Welcome, IP gateway, Portal pairing specification

Major changes

In this updated specification, we were able to substantially reduce the specification and implementation effort by simplifying the certificate requirements, while preserving the security of the proposed solution.

1. Welcome app will use a single certificate and connection to portal flexisip;
2. Ip-gateways will not act as a CA;
3. Portal will issue only client certificates (no intermediate CAs);
4. User friendly sip account creation;
5. No Internet connectivity account creation.

Please note that the public/private keys must use RSA.

IP gateway pairing (implemented by CNDEX)

The owner should be connected with its portal credentials to initiate the pairing process.
IP gateway generates a random alphanumeric domain, noted DOMAIN, and generates a pair of RSA private/public keys.
Owner's credentials are used to setup a TLS connection to portal with digest authentication.
IP gateway immediately submits a CSR providing ipgw@DOMAIN as CNAME, together with a friendly name (e.g. Location: Loft).
Portal processes the CSR and delivers a certificate back to the ip gateway, signed by the portal CA. In case the generated DOMAIN is already in use, the portal returns an error and the procedure is restarted from the beginning.
All further connections issued by the IP gateway must be made using this certificate.

At this point:

- the public CA certificate of the portal is retrieved and stored in /etc/b2b/tls/cafile.pem.
- the ipgw@DOMAIN private key and certificate are concatenated and stored in /etc/b2b/tls/agent.pem.
- an internal flexisip password is generated for user ipgw@DOMAIN (b2b proxy) and added

to flexisip password file.
- the /etc/b2b/b2b.conf is generated.

Welcome app pairing

1. App gets paired with portal.

Welcome generates a pair of RSA private/public keys.
User's credentials are used to setup a TLS connection to portal with digest authentication.
Welcome immediately submits a CSR for username@portal, together with a friendly name ('John's iphone').
Portal processes the CSR and delivers a certificate back to Welcome, signed by the portal CA.
All further http connections issued by the Welcome must be made using this certificate.

If app is IP gateway owner's app (same credentials as the IP gateway), then portal automatically adds the IP gateway to list of the IP gateways this app can monitor.

Note that in case of a user having multiple devices, it is expected that several certificates with the same CN will be issued. This behaviour is different from the ipgw pairing, where the CN is unique.

The same portal API as for ip-gateway pairing is used.

2. App asks portal for all joinable IP-Gateways (either from own account or from accounts that gave guest access)

The API may be:
-> GET http://theportal/api/joinable_gateways
<- list of gateways unique domains and friendly names (eventually empty).

3. App user chooses IP-Gateway from list

4. App configure sip account without Internet connectivity (implemented by CNDEX)

The owner should enter all needed information, and a QR code is displayed. Owner also

specifies if he allows external access.

The phone should scan the QR code to retrieve the account information. (implemented by Belledonne Communications). It checks that the entered hash was correct.

The owner activates the verified account.

This procedure is a light adaptation of current Welcome pairing procedure.

No further action is required for external access through portal flexisip.

When Internet connectivity is back, the ip-gateway sends pending events to Portal.

5. App asks IP-Gateway for access over portal connection

App sends a request for accessing this IP gateway.
Portal sends message to IP-gateway to notify that App is requesting a SIP account to use this IP gateway.

The API may be:
→ GET http://theportal/api/create_sip_account/DOMAIN
← event with success or fail

6. IP-Gateway owner is informed about new pending authorization request. (implemented by CNDEX)

The event must include the certificate of the requiring Welcome app and the friendly name.

The hash of the public key must be computed by ip-gateway.

7. IP-Gateway shows request to local administrator and waits for approval.(implemented by CNDEX)

The hash is not shown to the owner.

8. Administrator configures access rights.(implemented by CNDEX)

9. Once approved by owner, the IP gateway processes the subscription request.(implemented by CNDEX)

The IP gateway creates a sip account in its flexisip SIP server.

The SIP password is generated by the IP gateway and encrypted using the Welcome app certificate. Since only the app has the private key of this certificate, it is the only one that is able to decode the SIP password. This encrypted SIP password is pushed to portal together with a random token called “gruu” (see http://tools.ietf.org/html/rfc5627). Its purpose is to uniquely identify the device.

At this stage, the new account is not yet enabled. It will be enabled when the owner enters the first characters of the Welcome app public key hash, which must be provided by the Welcome user by “human” means (see step 11).

10. The Welcome app asks the portal if subscription request has been approved.

If yes, portal transfers the encrypted password and gruu.
App decodes the SIP password with its private key.
App connects to the SIP server using its portal certificate, its portal username and the gruu.

However, the account is not activated in the ip-gateway flexisip until the direct communication occurs.

Application might reproduce the steps from 5 to 12 in order to be granted access to other IP gateway (such as when this app is used by a “guest”).

The request may be:
→ GET https://theportal/api/get_approved_account/DOMAIN
← event containing encrypted password and gruu.

11. Welcome app user and owner communicate with each other (implemented by CNDEX)

The app user read to the owner a 8 characters value, which is a security code displayed by welcome, which is generated from its public certificate.
Preferably, this communication should be done by phone or directly.
It may however be done in an asynchronous way such as email or SMS.

This security code is composed of 2 parts:
- a 4 digits salt, randomly chosen;
- the first 4 digits of the hash of (salt:hash of Welcome public key).

This security code is generated after the reception by the Welcome of the sip account, which guarantees that a man-in-the-middle is too late for generating a certificate that would match this challenge.

The owner is required to enter this security code in the IP gateway administration website. The IP gateway checks that this security code is matching the certificate of the requesting user (that is known by the IP gateway because it was transmitted by the portal at step 6).
Once this is done, the owner is sure that there is no man-in-the-middle attack, able to read the sip account information transmitted to Welcome.

This action enables the account by adding the account to ACLs and username/password to ip-gateway flexisip.

12. App and IP-Gateway can talk to each other via SIP

The app can make a TLS connection to the portal SIP server by presenting its SIP certificate.

The portal SIP server verifies the connection from the clients by verifying the presented TLS SIP certificates. If the provided certificate is not signed by portal CA, then the connection is immediately closed.
–
When receiving a SIP request from the client, the portal SIP server will check if the username@domain in the From SIP header matches the username@domain in CN of the certificate presented during the TLS handshake. If they match, then the portal SIP server can process the request. If they don't match, server will send a 403 Forbidden.
A Welcome app client will have a CN like portal_username@portal.
The ip-gateway client (b2b proxy) will have a CN like ipgw@unique_ipgw_domain.

A Welcome app will register once on the portal flexisip, whatever the number of actual ip-gateways it has a sip account for.

A call from a given ip-gateway will be only forked to the list of allowed users which is defined on this ip-gateway. Only the apps with an allowed portal username will be able to receive the call. On the other side, when receiving a SIP request from the app (relayed by portal sip server), the IP gateway will challenge it according to the username and password created for this app. This secures access to the cameras as well as opening of the doors.

Using the gruu that was exchanged during the sip account creation procedure, it is even possible to restrict the forking, not only to a list of users, but more specifically to a list of devices. Thus it is possible to remove a stolen device from the list of recipients.

Because the security is handled by the ip-gateway, an attacker can not open the doors without

knowing a valid username and password stored in the targeted ip-gateway, even if the portal certificates are compromised.

13. Changed ACL and ip-gateway topology is sent to portal. (implemented by CNDEX)

The portal will store an event list.

14. Welcome fetch events from Portal.

In several circumstances, the app will fetch recent events from Portal.
The received events may be:

- Journal events (calls, misses, …);
- Topology changes (ipgw1234 adds a DSE, update names, …);
- ACL changes (door open right, monitor right, …).

Consequence of using client based certificates on the overall architecture

- No SQL database is required for the portal SIP server.
- The public key corresponding to the private one used to sign the SIP certificates must be installed on the SIP server in order to allow verification of clients.

ANNEX

1. Software requirements for the B2B proxy

The b2b proxy depends on polarssl and bellesip.

It is configured through /etc/b2b/b2b.conf, /etc/b2b/tls/agent.pem, /etc/b2b/tls/cafile.pem, /etc/b2b/acl.list.

The b2b.conf file will include portal flexisip sip uri, internal flexisip sip uri, b2b password and ipgw domain.

There are no changes in flexisip.

2. Configuring access list for the B2B software (specified by BJE, implemented by CNDEX).

The ipgw updates the /etc/b2b/acl.list;

A signal is sent to the b2b process for reloading ACLs.

The format of acl.list is a whitespace separated list of allowed usernames.

It is the responsibility of the ipgw to update the acl.list and notify the process.

3. Kill switch to prevent external access (implemented by CNDEX).

Ipgw should start/stop the b2b service.

4. Unspecified parts

- how the Welcome is permitted to see an ipgw as joinable;

Initiator should be the owner to prevent spams and security risk by app users;