# strongSwan - Feature #1268

## Android client wrapper not setting IKE_SA remote id (rightid) correctly.

11.01.2016 13:45 - Michael Schmoock

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 11.01.2016 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Tobias Brunner | **Estimated time:** | 0.00 hour |
| **Category:** | android | | |
| **Target version:** | 5.5.0 | | |
| **Resolution:** | Fixed | | |

**Description**

When Using the Android client wrapper to connect to a linux strongSwan server, the server is not able to recognize the remote id string (rightid) that should be set as the servers DNS name in the IKE_SA_INIT if not specified otherwise. Instead it can only see '%any' when android tries to connect:

```
Jan 11 10:29:32 vpntest1 charon: 12[CFG] looking for peer configs matching 10.3.10.25[%any]...10.3
.3.110[CN=android.somedns.org]
```

However when doing the same thing using linux+strongSwan also client one gets the intended behaviour:

```
Jan 11 11:26:17 vpntest1 ipsec: 08[CFG] looking for peer configs matching 10.3.10.25[vpntest1.some
dns.org]...10.3.2.120[CN=linux.vpntest1.somedns.org]
Jan 11 11:26:17 vpntest1 ipsec: 08[CFG] selected peer config 'linux1'
...
```

The problem is that there are valid use cases to use the servers remote id for distinguishing between various server side settings. When the server can only see '%any' here it is not able to set special config options for that remote id.

Since linux+strongSwan as client does set this remote id string and android, that use the same sources, does not, the problem should be located in the android JNI client wrapper sources.

### Associated revisions

**Revision 8b3bf4a4 - 02.05.2016 18:38 - Tobias Brunner**

android: Use configured remote ID in auth-cfg

If one is explicitly set we don't use loose identity matching and send it as
IDr to the server.

Closes #strongswan/strongswan#29.
Fixes #1268.

### History

**#1 - 12.01.2016 10:09 - Michael Schmoock**

The above Issue description is inaccurate. What the Android wrapper does not do is to set the IDr (right/remote ID) within the IKE_AUTH request.

**#2 - 12.01.2016 11:26 - Michael Schmoock**

Fixed by pull Request: https://github.com/strongswan/strongswan/pull/27
Or this change: https://github.com/willsteel/strongswan/commit/d19b7b52dde5df4f99a1551c930aafe9d6136ca0

```
src/frontends/android/app/src/main/jni/libandroidbridge/backend/android_service.c
 @ -727,7 +727,7 @ static job_requeue_t initiate(private_android_service_t *this)
      auth = auth_cfg_create();
      gateway = identification_create_from_string(server);
      auth->add(auth, AUTH_RULE_IDENTITY, gateway);
 -    auth->add(auth, AUTH_RULE_IDENTITY_LOOSE, TRUE);
 +    auth->add(auth, AUTH_RULE_IDENTITY_LOOSE, FALSE);
      auth->add(auth, AUTH_RULE_AUTH_CLASS, AUTH_CLASS_PUBKEY);
      peer_cfg->add_auth_cfg(peer_cfg, auth, FALSE);
```

**#3 - 18.01.2016 18:13 - Tobias Brunner**

*- Status changed from New to Feedback*

> The above Issue description is inaccurate. What the Android wrapper does not do is to set the IDr (right/remote ID) within the IKE_AUTH request.

Yes, that's on purpose. It allows the responder to use a different identity than the client intends to enforce (the configured hostname/IP). This is e.g. the case when *leftid* is not set on the server and it defaults to the subject DN of the server certificate. This way the hostname/IP only has to be contained as subjectAltName in the server certificate and does not have to be used as IKE identity by the server.

> The problem is that there are valid use cases to use the servers remote id for distinguishing between various server side settings. When the server can only see '%any' here it is not able to set special config options for that remote id.

Yes, that's true but many other simple remote access clients do the same thing as is simplifies the configuration. We might change that in the future (e.g. add an optional field to explicitly set the expected remote identity) but until then I won't apply that patch.

#### #4 - 20.01.2016 17:38 - Michael Schmoock

My colleague added a proper merge request that makes the IDENTITY_LOOSE flag configurable by connection settings:
https://github.com/strongswan/strongswan/pull/29

The default still applies the same behavior as current sources.
Key: "conenction.loose_identity"
Value: Boolean
Default: true

#### #5 - 02.05.2016 19:01 - Tobias Brunner

*- Tracker changed from Issue to Feature*

*- Status changed from Feedback to Closed*

*- Assignee set to Tobias Brunner*

*- Target version set to 5.5.0*

*- Resolution set to Fixed*

The next version of the app (see associated commit) will allow configuration of the remote identity via "connection.remote_id". If it is set that identity is sent as IDr to the server. So you can simply set it to the same value as "connection.gateway" to disable the loose identity matching (in the GUI I've added auto-completion of the configured server address/hostname to the remote identity field).