ALLSEEN ALLIANCE

Core Security Status Meeting

October 14, 2014

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.

Reminder:

This call is being recorded

Agenda

1. Contributor status
2. Technicolor discussion items
3. Security 2.0: Authorization Data Improvements
4. Review combined milestone schedule

Contributor Status

Notes

- Technicolor
  - Working on integrating QCE code
  - Proceeding well with the permission management interface
  - Android sample has been moved to Dec 1
    - To prepare for summit
- Microsoft
  - Beginning efforts for 14.12 release items
    - Design overview to the Core WG mail list shortly
    - On track for 14.12 schedule
  - Additional Security 2.0 effort begins once 14.12 is complete
- QCE
  - Test plan in place
  - Working on remaining items for permission management interface
  - Overall on track

Technicolor discussion items

Technicolor discussion items

- **Discussion**
  - Proposed changes to the Authorization Data format
    - Combine the actions under the interface or interface members instead of listed in the provider or consumer sections
    - Replacing the AllowAllExcept list with an explicit Deny action.
    - Is mutual auth always required?
  - Claiming process based on sessionless signal
  - Permission interface and BusAttachment creation
  - Limit number of BAs used by security manager (should be 1 BA per app)
  - Session set-up between app and security manager based on predefined port or About
- **Agreement**
  - Starting at 15.04, applications cannot disable security 2.0
    - Security manager is optional

Security 2.0: Authorization Data Improvements

Technicolor proposal

Current implementation (based on HLD)

- PermissionPolicy
  - Version
  - SerialNumber
  - Admin: Peer*
  - Providers: Term*
  - Consumers: ACL*
- Peer
  - Type: any/GUID/PSK/DSA/guild
  - Tag: ID/GuildAuthorithy/PSK
- Term
  - Peers: Peer*
  - ACL*
- ACL
  - Allow: Rule*
  - AllowAllExcept: Rule*
- Rule
  - ObjPath
  - InterfaceName
  - InterfaceMember: Member*
- Member
  - Name
  - Type
  - ReadOnly
  - MutualAuth

Merge Providers and Consumers

Current

- PermissionPolicy
  - Version
  - SerialNumber
  - Admin: Peer*
  - Providers: Term*
  - Consumers: ACL*

Proposal

- PermissionPolicy
  - Version
  - SerialNumber
  - Admin: Peer*
  - Rules: Term*

- Merge Providers and Consumers section into one ‘rules’ section
- Applications will provide and consume the same interface (to avoid separate code paths for local and remote data)
- Avoid redundancy in policy
- Remove asymmetry between providers and consumers by also allowing peer specific rules for consumers

Core Security Status Call (SCSC) 10/14: proposal accepted

Simplify actions

Current

- ACL
  - Allow: Rule*
  - AllowAllExcept: Rule*
- Member
  - Name
  - Type
  - ReadOnly
  - MutualAuth

Proposal

- Member
  - Name
  - Type
  - Action
  - MutualAuth

- Determining allowed action is spread over multiple structures
- Merge them to one field: Action
- Four basic actions
  - D(eny): explicit deny to stop resolution of any other rule
  - P(rovide): allows to send signals, provide methods and properties
  - O(bserve): allows to retrieve signals and to GetProperties
  - M(odify): Observe + SetProperties and call methods
- Combinations are possible

SCSC 10/14: proposal accepted, assuming deny trumps any other rule

Require a specified Type

Current

- Member
  - Name
  - Type (optional)
  - ReadOnly
  - MutualAuth

Proposal

- Member
  - Name
  - Type (required)
  - ReadOnly
  - MutualAuth

- During the claiming procedure, the application can not be introspected to determine the type.
- Currently an order is implied in case of name clashes (first method call/signal, then property).
- To simplify look-up (one with specified type and one without), require a specified type.

SCSC 10/14: proposal accepted

Remove MutualAuth field

Current

- Member
  - Name
  - Type
  - ReadOnly
  - MutualAuth

Proposal

- Member
  - Name
  - Type
  - ReadOnly
  - ~~MutualAuth~~

- MutualAuth should be enabled by default
- Currently by default true for providing signals, but not for methods and properties
- Without enabling this by default, we can't live up to the manifest promise 'make sure app can only provide/consume data on AllJoyn it requested during install time'

SCSC 10/14: agreement to change default value to true (also for providing methods and properties), but to keep field as optional

Proposed Rule Structure

```
{
    "rules": {
        "org.allseenalliance.control.OnOff": [
            { "m.on": "C" },
            { "m.off": "C" }
        ],
        "org.allseenalliance.control.TV": [
            { "*": "C" }
        ],
        "org.allseenalliance.control.ParentalControl": [
            { "*": "D" }
        ],
       "org.allseenalliance.*": [
           { "*": "C" }
        ]
    }
}
```

- The outer map contains a mapping between interface names to a rule-set for that specific interface (or path).
- The rule-set for each interface is another mapping between members and actions.
- Each member name will be prefixed by its type to avoid duplicate look up (with/without specified type) in the resolution algorithm.
- Default rules can be defined using a wildcard character.

Proposed Resolution Algorithm

1. result = deny
2. while (ifn != null)
    2.1. if (ifn-rules = find ifn in rules)
        2.1.1. if (rule = find mbr in ifn-rules)
            2.1.1.1. result = rule.action
            2.1.1.2. return from while
        2.1.2 else
            2.1.2.1. mbr = "*"
    2.2. else
        2.2.1. ifn = remove last section + "*"
3. return result

Combined Milestones

Security 2.0 Combined High Level Schedule 1 of 4

- October
  - Technicolor
    - Proxy distribution agent – API definition Complete
  - QCE
    - Functional test planning begins Oct 13
- November
  - QCE
    - Permission management (SC/TC) DC Nov 21
    - Java android test applications (SC) DC Nov 21
    - IOS sample test applications (SC) DC Nov 21
  - Technicolor
    - Security manager utility v0.5 Nov 3

Security 2.0 Combined High Level Schedule 2 of 4

- December
  - Microsoft
    - Finalize Threat Analysis effort
    - Complete design documentation
    - Complete design reviews with Alliance
  - Technicolor
    - Android sample v0.5 Dec 1
    - Security Manager Utility v1.0 Dec 19
    - Android sample v1.0 Dec 19

Security 2.0 Combined High Level Schedule 3 of 4

- January
  - QCE
    - Finalize test plan and test cases Jan 12
    - Functional Core testing begins Jan 12
  - Technicolor
    - Security manager android sample application v1.5 DC Jan 12
  - Microsoft
    - Windows keystore implementation
    - Federated user identity prototype
    - Windows security manager sample application

Security 2.0 Combined High Level Schedule 4 of 4

- February
  - Microsoft
    - Optional credential management plug-in model
    - Test case documentation
    - Windows telemetry
  - Integration testing needs to begin
    - Requires additional planning and coordination between contributors
- March
  - Test effort completes Mar 31
- April
  - Security 2.0 included as part of the 15.04 Core release

Thank You

Follow Us On 

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