Argonaut Project:
Authorization Risk Assessment

Version 2.0
December30, 2015

Argonaut Project:
Authorization Risk Assessment

1. Introduction

The Argonaut Project is a multi-stakeholder initiative to advance the development and adoption of open, API-based interoperability in the health care industry. Specifically, the Argonaut Project aims to produce specifications and implementation guidance that will enable software vendors and healthcare organizations to implement software applications capable of querying electronic health records (EHRs) for discrete data elements and structured documents using RESTful services specified using HL7 Fast Healthcare Interoperability Resources (FHIR).[1]

Phase 1 of the Argonaut Project commenced by defining four use cases, selected as representing highly functional scenarios that present manageable security risks, and containing trust risks to within a single healthcare organization. These use cases were:

1. Patient uses provider-approved web application to access health data
2. Patient uses provider-approved mobile app to access health data
3. Clinician uses provider-approved web application to access health data
4. Clinician uses provider-approved mobile app to access health data

Phase 2 of the Argonaut Project added a fifth use case[2] – cross-organizational access. That is:

5. Clinician in “organization A” uses EHR-A to access patient data held by EHR-B, operated by “organization B”

Version 1.0 of this document reported on the risk assessment conducted for Phase 1 of the Argonaut Project; Section 2 comprises this initial report. Section 3 extends this risk assessment to address use case 5.

A critical factor in the success of the Argonaut Project is to assure that only authorized human users and software applications are able to access EHR data using the defined APIs, and that they are able to access only those resources for which they have been authorized. The OAuth 2.0 Authorization Framework[3], developed by the Internet Engineering Task Force (IETF), is well suited for this purpose and has been profiled by the open *SMART Platforms* initiative in several specifications.[4,5,6]

2. Phase 1 Risk Assessment – Application Authorization

2.1 Purpose

The purpose of this risk assessment was to examine the *SMART on FHIR* authorization specifications with respect to how they address security risks associated with the use of OAuth 2.0, and the OpenID profile of OAuth 2.0, for the identified use cases, and to identify changes warranted for the Argonaut application-authorization profile(s).

2.2 Approach

2.2.1 Sources

The risk assessment was based on threats, vulnerabilities, and mitigation recommendations provided in the following specifications:

- The OAuth 2.0 Framework,[3] Section 10 (Security Considerations)
- The OAuth 2.0 Threat Model and Security Considerations,[7] Sections 4 (Threat Model) and 5 (Security Considerations)
- The OAuth 2.0 Authorization Framework: Bearer Token Usage,[8] Section 5 (Security Considerations)
- OpenID Connect Core 1.0,[9] Section 16 (Security Considerations)

In addition, a security analysis[10] performed by the MITRE Corporation for the Department of Veterans Affairs was reviewed.

2.2.2 Process

The risk assessment process comprised the following four steps:

1. Review the security risks identified in the sources above, and assess their applicability to the four use cases defined for the Argonaut Project.[11]
2. Examine the *SMART on FHIR* Authorization profiles with respect to the mitigations recommended for each applicable security risk, and identify gaps.
3. For each gap, assess the readiness of the recommended mitigations, considering the maturity and adoptability factors and metrics developed by the Health Information Technology Standards Committee.[12]
4. Coordinate recommendations for additional mitigations.

2.3 Findings

Specific results of the risk assessment are given in Appendix A. Each row provides:

1. Reference to the source documentation
2. Description of the risk
3. Summary of recommendations contained in the source documentation
4. Description of whether and how the *SMART on FHIR* authorization profiles[1] address the risk
5. Description of changes implemented in the Argonaut authorization profile

2.4 Argonaut Profile Modifications

At the time of this assessment, the *SMART on FHIR* authorization profiles included two profiles: a confidential profile, for apps capable of protecting a secret used for authenticating themselves; and a public profile, for apps that are not capable of

[1] As of the start of the Argonaut Project.

protecting a secret. These profiles referenced a separate specification that provided details on scopes and launch context.

As the assessment proceeded, we noted a need for some high-level changes to enhance clarity and consistency with the OAuth 2.0 specifications, and to enhance implementability, with the ultimate objective of encouraging widespread adoption both within the Argonaut community and beyond. Section 5.1 descries these high-level modifications that were made to the *SMART on FHIR* profiles to create the single Argonaut profile.

Section 5.2 describes more granular, detailed changes that were made, directly traceable to the risk assessment, the results of which are provided in Appendix A.

2.4.1 Profile-Level Changes

2.4.1.1 Two profiles merged

At the time of this assessment, the *SMART on FHIR* authorization profiles included two profiles: a confidential profile, for apps capable of protecting a secret used for authenticating themselves; and a public profile, for apps that are not capable of protecting a secret. The two profiles were quite similar: both implemented the Authorization Code Grant model, which requires the client to obtain an authorization code that is then exchanged for an access token. The only real difference was that the confidential profile authenticated itself to the authorization server when it exchanged the code for an access token, and when it exchanged a refresh token for a new access token. So for simplicity in understanding and implementation, and ease of on-going profile maintenance, the Argonaut Project elected to merge the two profiles into a single profile, with differences in applicability between confidential and public clients clearly noted.

2.4.1.2 Separation of OAuth actors

The OAuth 2.0 profile identifies two central actors whose behaviors are central to the Argonaut use cases: the authorization server, which mediates applications' requests for access to resources and grants access consistent with organizational policy and end-user permissions, and the resource server, which holds the resources being accessed.

The *SMART on FHIR* profiles combined these two actors into one "EHR" actor, creating a possibility of misinterpretation for implementers. The Argonaut profile separately addresses these two actors as the "EHR authorization server" and the "EHR FHIR resource server."

2.4.1.3 Use of token to access resource

The *SMART on FHIR* profiles did not describe the final step – the app's use of the access token to access a resource. Although this step was shown in the sequence diagram, it was not described in the text.

The Argonaut profile adds a description of this final step – an app presenting a "bearer" token to the FHIR resource server. To counter the risk of token leakage, the app presents the token to the resource server within an Authorization header (see Appendix A).

2.4.1.4 Clarification of launch context and OAuth 2.0 execution sequences

The *SMART on FHIR* profiles began by describing the launch sequences for an app launching from within the context of an EHR, and as a stand-alone ("native") application. This description duplicated content that later appeared in the authorization description, which made it somewhat challenging to interpret. In addition, each of the two profiles included a single sequence diagram.

For increased clarity, five new sequence diagrams were developed: 1) EHR launch; 2) standalone launch; 3) app request for authorization; 4) app exchange of authorization code for access token; and 5) app use of bearer token to access FHIR resources, and exchange of refresh token for new access token. The description of the app launch was separated from the specification of OAuth2 authorization request and use. Finally, a new, brief overview of the entire process was added.

2.4.1.5 User identity authentication

When an app desires to authenticate the identity of an end-user, it may do so by asking the authorization server for an OpenID Connect token attesting to the user's identity. The app sends the request to the authorization server by including two parameters in the scope of the authorization request: openid and profile. Although the *SMART on FHIR* profiles included this information in the scopes and launch context specification, requesting an ID token was not integrated into the profile descriptions or examples.

The Argonaut profile integrates the request for end-user identity authentication into the body of the profile, with the app receiving an ID token in response to the request.

2.4.2 Specific Changes

The following summarizes the specific modifications that were made in translating the *SMART on FHIR* authorization profiles into the Argonaut profile, as a result of this risk assessment.

2.4.2.1 End-user authorization

The *SMART on FHIR* profiles included asking for end-user authorization for an app to access EHR data as "optional." Although the decision of whether to ask for end-user authorization is a policy decision made by the authorization server, we added the general rule that if an EHR launches an app (confidential or public) for an authenticated user who has explicitly requested the launch, end-user authorization is "optional." Else the authorization server "should" request the user's authorization.

2.4.2.2 App responsibility to protect

Although only confidential apps have a provable ability to protect secret information, both confidential and public apps have a responsibility to implement measures to protect authorization codes, access tokens, ID tokens, and refresh tokens from unauthorized access, use, and modification. The Argonaut profile includes specific guidance for app developers to help them build apps that fulfill this obligation. For example, app developers are warned not to store bearer tokens in cookies that are transmitted in the clear, and to assure than any values the app receives are not injected with executable code (e.g., SQL). The app developer is also given guidance on how an app should behave should it receive a FHIR resource containing a ‘reference” to a resource hosted on a different server.

2.4.2.3 Refresh tokens

OAuth 2.0 enables an authorization server to take several actions in response to an app’s request for authorization to access a protected resource. The authorization server can refuse the request, it can issue an access token only, or it can issue a refresh token along with the access token. An app can exchange a refresh token for a new access token, when an access token expires. *SMART on FHIR* profiles included only issuance of an access token (or denial of access), which tends to encourage issuance of tokens with long expiration times, which carries attendant risks, as the authorization server has no opportunities to retract the access token during the token’s lifetime. Refresh tokens enable authorization servers to issue access tokens with shorter expiration times, along with a refresh token that can be exchanged for a new access token, giving the authorization server opportunities to refuse the exchange.

The Argonaut profile includes the issuance and exchange of refresh tokens, and recommends assigning shorter expiration times to access tokens, and longer to refresh tokens. Refresh tokens require the same protection as access tokens, both at rest and in motion. Confidential clients are required to authenticate themselves when exchanging a refresh token for a new access token.

To protect against the risk of a counterfeit resource server phishing for access tokens, the Argonaut profile includes an additional parameter in the app’s request for authorization. The `aud` parameter passes the FHIR resource server’s endpoint URI to the authorization server, along with the request for access. The authorization server then can validate that the URI is a known and trusted value prior to returning an authorization code.

2.4.2.4 “State” requirement

A primary OAuth 2.0 risk is cross-site request forgery (CSRF) in which an attacker causes the user agent (e.g., browser, mobile device) to follow a malicious URI instead of the app’s authorized URI, resulting in the app’s using the attacker’s authorization code or access token to access the attacker’s resource instead of the protected resources the app actually intended to access. The use of the `state` parameter to help ensure continuity of a client’s “session” throughout the OAuth flow is a key protection against CSRF.

The *SMART on FHIR* profiles “recommended” the use of the `state` parameter, but did not require it. The Argonaut profile “requires” the use of the `state` parameter wherever it is valid in the OAuth flow. The app then can assure that any request sent to its redirection URI includes a `state` value binding the request to the user-agent’s `state`; the authorization server can assure that any authorization request includes this `state` value; and can include his value when it redirects the user-agent back to the app.

2.4.2.5 “Aud” parameter

Access token phishing by a counterfeit resource server is another OAuth 2.0 risk was not addressed by the original *SMART on FHIR* profiles. In this threat scenario, an attacker pretends to be a resource server from which the app may want to retrieve a resource. A client sends a valid access token to the counterfeit resource server, which then uses that token to access other services and resources from the legitimate resource server, on the end-user's behalf.

A countermeasure for this attack is to have the app include in the request it sends to the authorization server, the endpoint URL of the resource server the app talked to. This is accomplished using the `audience` (`aud`) parameter. The authorization server then associates this endpoint URL with the access token it returns to the client. When the token is presented to the legitimate resource server, the resource server validates the association, enabling it to detect tokens presented by a counterfeit server.

The Argonaut authorization profile requires that the `aud` parameter be included in an app's request for authorization.

3. Phase 2 Risk Assessment – Cross-Organizational Authorization (Phase 2)

3.1 Purpose

The purpose of the Phase 2 risk assessment was to evaluate risks associated with the application programming interfaces (APIs) that enable the use of the OAuth 2.0 protocol to support cross-organizational authorization, and to identify appropriate and reasonable countermeasures for addressing these risks.

3.2 Approach

3.2.1 Sources

The sources for the Phase 2 assessment included the sources used for the Phase 1 assessment, listed in section 2.2.1 above, plus:

- JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants[13]

3.2.2 Process

Whereas the Phase 1 risk assessment focused on modifications to the pre-existing *SMART on FHIR* authorization profiles, the cross-organizational authorization profile was new development.

The profile, and its risk assessment, focused on the OAuth 2.0 application programming interfaces (APIs) that enable one organization's EHR authorization server to request authorization for a FHIR resource held by another organization, and to then retrieve that resource. The scope of the risk assessment did not include internal policies or technical implementations associated with the use of these APIs.

The risk assessment process comprised the following steps.

1. Review the threats identified in the source documents to determine their applicability to cross-organizational authorization.

2. For each applicable threat, review the recommended and potential countermeasures with respect to their potential value and implementability.
3. Incorporate into the profile countermeasures likely to be deemed reasonable and feasible by most healthcare organizations, with additional countermeasures of potential value to some as "optional."

3.3 Findings

Specific results of the Phase 2 risk assessment are given in Appendix B. Each row in the table provides.

1. Reference to source documentation
2. Description of the risk
3. Summary of recommendations contained in the source document
4. Description of how the cross-organizational authorization profile addresses the risk

3.4 Discussion

Because the cross-organization data access profile addresses only the APIs used to enable the exchange of FHIR resources between two organizations, risks relating to internal communications between an end-user and an authorization server, or between the a FHIR resource server and its associated authorization server, and risks relating to end-user behaviors and devices, are outside the scope of this profile. Similarly, the access policies an organization enforces with respect to its own FHIR resources, and policies associated with the trust agreements between the two sharing organizations, are organizational decisions outside the scope of this risk assessment. Also, simply by exposing APIs, both organizations create a target for denial-of-service (DOS) attacks, which will need to be addressed by each organization.

The grant type selected for the cross-organizational authorization profile is the "jwt-bearer" grant type defined in RFC7523, *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants.* RFC7523 profiles a more general framework (RFC7521[14]) that extends OAuth 2.0 to allow for the use of assertions (i.e., security tokens) as client credentials and/or authorization grants. The "jwt-bearer" grant type uses a JSON Web Token (JWT) bearer token to request an OAuth 2.0 access token and to authenticate the identity of the requester.

Use of the jwt-bearer grant type avoids risks relating to URI redirection, including cross-site request forgery. The JWT authorization and authentication tokens are passed directly from the EHR authorization server of the requesting organization to the EHR authorization server of the organization holding the FHIR resource. No redirection is involved.

The cross-organizational profile enables the issuance and use of a bearer access tokens, which authorizes release of a resource to the "bearer" of the token, as defined in the token's scope. Several measures are used to prevent the use of the token by "bearers" other than the authorized entity. These include:

1. All exchanges are between mutually trusting organizational entities – the EHR-A authorization server and the EHR-B authorization server, and the EHR-A authorization server and the EHR-B resource server – in accordance with pre-established business agreements.
2. Before any JWT tokens or access tokens are exchanged between two servers, the sender authenticates the identity of the receiver and then establishes an encrypted TLS link over which the tokens are transmitted.
3. The originator of the authorization request (EHR-A) includes with its request a JWT authenticating the originator's (EHR-A's) identity.
4. Access tokens are issued with a limited lifetime; the profile recommends an expiration time of one hour or less.
5. Refresh tokens are not supported.
6. Replay-attack detection is enabled through the use of an assertion ID (jti), and issuance and expiration time attributes (iat and exp)

4. References

[1] Health Level 7. Fast Healthcare Interoperability Resources (FHIR). Available from http://www.hl7.org/implement/standards/FHIR-Develop/index.html (accessed 2/20/15)

[2] Argonaut Project. Use Cases for Argonaut Project. Version 1.1. July 31, 2015. Available from http://argonautwiki.hl7.org/images/e/ec/Argonaut_UseCasesV1-1.pdf (accessed 12/2/15)

[3] Internet Engineering Task Force. RFC 6749. The OAuth 2.0 authorization framework. Oct 2012. Available from http://www.rfc-base.org/rfc-6749.html. (accessed 2/20/15)

[4] SMART Platforms. SMART on FHIR authorization: confidential clients. Available from http://docs.smartplatforms.org/authorization/confidential/. (accessed 2/20/15)

[5] SMART Platforms. SMART on FHIR authorization: public clients. Available from http://docs.smartplatforms.org/authorization/confidential/. (accessed 2/20/15)

[6] SMART Platforms. SMART on FHIR: scopes and launch context. Available from http://docs.smartplatforms.org/authorization/scopes-and-launch-context/. (accessed 2/20/15)

[7] Internet Engineering Task Force. RFC 6819. The OAuth 2.0 threat model and security considerations. Jan 2013. Available from https://tools.ietf.org/html/rfc6819. (accessed 2/24/15)

[8] Internet Engineering Task Force. RFC 6750. The OAuth 2.0 authorization framework: Bearer token usage. Oct 2012. Available from https://tools.ietf.org/html/rfc6750. (accessed 2/24/15)

[9] OpenID connect core 1.0 incorporating errata set 1. Available from http://openid.net/specs/openid-connect-core-1_0.html. (accessed 2/24/15)

[10] Russell, M. Secure RESTful interface security analysis and guidance. MITRE Corporation. 2014. Available from http://secure-restful-interface-profile.github.io/pages/. (accessed 2/24/15)

[11] Use cases for Argonaut Project. Draft Version 0.2. January 18, 2015.

[12] Baker D, J Perlin, J Halamka. Evaluating and classifying the readiness of technical specifications for national standardization. *JAMIA*. Available from http://jamia.oxfordjournals.org/content/jaminfo/early/2014/12/17/amiajnl-2014-002802.full.pdf. (accessed 2/24/15)

[13] Internet Engineering Task Force. RFC 7523. JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. Available from https://tools.ietf.org/html/rfc7523 (accessed 12/2/15)

[14] Internet Engineering Task Force. RFC 7521. Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants. Available from https://tools.ietf.org/html/rfc7521. (accessed 12/2/15)