Ogni messaggio SIP entrante richiede l'esecuzione della logica di routing.

Introduzione a SER – Architettura (core - 1)

- SER è costituito da un processo *core*
  - Funzionalità di base per la gestione dei messaggi SIP
    - funzioni (forward, t_relay, strip etc)
    - espressioni condizionali (if statements)
    - operatori (method, uri, src_ip, dst_ip, src_port)
    - operandi (==, =~)
    - manipolazione delle SIP URI
      - Matching
      - Rewriting
    - Blocchi delle route
      - Main route
      - Secondary route
      - Failure route
      - Reply route

Introduzione a SER – Architettura (core - 2)



Introduzione a SER – Architettura (moduli)

- Funzionalità specifiche disponibili nei moduli aggiuntivi
- L'architettura modulare conferisce al *core* velocità e stabilità di processamento
- Compilare assieme a SER anche i moduli per le funzionalità desiderate
- L'utilizzo dei moduli deve essere dichiarato in *ser.cfg (loadmodule)*
- Sono dotati di parametri di configurazione
  - Ogni parametro ha un valore di default
  - La direttiva *modparam* permette di cambiare tale valore
- Documentazione sui moduli: http://www.iptel.org/doc/module

Introduzione a SER – Message processing (1)

- L'instradamento delle richieste può essere
  - Stateless
    - Messaggi SIP gestiti in modo indipendente
    - Richieste e risposte non sono correlate
  - Stateful
    - È in grado di correlare i messaggi SIP (transazioni)
    - Rispetto ad un proxy stateless può:
      - Effettuare forking
      - Assorbire le ritrasmissioni
      - Tracciare chiamate effettuate e perse (accounting)
  - Tramite i *flag* è possibile marcare il messaggio SIP per effettuare eventuali
    - ulteriori azioni sullo stesso, prima di inoltrarlo
    - azioni sulla relativa risposta
      - Il valore del *flag* resta in memoria per tutta la transazione

Introduzione a SER – Message processing (2)

- Ogni messaggio SIP entra nella *main route*
- Può essere inoltrato alle *secondary route* in base al metodo (REGISTER, INVITE...)
- Altre route secondarie possono svolgere specifiche funzioni
  - Inoltro al PSTN gateway
  - ENUM lookup
  - Trasferimento di chiamata
- Se il proxy opera in modalità *stateful* è possibile gestire
  - risposte positive nella *Reply route* (es. NAT traversal)
  - risposte negative nella *Failure route* (es. inoltro a voicemail)

Gestione utenze – Numerico vs letterale

- L'indirizzamento SIP è basato su URI
  - sip:user@domain.cc
  - La parte *user* può essere numerica o letterale
    - Letterale
      - Mnemonico
      - Facile integrazione con sistemi di autenticazione esterni
      - Non si integra con la telefonia tradizionale
    - Numerico
      - Integrazione con telefonia tradizionale e GDS
      - Semplice gestione del routing

Gestione utenze – Piano di indirizzamento (1)

- Il SER del GARR utilizza URI con *user* letterali
- Problema dell'integrazione con la telefonia tradizionale risolto utilizzando gli *alias* (e naturalmente ENUM)
  - Si associa un numero ad un utente interno

username	pierpaolo.culurciello
alias	3210

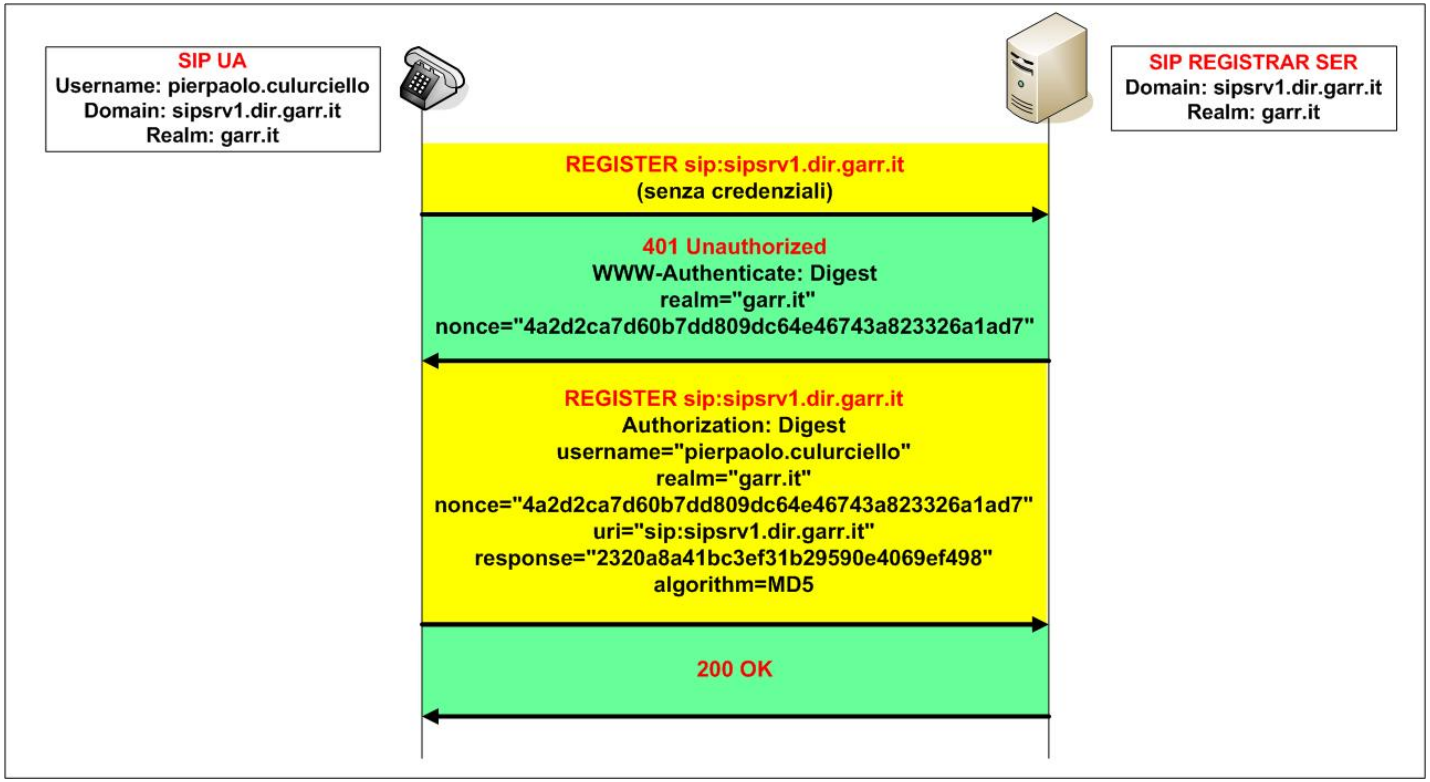
1. Un utente esterno chiama +390649623210
2. Tramite ENUM si cerca sul DNS la SIP URI associata
3. L'URI corrispondente è 3210@sipsrv1.dir.garr.it
4. 3210 è l'alias associato ad un utente registrato a SER

Gestione utenze – Piano di indirizzamento (2)

- Utilizzando gli *alias*, un utente è raggiungibile:
  - Tramite SIP URI
    - sip:pierpaolo.culurciello@sipsrv1.dir.garr.it
    - sip:3210@sipsrv1.dir.garr.it
  - Tramite ENUM
    - +390649623210 (nrenum.net)

AAA – Schema di autenticazione

- SER utilizza Digest Authentication
  - Il client invia un REGISTER senza credenziali
  - SER sfida il client inviando *realm* e *nonce*
  - Il client calcola l'hash MD5 di *username:realm:password (HA1)*
  - Il client risponde con l'hash MD5 di *HA1*, *nonce* e altre info

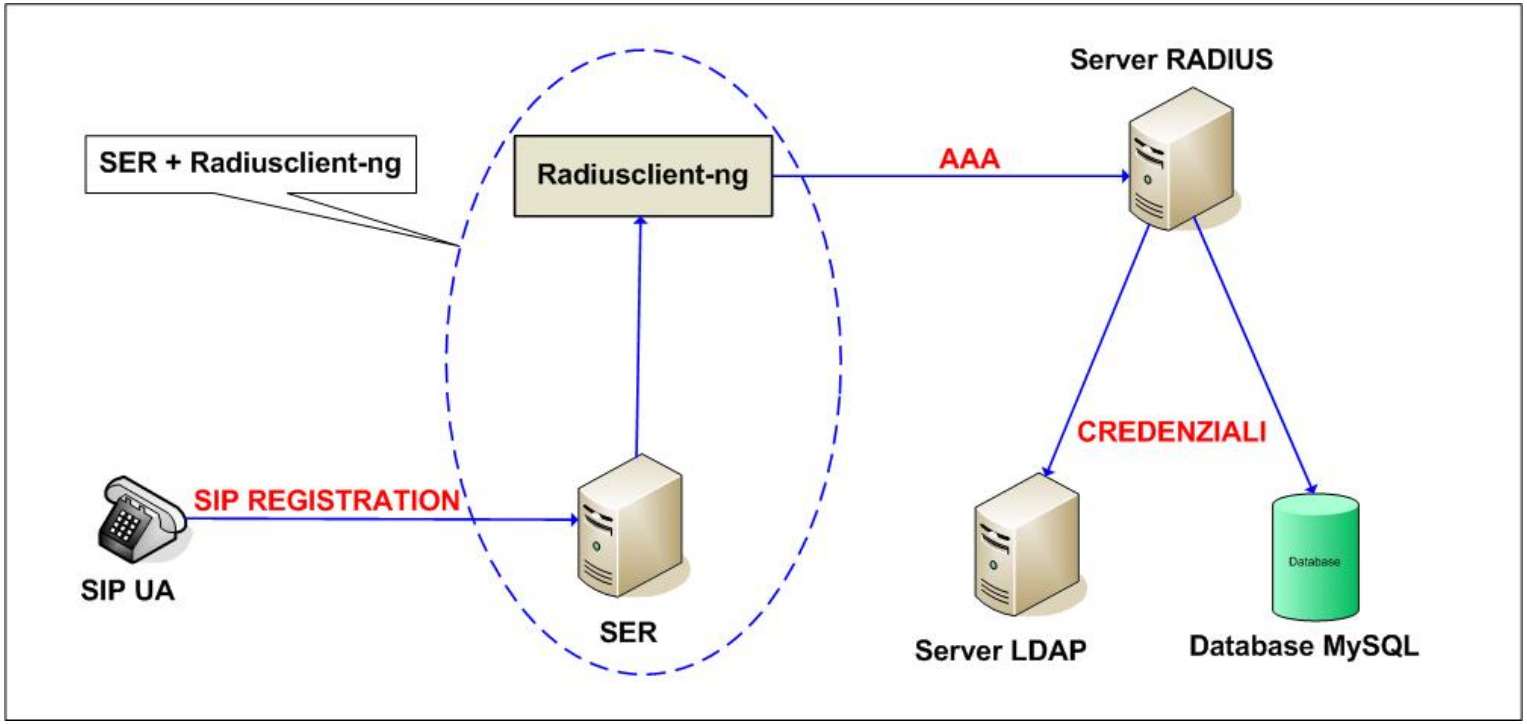


AAA – Metodi di autenticazione

- Sono stati configurati 3 metodi di autenticazione:
  1. Autenticazione effettuata da SER con credenziali su DB MySQL
     - Molte funzionalità di SER fanno uso del DB
  2. Autenticazione tramite RADIUS con credenziali su DB MySQL
  3. Autenticazione tramite RADIUS con credenziali su LDAP
     - SER (0.9.x) non comunica direttamente con LDAP
     - Accesso alle credenziali tramite server RADIUS
- Integrazione con una preesistente infrastruttura di autenticazione basata su RADIUS o LDAP

AAA – Metodi di autenticazione (RADIUS)

- SER comunica con RADIUS tramite Radiusclient-ng
- *Shared secret* e dizionario di SER configurati su Radiusclient-ng e server RADIUS
- LDAP: credenziali memorizzate solo in chiaro
  - necessario tunnel TLS tra server RADIUS e LDAP



AAA – Autorizzazione

- Il modulo *grp* consente di creare ACL
- È possibile consentire o vietare l'accesso ad un servizio
  - Un utente del gruppo "PSTN" può chiamare numeri PSTN
  - Un utente del gruppo "CALLFORWARDING" può trasferire chiamate entranti su un altro numero
- L'appartenenza ad un gruppo si verifica controllando l'username in uno tra:
  - R-URI
  - To URI
  - From URI
  - Credentials
- Per l'accesso su PSTN è opportuno verificare le credenziali

AAA – Accounting (1)

- Il VoIP admin potrebbe voler tracciare determinate chiamate (es. in uscita su PSTN)
- SER è “transaction-stateful”, non “call-stateful”
- È possibile tracciare solo le transazioni concluse
- Le transazioni che identificano una chiamata sono
  - INVITE
  - ACK
  - BYE
- Per default solo le transazioni che iniziano un dialogo SIP visitano SER (es. INVITE)
- È necessario forzare le transazioni successive (ACK, BYE) ad attraversare il SIP proxy (Record Routing)

AAA – Accounting (2)

- In definitiva:
  - SER non dispone di moduli per tracciare chiamate
  - Il modulo *acc* consente di tracciare solo transazioni
    - Per default su *syslog*
    - Va ricompilato per l'accounting su MySQL o RADIUS
  - I dati ottenuti devono essere rielaborati:
    - Correlare le transazioni (es. tramite CallID)
    - Se dati su db MySQL, creare viste per:
      - Chiamate attive
      - Durata chiamate effettuate
      - Chiamate verso PSTN
      - E altro ancora...

SESSIONE PRATICA

Introduzione a SER – message processing

- Esempio di chiamata in ingresso
  1. Un utente esterno compone lo 0649623210
  2. ENUM ritorna la URI 3210@sipsrv1.dir.garr.it
  3. INVITE è inoltrato al SIP proxy responsabile del dominio
  4. Nella main route, verificato che è destinato al proxy ed è un INVITE, si inoltra alla route(3)
  5. Verificato che la chiamata è destinata ad un utente interno si effettua il lookup dell'alias
  6. R-URI modificata in pierpaolo.culurciello@sipsrv1.dir.garr.it

```
route[3] {
        ...
        if (uri=~"^sip:(\+39)?3[0-9]{9}@") {   #chiamata verso cellulari
```

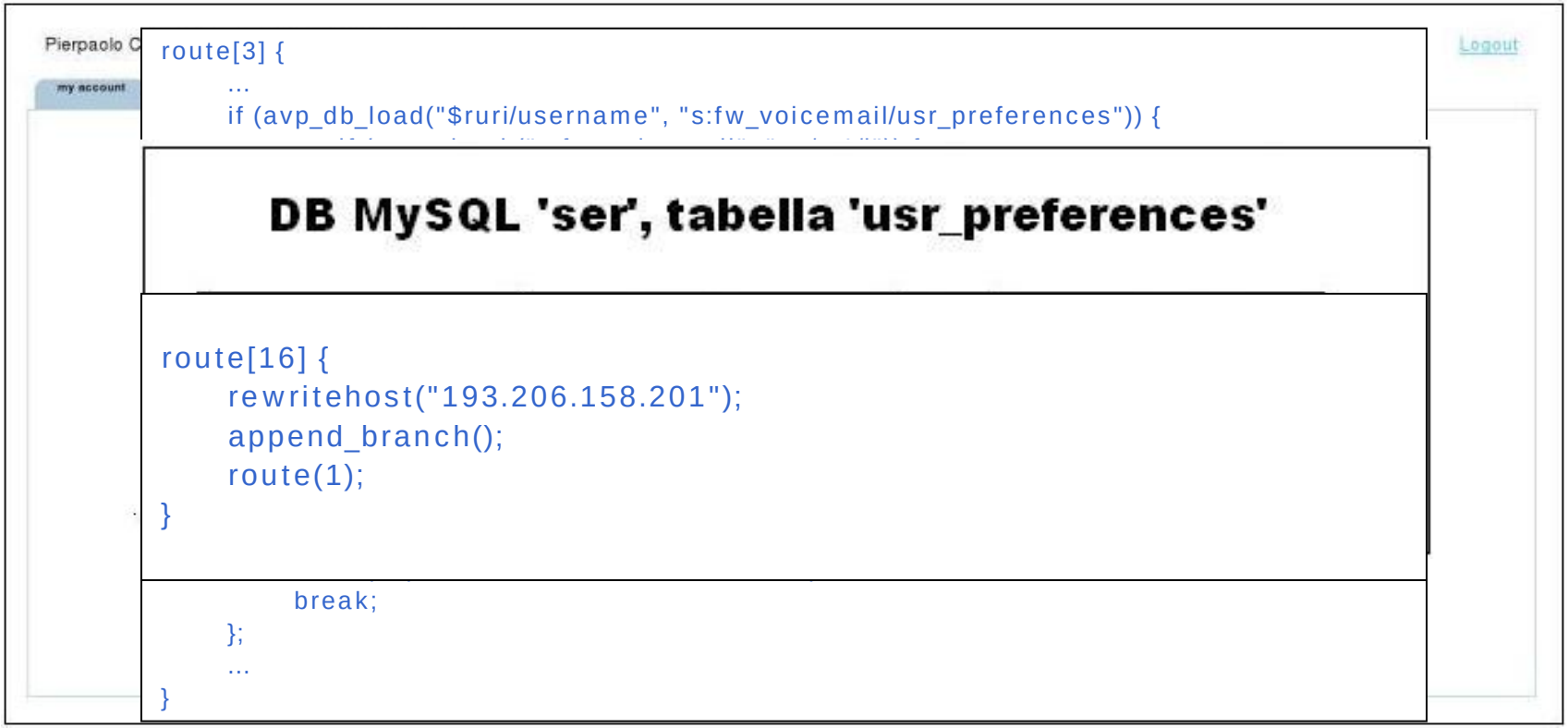
DB MySQL 'ser', tabella 'location'

username	domain	contact	expires
pierpaolo.culurciello	sipsrv1.dir.garr.it	sip:pierpaolo.culurciello@193.206.159.194:1025	2009-06-12 12:22:53

```
}
```

Introduzione a SER – message processing

- Esempio di utilizzo di *flag* e *failure route*: voicemail
  1. Chiamata in ingresso per l'utente pierpaolo.culurciello
  2. L'utente ha abilitato la voicemail (ad esempio da Serweb)
  3. L'utente non risponde, chiamata in timeout (408 Request Timeout)
  4. Failure route: si verifica se il flag per l'inoltro alla voicemail è settato
  5. Si riscrive la R-URI per l'inoltro al PSTN gateway e si crea un nuovo



AAA – Schema di autenticazione


```
route {
        ...
        if (method=="REGISTER") {
                route(2);
                break;
        };
        ...
}
...
route[2] {
        ...
        if (!www_authorize(“garr.it","subscriber")) {
                www_challenge(“garr.it","0");
                break;
        };
        if (!check_to()) {
                sl_send_reply("401", "Unauthorized");
                break;
        };
        ...
}
```

- *check_to()* verifica corrispondenza tra l’username:
  - nelle credenziali (*authorization username*)
  - nel campo To (*SIP username*)
- Con RADIUS si utilizza *radius_www_authorize(“realm”)*

AAA – Autenticazione (MySQL)

- Caricare i moduli:
  - *auth:* funzionalità di base dell'autenticazione SIP
  - *auth_db:* acquisizione credenziali da un database

```
loadmodule "/usr/local/lib/ser/modules/auth.so"
loadmodule "/usr/local/lib/ser/modules/auth_db.so"
```

- Parametri per connessione al db MySQL (*db_url*)

```
modparam("auth_db","db_url","mysql://user:passwd@host/db")
```

- Garantire sicurezza nella memorizzazione e trasmissione delle credenziali:

```
modparam("auth_db", "calculate_ha1", 0)
modparam("auth_db", "password_column", "ha1")
```

- ha1 è l'hash MD5 di username:realm:password

AAA – Autenticazione (RADIUS - 1)

- L'infrastruttura di autenticazione SIP tramite server RADIUS è composta da:
  - SIP registrar: SER 0.9.6
  - Server RADIUS: FreeRADIUS 1.1.3
  - Libreria Radiusclient-ng 0.5.5.1
  - Database: MySQL 5.0.45
  - Server LDAP: OpenLDAP 2.3.43

AAA - Autenticazione (RADIUS - 2)

- Configurazione di SER (file ser.cfg)
  - Caricare il modulo *auth_radius* (oltre ad *auth*)

```
loadmodule “/usr/local/lib/ser/modules/auth.so”
loadmodule “/usr/local/lib/ser/modules/auth_radius.so”
```

  - Indicare il file di configurazione di Radiusclient-ng:

```
modparam("auth_radius", "radius_config",
"/usr/local/etc/radiusclient-ng/radiusclient.conf")
```

  - Utilizzare *radius_www_authorize* anzichè *www_authorize*

```
if (!www_authorize(“garr.it","subscriber")) {
        www_challenge(“garr.it","0");
        break;
};
```

AAA – Autenticazione (RADIUS - 3)

- Configurazione di Radiusclient-ng
  - Specificare il server RADIUS ed il *realm* di autenticazione in radiusclient.conf

```
authserver localhost
acctserver localhost
default_realm garr.it
```

  - Inserire lo shared sercret in servers

```
localhost   radius_password
```

  - Aggiungere il contenuto del dizionario di SER in quello utilizzato da Radiusclient-ng

```
# cat /usr/local/etc/ser/dictionary.ser >>
  /usr/local/etc/radiusclient-ng/dictionary
```

AAA – Autenticazione (RADIUS - 4)

- Configurazione del server RADIUS (1)
  - Consentire a Radiusclient l'accesso al server nel file clients.conf

```
client 127.0.0.1 {
      secret            radius_password
      nastype           other
}
```

  - Indicare il realm di autenticazione in proxy.conf

```
realm garr.it {
     type         = radius
     authhost     = localhost:1812
     accthost     = localhost:1813
     secret       = radius_password
}
```

  - Includere nel file dictionary gli attributi usati da SER

```
$INCLUDE      /usr/local/etc/radiusclient-ng/dictionary.ser
```

AAA – Autenticazione (RADIUS - 5)

- Configurazione del server RADIUS (2)
  - Abilitare in radiusd.conf il modulo per *Digest Authentication*
  - Specificarne l'utilizzo nelle sezioni a*uthenticate* e a*uthorize*

```
authorize {
modules {
        ...
        digest {
        }
        ...
}
        digest
        ...
}
```

AAA – Autenticazione (RADIUS - 5)

- Configurazione del server RADIUS (3)
  - È possibile memorizzare le credenziali nel file users

```
pierpaolo.culurciello
Auth-Type := Digest,
Digest-HA1 := "88c34fdd259f236824da1ea061a369d5"
```

  - Abilitare in radiusd.conf nella sezione *authorize* l'utilizzo del file:

```
authorize {
        ...
        files
        ...
}
```

  - In caso di credenziali su DB MySQL o server LDAP sostituire *files* con *sql* o *ldap*

AAA – Autenticazione (RADIUS + MySQL)

- Configurazione del server RADIUS (4)
  - Nel sorgente di Freeradius è compreso il file mysql.sql con la struttura del DB MySQL per RADIUS
  - Impostare i parametri di connessione a *mysqld* in sql.conf

```
sql {
        driver = "rlm_sql_mysql"
        server = "localhost"
        login = "radius"
        password = “*********"
        radius_db = "radius“
        ...
}
```

AAA – Autenticazione (RADIUS + LDAP)

- Configurazione del server RADIUS (1)
  - Mappare gli attributi di RADIUS ed LDAP

```
# cp /etc/raddb/ldap.attrmap /etc/openldap/schema/radius.schema
```

  - Includere la mappatura in slapd.conf

```
include       /etc/openldap/schema/radius.schema
```

  - Configurare nella sezione *modules* il lookup su LDAP
    - Indicare la directory (*basedn*) su cui effettuare il *bind*
    - I permessi sono importanti!
    - Utilizzare una *identity* che ha accesso alla *directory*
    - Password in chiaro su LDAP, configurare tunnel TLS

AAA – Autenticazione (RADIUS + LDAP)

- Configurazione del server RADIUS (2)
  - In radiusd.conf inserire i parametri di connessione

```
ldap {
        server      = "localhost"
        identity    = "cn=admin,dc=sipsrv1,dc=dir,dc=garr,dc=it"
        password    = guess_me
        basedn      = "ou=utenti,dc=sipsrv1,dc=dir,dc=garr,dc=it"
        filter      = "(uid=%{Stripped-User-Name:-%{User-Name}})"
        ...
        port = 636
        tls_cacertfile          = /etc/pki/tls/certs/CA/cacert.pem
        tls_cacertdir           = /etc/pki/tls/certs/CA
        tls_certfile            = /etc/pki/tls/certs/RADIUS/servercert.pem
        tls_keyfile             = /etc/pki/tls/certs/RADIUS/serverkey.pem
        tls_randfile            = /dev/random
        tls_require_cert        = "demand"
        ...
        password_attribute      = userPassword
        ...
}
```

AAA – Autorizzazione

- In ser.cfg:
  - Prima di inoltrare alla PSTN, autenticare la richiesta
  - Ottenute le credenziali, verificare che l'utente sia abilitato ad effettuare chiamate su PSTN

```
route[5] {
    ...
    route (15);
    ...
```

```
 route[5] {
    ...
    route (15);
    ...
    if (!is_user_in("Credentials", "PSTN")) {
            sl_send_reply("401", "Unauthorized PSTN call");
            break;
    };
    ...
}
```

```
}
```

AAA – Accounting (1)

- Accounting su syslog di default
- Abilitare il supporto MySQL/RADIUS nel Makefile del modulo *acc*

```
DEFS+=-DSQL_ACC
DEFS+=-DRAD_ACC
```

- In ser.cfg:
  - Caricare i moduli *tm* ed *acc*
  - Impostare al valore X il parametro *acc_flag*
  - Usare *setflag(X)* prima di *t_relay()* per marcare le transazioni

```
loadmodule "modules/acc/acc.so“
modparam("acc", "db_url", "mysql://ser:*****@localhost/ser")
modparam("acc", "db_flag", 2)
...
route[1] {
    t_on_reply("1");
    setflag(2);
    if (!t_relay()) {
            sl_reply_error();
    };
}
```

AAA – Accounting (2)

- L'accounting è memorizzato nella tabella *acc* del database MySQL di SER
- Sono inseriti i messaggi che iniziano una transazione
- Devono essere correlati per ottenere informazioni sulle chiamate

```
SELECT DISTINCT main.from_uri AS `Chiamante`, main.to_uri AS `Chiamato`,
                main.sip_callid AS `Callid`, DATE( main.time ) AS `Data`,
                TIME( main.time ) AS `Ora` ,
                TIMEDIFF( byes.time, acks.time ) AS `Durata`
FROM ser.acc AS main INNER JOIN ser.acc AS acks ON (
```

Chiamante	Chiamato	Callid	Data	Ora	Durata
sip:pierpaolo.culurciello@sipsrv1.dir.garr.it	sip:0113977533@sipsrv1.dir.garr.it;user=phone	3c2e88dea39e-10bi3yn7bska	2009-05-28	13:48:37	00:09:31
sip:pierpaolo.culurciello@sipsrv1.dir.garr.it	sip:0498295772@sipsrv1.dir.garr.it;user=phone	3c2e4a0392fb-bnqf8i0kuo7w	2009-05-28	09:20:35	00:05:50

```
          AND ( acks.to_uri = main.to_uri OR acks.to_uri = main.from_uri )
          AND main.sip_method = acks.sip_method
          AND acks.sip_method = 'ACK' ) INNER JOIN ser.acc AS byes ON (
                                acks.time <= byes.time
                                AND acks.sip_callid = byes.sip_callid
                                AND byes.sip_method = 'BYE' )
ORDER BY `Ora` DESC;
```