ALLSEEN ALLIANCE

Core Working Group Summit

October 21, 2015

The contents of this document, and the interfaces described, and all the information herein, are the result of collaborative discussions by the Core Working Group. This summary documents the final consensus of the team.

Antitrust Compliance Notice

- AllSeen Alliance meetings involve participation by industry competitors, and it is the intention of AllSeen Alliance to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of and not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at AllSeen Alliance meetings and in connection with AllSeen Alliance activities are described in the AllSeen Alliance Antitrust Policy. If you have questions about these matters, please contact your company counsel, or if you are a member of AllSeen Alliance, feel free to contact Lee Gesmer or Andrew Updegrove, of the firm of Gesmer Updegrove LLP, which provides legal counsel to AllSeen Alliance.

Agenda

1. Core WG Technical Meeting
   - Group discussion
   - Architecture discussion
2. Core WG Summit
   - What we have done so far
   - Status on open source test assets
   - Tech overview of Security 2.0
   - 16.04 feature set

Core WG Technical Meeting

Core WG Technical Meeting

- Group discussion
  - Identify top technical issues with AJCore
  - Identify leads/teams for each of them to begin efforts to resolve
  - Leads for each of the architecture topics will present 5-10 minutes summarizing issue and next steps

Core WG Technical Meeting

- Architecture topics
  - MSFT
    - Scalability and performance: any ideas related to Erdinc's recent measurements? (TBD next week)
      - Any other known issues?
    - Enabled AJ mDNS to interoperate with off the shelf mDNS implementations (Dave Thaler)
      - How would this work? What are the implications for the current implementation?
    - Mutex class (Dan Mihai)
      - Detecting potential deadlocks
      - Should recursive lock acquires be banned?
    - Anyone interested in enabling AllJoyn across the LAN / multiple network links? (Dan Mihai)
      - What are some of the technical challenges?
  - QCE
    - Concurrent callbacks - design, limitations, common deadlocks, etc. (Todd Malsbary)
      - Previously I discussed these with Todd and other QCE folks.
    - Optimize PubSub signals (Todd Malsbary)
      - Broadcast/sessioncast/addmatch: How those interact for optimal delivery?

Core WG Summit

Agenda

1. What we have done so far
2. Status on open source test assets
3. Tech overview of Security 2.0
4. 16.04 feature set

What we have done so far

Marcello Lioy

What have we done so far?

- 14.02
  - First official Alliance release
- 14.06
  - Next Generation Name Service (NGNS)
  - Security Enhancements: support for ECC based authentication
  - Policy DB: rules to manage the flow of messages through the router to and from applications
  - Sessionless Signal (SLS) optimizations
  - Events and Actions
  - UDP Transport (experimental)

What have we done so far?

- 14.12
  - Major Stabilization fixes for the system when it is under load
  - Hardening and performance improvements of the UDP RN to RN transport
  - Integration of the About feature into the core framework
  - Support for the Self-Join session feature
  - Auto-pinger feature
  - Migration of PropertyChanged signal from BusListener to ProxyBusObject
  - Heartbeat from the Router to TC application
  - UDP Transport: router to router transport using reliability protocol on top of UDP
  - Blacklisting feature for TC RN discovery
  - Configuration of the heartbeat from the routing node to the TC
  - Support for discovering RN using mDNS/NGNS
  - Logging messages print out wall clock time on Linux
  - EXPERIMENTAL: Add an implementation for About Client with Announce processing and handler registration

What have we done so far?

- 15.04
  - Property cache feature
  - Support for X.509 ECDSA digital certificates, which replace the X.509 RSA certificates
  - Remove excessive name transfer for all sessions.
  - Observer Functionality
  - It is now possible to build for Android without requiring OpenSSL
  - Add to About C API: a second announce_using_datalistener method was added
  - Add support for detecting idle RNs allowing the RN to shut down automatically if no LNs have used its services for a while
  - Support for iOS 8.1
  - Allow the setting of the UNMARSHAL_TIMEOUT
  - AJTCL Router Selection
  - Support for X.509 ECDSA digital certificates, which replace the X.509 RSA certificates
  - AJTCL adds information string to reply messages
  - Add support for specifying NVRAM file location on Windows
  - EXPERIMENTAL: UDP Transport for TC <-> RN connections. Thin Client to Routing Node version of a UDP-based transport

What have we done so far?

- 15.09
  - Security 2.0 was added as a Developer Preview Feature
  - ARDP/UDP transport between TC and RN is product ready
  - Change Win7 LN SC Application to connect to the Win10 named pipe, when running on Win10
  - Add About feature to JavaScript binding
  - SC/TC: Expand conversation hash to include all parts of auth conversation
  - Reorganize TC code base
  - Add APIs to support asynchronous method replies

What have we done so far?

- Total tickets addressed
  - Tasks: 376
  - Bugs: 1,693
- Collaboration
  - Stats
    - Chart of commits across contributing companies
    - Estimated number of committers
  - Talk to project teams
    - Security 2.0
    - AllJoyn.js
    - DDAPI

What have we learned?

- What have we learned
  - Process
    - Pain points
      - Testing efforts
      - …
    - Improvements
      - Alliance engaged 3rd party vendor for testing
      - Triage
      - Ticket classification
  - Collaboration
    - Lessons
    - Engagement (ask vs mail list vs JIRA)
      - Parallel processes due to code mirroring on other sites

Status on open source test assets

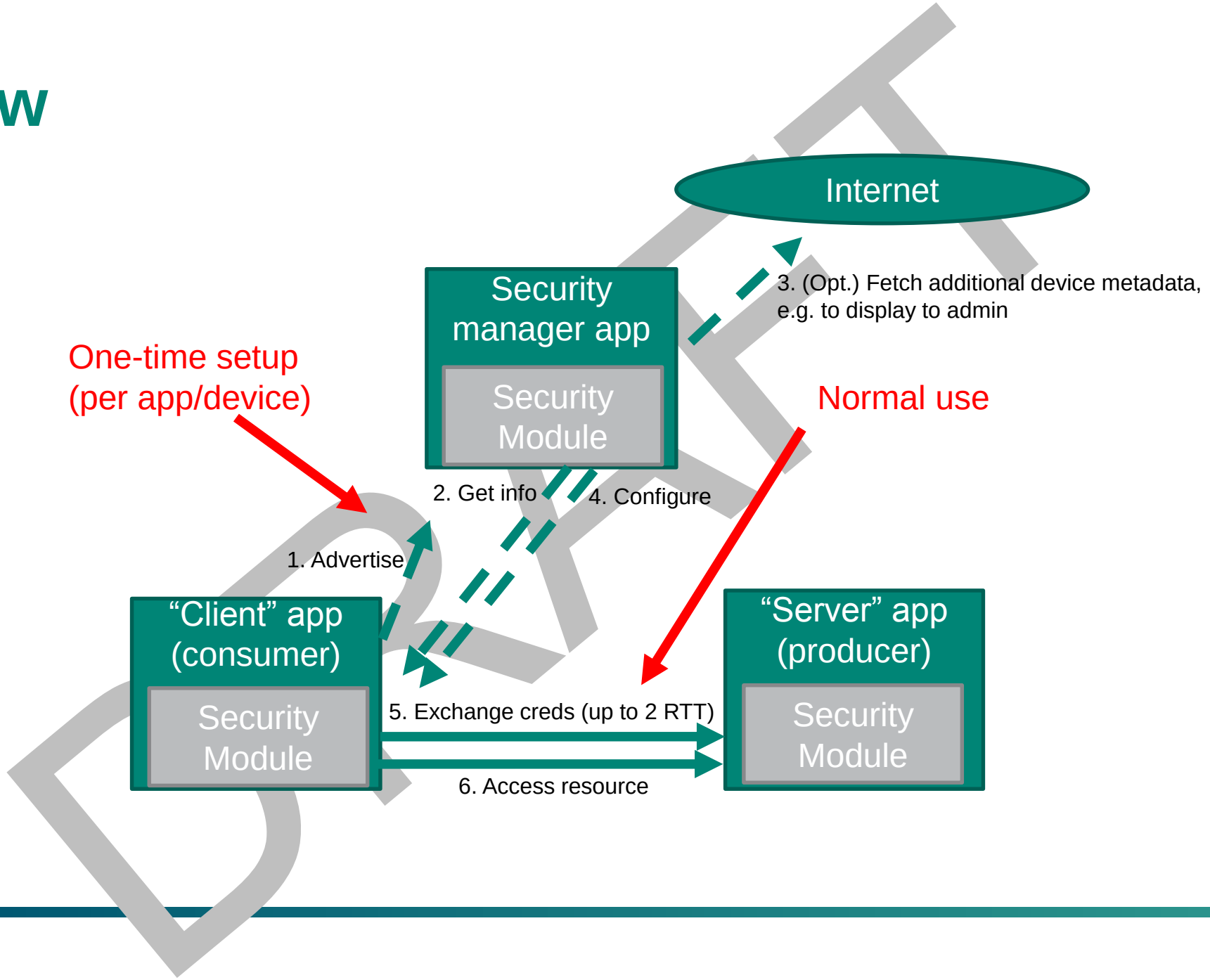
David McBride

Placeholder for test status slides

AllJoyn Security 2.0 Model

Dave Thaler
Software Engineer, Microsoft

Overview


Internet
Security manager app
Security Module
3. (Opt.) Fetch additional device metadata, e.g. to display to admin
One-time setup (per app/device)
Normal use
2. Get info
4. Configure
1. Advertise
"Client" app (consumer)
Security Module
"Server" app (producer)
Security Module
5. Exchange creds (up to 2 RTT)
6. Access resource

Security principals (1/2)

- Identity (“identity cert”) identifies a single app or device
  - An identity is actually a certificate chain
    - Each cert is signed by another cert, up to some root
  - Every application/device has its own identity
  - Root is the identity/group that runs/owns it
  - Delegation is permitted if one’s identity cert allows it
    - Delegation means you’re allowed to sign other certs
    - Typically means an assertion of ownership

Security principals (2/2)

- Security group identifies a set of apps/devices
  - Membership in a security group is also a certificate chain with a unique membership cert as a leaf
  - Every application/device has membership certs for 0 or more security groups of which it is a member
  - Can get a membership cert chain from a different root from your identity
    - E.g., your app/device can also be a member of someone else's group
  - Delegation is permitted if one's membership cert allows it
    - Delegation means you're allowed to admit others into the group

Certificate Revocation

- What if some device is sold, stolen, or compromised without being factory reset?
  - Certificates inherently have some lifetime
- Early revocation permitted via “Certificate Revocation List” checking
  - No dependency on runtime availability of some service
  - Not implemented in 15.09 release
  - Can also be mitigated by just updating policies on peers to disallow access by revoked certificate

Examples

- Users:
  - Dad (self-signed)
  - Mom (self-signed)
  - Son (signed by Mom)
- Devices:
  - TV: admin = Dad
  - DCR: admin = HomeAdmin
  - Son’s tablet: admin = Son
- Mom’s Security groups:
  - HomeAdmins: { Dad, Mom }
  - LivingRoomDevices: { TV, DVR }
- TV’s admin is dad but it’s still in Mom’s security group

Authentication

- An app has a set of trusted roots
- A peer's cert chain is "trusted" to be what it claims to be if a trusted root appears anywhere in that certificate chain
  - This doesn't mean it's allowed to do anything yet though… that's "authorization" which we'll come to next
- Identity cert chains exchanged at start of connection (usually mutual auth)
- Then exchange membership certs that chain up to a cert in peer's identity cert chain
  - Thus, you only disclose memberships relevant to that peer
- Another option is the "real estate agent" scenario where your membership cert can be pre-provisioned in peer app/device and yours

Authorization

- AllJoyn has both an “ACL” model and a “capabilities” model
  - Both checks must succeed for a remote call to succeed
- Access Control Lists (ACLs) are on your resources and control what peers are allowed to do
  - Analogous to a guest list at a private gathering
- Capabilities are what you are allowed to do (enforced by peer)
  - Analogous to a drivers license that says what you can drive
  - Capabilities help protect against compromised apps/devices

ACLs

- ACL (“policy”) are private between the app/device and its admin(s)
- Resources can be ACL’ed to a set of any of:
  - All (anonymous)
  - Any authenticated
  - Any authenticated that chains up to a given certificate authority
  - A specific security group
  - A specific identity (public key)
- ACL entries have separate flags for read vs write
- Optionally can also ACL who you will send outgoing calls to

Capabilities, similar to “app manifest”

- Each application has a list of what interfaces (not object instances), it can possibly access and expose
  - AllJoyn interfaces use a hierarchical resource naming scheme that includes DNS names for uniqueness, similar to an XML namespace
- Authorized set of capabilities get signed by same entity as your identity cert (referred to as a “security manager”)
- During one-time setup, security manager gets this, and authorizes it for all or some (possibly empty) subset of these as part of giving it identity & membership certs
- Capabilities presented along with one’s certs when making a connection
  - Just like presenting a drivers license both asserts identity and permission to drive

Bootstrapping (one-time-setup)

- New app/device advertises itself as unclaimed
  - In future, could even be passive advertisement (e.g., QR code)
  - Most common new device example is where device is a temporary WiFi SoftAP with a special SSID convention (or potentially IE's, etc. in future)
- Security Manager app sees it, and queries its potential capabilities
  - Usually uses some sort of PIN/passphrase scheme at this point, but could be anything
- (Opt.) Get trusted textual descriptions etc. to assist human in acknowledging granting of capabilities [not implemented in 15.09]
- Security Manager configures app/device:
  - "Onboarding" = configuring L2 network credentials (e.g., WiFi keys) if needed
  - "Claiming" = configuring
    - 1 or more trusted root1 certs
    - Identity cert chain
    - 0 or more membership cert chains
    - Signed capabilities
    - ACLs
- 1Manufacturer functionality such as app/firmware update might also use a manufacturer cert to verify code, but usually considered separate root from rest of functionality

16.04 feature set

Marcello Lioy

16.04 features

- 23 JIRA tickets in 5 categories
- Categories (with examples)
  - App Developer Usability
    - Fix the Logger so it can work with ETW on Windows
    - Support Android Studio
  - Core Developer Usability
    - Add functions for runtime creation of Aj_Objects
  - Security enhancement
    - Design and implement new password-based authentication mechanism
    - ECDSA Certificate generation APIs should be implemented for language bindings.
  - Optimization
    - Teach BusAttachment to keep track of registered AddMatch rules
  - System Evolution
    - Add support for extended introspection XML format

16.04 list of features as of 9/10/15

Key	Summary	Bucket
ASACORE-2034	Deadlock if max BusAttachment concurrency is reached	App Developer Usability
ASACORE-1993	Drop connection to router node if bus operations timeout	App Developer Usability
ASACORE-1759	Add support for a new callback that will inform a multi-point session joiner that the session host has accepted its join request	App Developer Usability
ASACORE-1556	Fix the Logger so it can work with ETW on Windows	App Developer Usability
ASACORE-2273	Alljoyn Thin Core API to get socket descriptor...	App Developer Usability
ASACORE-1930	Distributables should have QCC_OS_GROUP_ defined	App Developer Usability
ASACORE-1811	Returning an error name/message on calling SetProperty	App Developer Usability
ASACORE-1374	Support Android Studio	App Developer Usability
ASACORE-1112	BusObject does not contain a GetInterfaces API call	App Developer Usability
ASACORE-1166	API Call to reset changeId for About Annoucements	App Developer Usability
ASACORE-942	About feature should provide an implementation of generating a device ID for each platform	App Developer Usability
ASACORE-1065	Add functions for runtime creation of Aj_Objects	App/Core Developer Usability
ASACORE-2404	Add Mutex::AssertIsOwner() functionality	Core Developer Usability
ASACORE-2386	Add a platform independent QCC_ASSERT macro	Core Developer Usability
ASACORE-2005	Teach BusAttachment to keep track of registered AddMatch rules	Optimization
ASACORE-2364	General support for key types other than ECC NIST P-256	Security enhancement
ASACORE-2363	Store more data about trust anchors	Security enhancement
ASACORE-2055	Design and implement new password-based authentication mechanism	Security enhancement
ASACORE-1715	Private key protection support for using passphrase not applicable for ECDHE_ECDSA mechanism.	Security enhancement
ASACORE-1714	ECDSA Certificate generation APIs should be implemented for language bindings.	Security enhancement
ASACORE-1430	Packet header encryption	Security enhancement
ASACORE-2254	Add support for a "const" annotation for properties	System Evolution
ASACORE-964	Add support for extended introspection XML format	System Evolution

Thank You

Follow Us On

- For more information on AllSeen Alliance, visit us at: allseenalliance.org & allseenalliance.org/news/blogs