Argonaut Project:
Bulk Data Export
Security Risk Assessment Report

Version 1.0

January 11, 2018

Dixie B. Baker, Security Assessment Lead

Josh Mandel and Daniel Gottlieb, Subject Matter Experts

Table of Contents

1 Introduction

1.1 Purpose

This document is the final report from a security risk assessment performed with respect to the draft specifications for authorizing, exporting, and downloading FHIR bulk data files. This work was performed at the request of the HL7 Argonaut Project.

1.2 Background

The Argonaut Project has undertaken the development of a technical specification defining application programming interfaces (APIs) through which an authenticated and authorized backend service ("client") can asynchronously request large volumes of health information (i.e., FHIR resources) relating to a specified group of individuals, receive status information regarding progress in the generation of the requested files, and retrieve the exported files. This specification will have broad application for providers and organizations responsible for improving, protecting, and managing the health of populations.

To support bulk-data export, and other pre-authorized accesses to FHIR resources, the project also includes the development of an authorization profile for an API to enable a pre-authorized client to request and receive an access token.

1.3 Scope

This security risk assessment identifies risks associated with the Bulk Data Export APIs supporting the asynchronous authorization for and retrieval of large-volume data sets.

The scope includes risks associated with[1]:

- Assurance of the identity of the requesting client and the authenticity of the request, including client registration and sharing of public encryption keys
- Requests for and issuance of an access token authorizing the access
- Service calls from the client to the FHIR resource server
- Client query for status of data extraction
- Delivery of bulk FHIR data files to the requester

This scope includes both identified vulnerabilities and specification ambiguities that could produce vulnerabilities in implementations.

The scope of this security risk assessment includes risks associated with the workflow shown in Figure 1, which includes the following data flows:

- Between client (i.e., backend service) and FHIR Authorization Server – used for obtaining access tokens

[1] The scope specified in the task statement included extension support. However, neither of the Bulk Data Export specifications supports extension.

- Between client and FHIR Resource Server (i.e., bulk data service) – used for requesting the extraction of FHIR resources into exportable files
- Between client and Status Server – used for querying status of data extraction process
- Between client and File Server – used for downloading extracted data files

1.4 Assumptions

This security risk assessment assumed that:

- All contractual and legal agreements necessary to enable data sharing between the data holder and the client requesting bulk-data extraction would have been fully executed prior to the use of these APIs (e.g., Business Associate Agreements, Data Use Agreements, service contracts), and are therefore outside the scope of this assessment.
- The Bulk Data Export subject matter expert (SME) team would support this assessment by:
    - Providing information and clarification in response to specific questions
    - Providing guidance in identifying relevant resources, including developer/implementer discussions
    - Reviewing and providing inputs to work in progress
    - Initiating and overseeing remediation activities responsive to identified risks, as appropriate
- Some identified risks might warrant changes to a specification while this assessment was under way; the risks that motivate such changes would be documented in this assessment report, along with the remediation actions taken.

2 Approach

2.1 Sources

The following sources of information regarding the Bulk Data Export specifications were used:

- Bulk Data Wiki, http://wiki.hl7.org/index.php?title=201809_Bulk_Data
- FHIR Bulk Data Overview presentation, https://docs.google.com/presentation/d/14ZHmam9hwz6-SsCG1YqUlQnJ56bvSqEatebltgEVR6c/edit#slide=id.p
- SMART Bulk Data Server Reference Implementation, https://bulk-data.smarthealthit.org
- Example FHIR Downloader (Backend Service/Bulk Data) App, https://github.com/smart-on-fhir/sample-apps-stu3/tree/master/fhir-downloader

- Zulip Bulk Data Discussion, https://chat.fhir.org/#narrow/stream/bulk.20data
- FHIR Bulk Data Access Implementation Guide (top-level document), https://github.com/smart-on-fhir/fhir-bulk-data-docs
- DRAFT SMART Backend Services: Authorization Guide, https://github.com/smart-on-fhir/fhir-bulk-data-docs/blob/master/authorization.md
- DRAFT FHIR Bulk Data Export Implementation Guide, https://github.com/smart-on-fhir/fhir-bulk-data-docs/blob/master/export.md

The following sources relating to OAuth 2.0 and its associated security risks and remedies were used:

- The OAuth 2.0 Authorization Framework, RFC 6749, https://tools.ietf.org/html/rfc6749
- The OAuth 2.0 Threat Model and Security Considerations, RFC 6819, https://tools.ietf.org/html/rfc6819
- OAuth 2.0 Security Best Current Practice. Draft-ietf-oauth-security-topics-10. https://tools.ietf.org/html/draft-ietf-oauth-security-topics-10
- The OAuth 2.0 Authorization Framework: Bearer token usage, RFC 6750, https://tools.ietf.org/html/rfc6750.
- JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants, RFC 7523, https://tools.ietf.org/html/rfc7523

The Bulk Data Export subject matter experts -- specifically, Josh Mandel and Daniel Gottlieb -- also were significant sources of information regarding the FHIR Bulk Data Export specifications and the collaborative process that produced them.

2.2 Process

This security risk assessment process included the following activities.

1. The Assessment Lead reviewed available resources to understand the draft APIs and to identify potential associated security risks.
2. As the review proceeded, the Assessment Lead discussed the identified risks with the Bulk Data Export SME team, and collaboratively planned for any needed remediations.
3. The Assessment Lead generated GitHub Pull Requests recommending specific changes to the top-level document, the *SMART Backend Services: Authorization Guide*, and the *Bulk Data Export Implementation Guide*.
4. The SME Team reviewed the Pull Requests, and initiated further discussion, as required.
5. The SME Team completed remediation activities and closed the Pull Requests.
6. The steps above were repeated until all identified issues were resolved, and Pull Requests merged (or otherwise resolved).

7. The Assessment Lead documented findings in the initial draft of the Security Assessment Report, including any remedial actions that were taken during the course of the security assessment.
8. The Assessment Lead completed the draft Bulk Data Export Security Risk Assessment Report.
9. The Bulk Data Export SME team reviewed the draft Security Risk Assessment Report and provided feedback through written comments and clarifying discussion.
10. The Assessment Lead incorporated team comments into the draft Report.
11. The Assessment Lead delivered the final draft Bulk Data Export Security Risk Assessment Report to the Project Manager for review and acceptance.
12. The Assessment Lead was available to present interim and final results to the Argonaut Steering Committee at the Project Manager's request.

3 Findings

3.1 Workflow-Associated Risks

Security risks associated with the workflow specified in the *Bulk Data Export Implementation Guide* ("*Bulk Data Export Guide"*) with authorization implemented in accordance with the *SMART Backend Services: Authorization Guide* ("*Authorization Guide"*) are given in Figure 1 below. (This document will refer to these two documents collectively as the "Bulk Data Export specifications.") Note that the workflow described in the *Bulk Data Export Guide* could be implemented using alternative means of access control, such as mutual Transport Layer Security (TLS) or signed universal resource identifiers (URIs), which could alter or eliminate some of these risks.

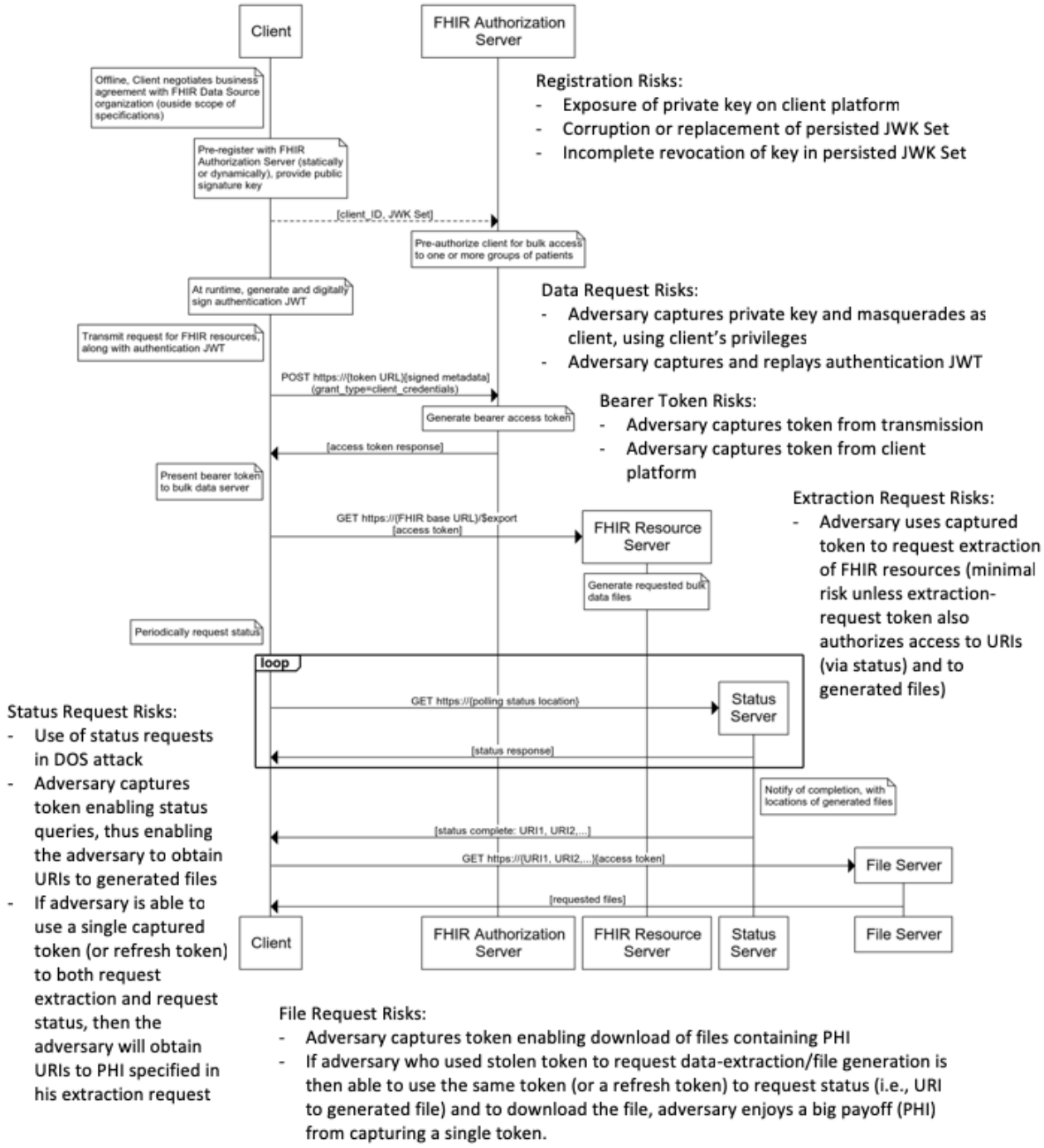

Figure 1. Security risks associated with Bulk Data Export workflow

3.1.1 Registration Risks

Prerequisite to the Bulk Data Export workflow are business agreements between the data-requesting organization and the data holder. These agreements provide the foundational trust required to enable bulk-data export and are outside the scope of both of the Bulk Data Export specifications.

Once the agreements are in place, the client software is registered with the data holder's authorization server. This process includes the client's providing its public encryption key to the data holder. As specified in the *Authorization Guide*, the key is provided as a JSON Web Key (JWK) within a set of keys (JWK Set). The client retains the private key. The JWK Set may be passed to the data holder directly, or the client may provide a URI to the JWK Set. The authorization server records an access rule pre-authorizing the client to one or more FHIR resource groups.

If an adversary captures the private key (e.g., using one of the methods discussed in the OAuth 2.0 documentation), it will be able to masquerade as the client to request data export, status, and exported files. If an adversary is able to control the content returned when the Server dereferences the client's JWKS URL, then the adversary will likewise be able to masquerade as the client. If the client organization provides the JWK Set directly, then the security risks relate to the data holder's need to protect the keys from corruption or replacement, and to assure that the keys used are current and have not been revoked. Paragraph 4.1 below contains further detail regarding these risks.

3.1.2 Data Request Risks

In order to request that FHIR resources be extracted in bulk to one or more data files, the client uses the OAuth 2.0 Client Credentials protocol, which requires that the client authenticate itself to the FHIR authorization server so that the server can apply the pre-programmed access rules in mediating the request. The client accomplishes this by generating a JWT and digitally signing it – that is, encrypting a hash of the JWT claims using the client's private key. This authentication JWT is then sent to the authorization server, along with the scope of data the client is requesting.

The client is responsible for protecting both the private key and the authentication JWT. If an adversary captures the private key, it can masquerade as the client and use all the privileges afforded the client, including pre-authorized access rights. An adversary who captures the authentication JWT can replay it in submitting its own request for data.

3.1.3 Bearer Token Risks

A "bearer" token is an opaque string of characters with the property that any party in possession of the token (the "bearer") can use the token in the same ways that any other party in possession of it might use it. Use of a bearer token does not require that the "bearer" prove that it is the party to whom the token was issued. Security risks associated with the use of bearer tokens are discussed in *RFC 6750, The OAuth 2.0 Authorization Framework: Bearer Token Usage*. Since whoever gains possession of a bearer token is able to use it, any adversary who is able to capture a bearer token, while at rest with the client or in transmission, can use it.

3.1.4 Extraction Request Risks

If an adversary is able to capture a bearer token authorizing the bearer to request access to the specified scope, the adversary will be able to request that the FHIR resource server extract the scope of FHIR resources authorized by the token. Because a resource server extracts data into files for later download, rather than synchronously returning the resources to the client, the associated risk is minimal – unless the adversary is able to use the same token (or an associated refresh token) to request status (which will provide URIs to the generated files) and file download.

3.1.5 Status Request Risks

Once the extraction request has been launched, the client is able to request status updates from a status server. A principal risk is that an adversary might repeatedly issue status requests to cause a denial of system services.

Also, if an adversary is able to capture an access token enabling status queries, the adversary will ultimately receive a "Complete Status" response containing the URIs pointing to the files containing the extracted protected health information (PHI). This response also will indicate whether an access token is required to download the extracted files. (See paragraph 4.5 below.)

3.1.6 File Request Risks

Once the requested data have been extracted into one or more files, the files are made available for download to an authorized client. The Bulk Data Export specifications do not dictate implementation, so this server may be the hosted with the FHIR resource server from which resource-extraction was requested, the server from which status is obtained, or it may be a server specifically configured for bulk-file downloads. Regardless of the specific implementation, this API holds the potential for the biggest pay-off for an adversary, and therefore is highly likely to be targeted.

The principal risk is that an adversary is able to capture the access data that will enable the adversary to download of the files. Depending on the implementation, an adversary may capture an access token (e.g., an OAuth 2.0 bearer token), a private key enabling the adversary to masquerade as the client in a mutual-TLS protocol, or a pre-signed URI to a container holding the downloaded files.

A related risk is that the file server itself is compromised. That is, an adversary successfully captures administrative rights to the file server and thereby takes possession of the sensitive files contained therein. Such adversary might continue to deliver files in response to client requests, so that neither the client nor the FHIR server is aware of the take-over. Meanwhile, the adversary puts the PHI to use for its own devious purposes.

3.2 Specific Risks and Countermeasures

A total of 10 specific security risks (i.e., instances of the above, plus other specification-related risks) were identified, as described in Appendix A. Each risk is associated with one or more exchanges and risk levels assessed in accordance with the definitions shown in Table 1. All of the risks identified were remediated or mitigated by changes made to the *Authorization Guide* and/or *Bulk Data Export Guide*.

Risk Level	Risk Level Definitions
High	Risk warrants immediate implementation of strong corrective measures.
Medium	Risk warrants corrective actions to be taken within a reasonable period of time.
Low	Risk likelihood is low and/or potential loss may be tolerated. Bulk Data Export leadership should determine whether to take corrective actions or to accept risk.

Table 1. Three levels of risk severity.

In the risk table given in Appendix A, each row provides:

1. Risk Identifier (R*x*)
2. Specification (*Authorization Guide* and/or *Bulk Data Export Guide)*
3. Description of Potential Risk Exposure
4. Risk Association and Severity Level
   - "C <-> AS" are exchanges between Client and FHIR Authorization Server
   - "C <-> RS" are exchanges between Client and FHIR Resource Server
   - "C <-> SS" are exchanges between Client and Status Server
   - "C <-> FS" are exchanges between Client and File Server
5. Countermeasures Implemented

4 Summary of Modifications Made

A total of 21 GitHub Pull Requests were submitted, discussed, approved, and merged as part of this Risk Assessment. Appendix B contains a log of the PRs merged. The columns in the log show:

1. Submit Date
2. Pull Request (PR) Number
3. Relevant Specification (*Authorization Guide* and/or *Bulk Data Export Guide*)
4. Title
5. Brief Description

All of the PRs listed in the table were discussed, many were modified, and ultimately all were merged into the relevant Bulk Data Export specification, as indicated. The following

sections summarize the major changes that were made to the two specifications, with specific risks referenced for each.

4.1 Bulk-Data-Export Security Considerations

Ref Risks: R5, R7, R8, R11

A conscious effort was made to specify access-control requirements only in the *Authorization Guide*, and to incorporate these requirements by reference in the *Bulk Data Export Guide*. However, transport risks and risks associated with the file server are unique to the bulk-data-export flow and therefore needed to be addressed in the *Bulk Data Export Guide*. For this purpose, a Security Considerations section was added to the *Bulk Data Export Guide*. This section contains the requirement for Transport Layer Security (TLS) for all client-server exchanges, recommends OAuth 2.0 access control as defined in the *Authorization Guide*, and discusses the need to protect the file server from external threats. (See section 4.5 below for further detail.)

4.2 Encryption Key Management and Protection

Ref Risks: R1, R2, R3, R4

Much of the security protection incorporated in the Bulk Data Export specifications relies heavily on public-key cryptography, which involves a pair of encryption keys with the inherent quality that if one key is used to encrypt a data string, the other is required in order to decrypt the encrypted string (i.e., "ciphertext"). One of the keys in the key pair, called the "public key," is made openly available, while the other key, the "private key," is kept secret. The key pair are used differently depending upon the security attribute the sender is seeking to achieve. If secrecy between a sender and a receiver is desired, the sender uses the receiver's public key to encrypt the message, so that only the receiver is able to decrypt the ciphertext using the private key. If the sender desires to prove that she originated a message, she encrypts it using her private key, and the receiver uses the sender's public key to decrypt the ciphertext. The validity of public-key encryption rests on the secrecy of the private key; it is essential for the holder of the private key to afford it very strong security protection. The effectiveness of public-key encryption rests on the integrity of the public key; if it is replaced with an adversary's public key, then the receiver can be fooled into thinking that a sender is the legitimate client when in fact it is the adversary. Thus, public keys also require strong security protection to assure their integrity.

Within the Bulk Data Export workflow (see Figure 1), the client uses public-key encryption to authenticate its identity to the FHIR authorization server responsible for mediating access to FHIR data. The *Authorization Guide* assumes that the trust foundation between the requesting organization and the data holder have been established through the execution of appropriate legal agreements. Thus, the first exchange described in the *Authorization Guide* is the registration of the backend service ("client") with the data holder's authorization server.

The *Authorization Guide* does not dictate the method used to register a client (i.e., static or dynamic), but does specify two alternatives for sharing the client's public key, which is represented as a JSON Web Key (JWK) within a JWK Set: providing the data holder a copy of the JWK Set, or providing the data holder a URI pointing to the JWK Set maintained by the client organization. At the start of this security assessment, sharing a URI was labeled

the “preferred” method, and sharing a copy of the JWK Set was “allowed, not preferred.” Recognizing the need to protect the integrity of public keys, we changed the weightings to “strongly preferred” (sharing URI) and “strongly discouraged” (sharing copy of JWK Set) and explained the attendant risks.

In addition, we added a requirement for the client to protect the private key from disclosure and corruption.

4.3 Transport Protection

Ref Risk: R5

Both PHI and other sensitive information are passed between the client and the FHIR authorization server, FHIR resource server, status server, and file server, including PHI embedded in URIs and files, authentication JWT, and bearer tokens. Thus, it is essential that the identity of the server be authenticated, and that the sensitivity and integrity of all information exchanged between a client and the authenticated server be protected. This level of protection is achieved using Transport Layer Security (TLS). The OAuth 2.0 specification (incorporated by reference in both Bulk Data Export specifications) requires TLS transport protection for all exchanges. However, since implementations of the APIs defined in the Bulk Data Export Guide may include servers that use access-control protocols other than OAuth, we added to both specifications an explicit requirement for TLS protection for all exchanges. Mutual TLS, wherein both the client and server are required to authenticate their identities, is optional.

4.4 Minimum Necessary

Ref Risk: R6

The Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule requires that:

> “When using or disclosing protected health information or when requesting protected health information from another covered entity or business associate, a covered entity or business associate must make reasonable efforts to limit protected health information to the minimum necessary to accomplish the intended purpose of the use, disclosure, or request.” [§164.502(b)(1)]

The OAuth 2.0 Authorization Framework carries the minimum-necessary requirement as well, stating that "clients SHOULD request access tokens with the minimal scope necessary."

By definition, bulk data exports may be quite large, and their scopes quite broad. These scopes may be “necessary” for their intended usage; nonetheless, they may offer an attractive potential pay-off for an adversary.

The Bulk Data Export specifications assume that foundational trust agreements have been negotiated and agreed upon between the client organization and the data holder. This assumption includes an agreement to request, and to provide upon request, the minimum necessary PHI, in accordance with the HIPAA Privacy Rule. Both specifications incorporate by reference the OAuth 2.0 Framework. The *Authorization Guide* requires that the client specify the scope of data being requested, and acknowledges that this scope may be less

than that authorized. This assessment added a requirement for the FHIR authorization server to limit scopes to those specifically authorized for the client.

4.5 Access Control through Bulk Data Workflow

Ref Risks: R7, R8, R9, R10

As shown in Figure 1, the risk of PHI exposure through the Bulk Data Export workflow becomes increasingly serious progressing across the three APIs (i.e., kick-off, status request, file download request). If an adversary manages to capture an access token or authentication JWT during the kick-off phase, the adversary will be able to request the extraction of FHIR resources, and to have the set of resources for which the authentic client has been authorized extracted. However, the Bulk Data Export flow does not return data directly to the client, but rather extracts the data into files for later download. So, the risk of PHI exposure for an adversary who captures an access token and presents it to a FHIR resource server is minimal, as the adversary would possess neither the extracted data nor the location of the data files.

If an adversary is able to capture an access token after the file extraction has begun, the adversary may be able to query for the status of the extraction process – and possibly launch a denial-of-service (DOS) attack. Ultimately, the "Status – Complete" response will provide the adversary the URIs to the extracted files – thus, knowledge of the location of the PHI.

Ultimately, if an adversary is able to capture the URIs to the extracted files and possesses an active access token that is recognized by the file server, the adversary will be able to download the extracted files containing PHI.

To counter these risks, a data holder will need to ensure that an adversary who possesses an access token that enables the adversary to request that data be extracted is not able to use that same token to request status and file download. Given that bulk-data extracts are asynchronous and are likely to require a considerable amount of time, the *Authorization Guide* assures that the same access token cannot be used for all three APIs by requiring that the lifetime of all access tokens be limited to 5 minutes ("exp" value of no more than 300 seconds), and by recommending that refresh tokens not be issued. This will help assure that an adversary will not be able to use the same access token through the entire Bulk Data Export workflow.

To protect against replay of an authentication JWT, the *Authorization Guide* requires that the authorization server confirm that the "jti" value of each authentication JWT has not been previously used. To protect against an adversary's use of status requests to launch a DOS attack, the *Bulk Data Export Guide* requires that servers keep an accounting of status requests and generate an error if a potential DOS attack is detected.

Recognizing that the bulk data export workflow could be implemented across several servers and that the status and file servers could use access-control mechanisms other than OAuth 2.0, the *Bulk Data Export Guide* was somewhat confusing and ambiguous concerning the need for an access token for each API. For example, the "Complete Status" response section specified a required field called accessTokenRequired, defined as a "boolean value indicating whether downloading the generated files will require an authentication token," and included a "note" describing instances when an access token may not be required. Later, when specifying requirements for file download, the Guide

stated that if the value of accessTokenRequired field is “true,” the requester must provide an “access token.” The wording of the accessTokenRequired field definition referred to an “authorization token,” rather than an “access token,” creating ambiguity, and the wording seemed to imply that requiring a token might be an exceptional case, rather than the default – even though the file download carries the highest risk of unauthorized disclosure of PHI of all of the data-export APIs.

The wording of the accessTokenRequired field was modified such that value of this field must be “true” if both the file server and the FHIR API server control access using OAuth 2.0 bearer tokens. The value may be “false” for file servers that use access-control schemes other than OAuth 2.0, such as downloads from Amazon S3 bucket URIs or verifiable file servers within an organization's firewall.

Also, a Security Considerations section was added to include the TLS requirement (see paragraph 4.2 above) and the requirement that for each specified request, the client must provide proof of authorization. Implementers of RESTful implementations are encouraged to implement OAuth 2.0 access management in accordance with the *Authorization Guide*, while non-RESTful implementations may use authorization schemes other than OAuth 2.0, such as mutual-TLS or signed URLs (i.e., accessTokenRequired value of “false”).

Other than the Security Considerations section in the *Bulk Data Export Guide* and the accessTokenRequired field for file download, security requirements for bulk-data export are specified in the *Authorization Guide*, and incorporated by reference in the *Bulk Data Export Guide*.

4.6 Readability and Specificity

In addition to the changes that were made in response to identified risks, a number of changes were made to increase the readability and specificity in both of the Bulk Data Export specifications. These changes included:

- Adding section describing audience and scope
- Adding a list of referenced specifications
- Adding a terminology section describing the HL7 standard terminology used in the specifications

Wording in both specifications was modified to conform to the HL7 standard terminology. The ambiguous term “backend server” was replaced with “client” throughout both specifications, and references to “EHR” servers were modified to refer to FHIR servers. Also, paragraph organization was improved for ease of use.

Ambiguity is anathema to security, so throughout both specifications every effort was made to assure that requirements were clear, concise, and unambiguous.

Appendix A: Bulk Data Export Security Risks

Risk ID	Specification		Description of Potential Risk Exposure	Risk Association				Countermeausres Implemented
	Authz	Data Export		C <-> AS	C <-> RS	C <-> SS	C <-> FS	
R1	X	X	Adversary captures JWK Set transmitted from Client to FHIR Authorization Service.	L				Mechanism and protocol used for transmitting the JWK Set to the Authorization Server is outside the scope of the specification. The JWK Set that is passed to FHIR Authz Service contains only public keys, so exposure carries minimal risk.
R2	X		Client fails to keep private key confidential, enabling adversary to masquerade as client, with same privileges afforded the actual client -- including the ability to request data export, status, and exported files.	H				Added requirement for client to protect private key from disclosure and corruption.
R3	X		Adversary inserts its own public key into the JWK Set persisted by the FHIR server, or replaces the JWK Set with its own.	M				Spec "strongly recommends" that the service provide the URI to the JWK Set, rather than a copy of the Set, and requires protection against corruption.
R4	X		Client's privileges are revoked, but JWK Set is not updated	M				Spec "strongly recommends" that the service provide the URI to the JWK Set, rather than a copy of the Set.
R5	X	X	PHI or other sensitive information (e.g., signed JWT, bearer token) transmitted between a Client and a server (i.e., authorization, resource, status, or file) is intercepted by an adversary.	M	L	L	L	All transmissions among key entities are TLS protected, including server authentication, encryption, and integrity protection. Mutual TLS is optional. OAuth 2.0 (RFC6749), incorporated in Authorization Guideline, warns against transmitting sensitive information in URLs.

Risk ID	Specification		Description of Potential Risk Exposure	Risk Association				Countermeausres Implemented
	Authz	Data Export		C <-> AS	C <-> RS	C <-> SS	C <-> FS	
R6	X	X	The HIPAA Privacy Rule requires that information shared with a third party should be limited to the "minimum necessary." In the case of bulk data, scopes may be quite broad, increasing the potential volume of PHI that could be exposed should an adversary obtain an access token for requesting that FHIR resources be extracted by a Resource Service and/or that files be downloaded from a File Server.	M	M			OAuth 2.0 Framework is incorporated by reference in each specification, and states that "clients "SHOULD request access tokens with the minimal scope necessary." . Authorization Guide requires that client specify scope of data requested, which may be less than that authorized. Added requirement for FHIR Server to limit scopes to those specifically authorized for Client.
R7	X	X	The PHI exposure risks associated with the three APIs defined in the Bulk Export spec become increasingly serious progressing through the workflow. An adversary who captures an access token or authentication JWT during the kick-off phase will be able to request the extraction of FHIR resources, but the server would not return the resources directly to the adversary, thus minimizing the risk of PHI exposure. Capturing a token during the status-query phase could give the adversary the URIs pointing to the locations of the extracted files, but may not provide access to those files. However, capturing a token in the download phase could enable the adversary to download PHI.	L-M	L-M	M	H	Required that the lifetime of access tokens be limited to 5 minutes. Required that proof of authorization be provided for each request, while allowing alternative means of providing this proof (e.g., OAuth 2.0 tokens, mutual TLS, pre-signed Amazon S3 buckets).

Risk ID	Specification		Description of Potential Risk Exposure	Risk Association				Countermeausres Implemented
	Authz	Data Export		C <-> AS	C <-> RS	C <-> SS	C <-> FS	
R8		X	Specification included a boolean field called *accessTokenRequired* in the Status Reponse "Complete" and defined this as an indicator of whether an "authentication token" was required (true=required). However, the text indicated that a "true" in this field meant that an "access token" was required. This inconsistency could lead to inconsistent, and ineffective, implementations. Also, the field definition gave examples of when a token may not be required, but did not state when a token should be required -- thus implying that requiring an access token is "optional" at the most security-critical step, file download.			H	H	Clarified that accessTokenRequired field MUST be "true" for all RESTful implementations, and a "false" value requires an alternative authorization solution.
R9	X		Adversary captures authentication JWT and masquerades as client requesting access token.	M				FHIR authorization server is required to validate that jti value has not been previously used.
R10		X	Adversary launches a denial-of-service attack through status requests.			H		Requires that servers keep an accounting of status requests and to generate a Too_Many_Requests error if a potential DOS attack is detected.

Risk ID	Specification		Description of Potential Risk Exposure	Risk Association				Countermeausres Implemented
	Authz	Data Export		C <-> AS	C <-> RS	C <-> SS	C <-> FS	
R11		X	Adversary obtains administrative control over file server containing extracted files.					This risk is not associated with any of the exchanges/APIs addressed in this specification. However, it is a realistic threat that implementers will need to address. Currently, a CMS team is discussing this topic. Added content to the Security Considerations section describing the threat and the need for implementers to address it.

Appendix B: Log of Changes Made to Bulk Data Export Specifications

Submit Date	PR#	Specification		Title	Summary
		Authz	Bulk Data		
10/24/18	86	X		Profile audience & scope	Revised wording to clarify scope to include processes to registration/pre-authorization backend service, and for runtime acquisition of access token. Explicitly stated that profile's applicability is not restricted to retrieval of bulk data. Clarified conditions under which profile applies.
10/24/18	87	X		Underlying Standards	Added section listing standards on which this profile relies.
10/24/18	88	X		Conformance Language	Added Conformance Language section, which is identical to HL7 wording at https://www.hl7.org/fhir/conformance-rules.html#conflang. Note that this section should follow the Underlying Standards section proposed in pull-request #87.
10/24/18	89	X		Registering a SMART Backend Service	Edits throughout this section. In particular, edits to stress the advantages of conveying the JWK Set through a URL and the security risks associated with directly providing the JWK Set to the EHR.
10/24/18	90	X		Obtaining an Access Token	Changes throughout the section. NOTE CHANGE NEEDED IN DIAGRAM: Change expires_in value to 300 or less. Moved access token response details to Authorization Server Obligations Section.
10/26/18	92	X		Authorization Server Obligations	Changed section title and incorporated two sub-sections: Signature Verification and Issuing Access Tokens. Note that the table in the Issuing Access Tokens section was moved down from the original "Obtaining an Access Token" section. Pull Request #90 (Obtaining an Access Token) deleted this table, so to avoid confusion, I did not delete it again here.
10/26/18	93	X		Scopes and Worked Example	Incorporated changes, primarily for clarity.
11/5/18	94		X	Add Introductory Sections	Proposes to add the following introductory sections: - Audience and Scope - Referenced Specifications - Terminology (HL7 wording)

Submit Date	PR#	Specification		Title	Summary
		Authz	Bulk Data		
11/5/18	95		X	Add Security Considerations	Initially included requirements for TLS-secured channels, client authentication and authorization, and need for access token presented in Authorization header. Also requirement that resource server to inspect token, and detail regarding token policy. After discussion with SMEs, most of this content was stripped out of the *Bulk Data Guide*, as more appropriately placed in *Authorization Guide*.
11/7/18	96		X	Kick-Off Requests	Renamed section "Bulk Data Export Requests." Incorporated changes relating to kick-off request and response. Deleted "Authorization" section, as this content was incorporated into the newly proposed "Security Considerations" section (PR #95) and/or in the *Authorization Guide.*
11/7/18	97		X	Bulk Data Delete Requests	Minor changes. Submitted separately to preserve section.
11/9/18	98		X	Bulk Data Status Request	Modifications throughout section.
11/9/18	99		X	File Request	Clarified section. Also, deleted Out-of-Scope enumerated list, as this information was incorporated into the proposed new introductory section (PR #86).
11/19/18	101		X	Bulk Data Access READme	Incorporated clarification that backend client has been pre-authorized. Added brief descriptions of the draft specifications.
11/20/18	102			"Client" and "FHIR server" changes	Incorporated throughout Scopes section.
12/12/18	104	X		Issuing Access Tokens	Added content regarding decision-making w.r.t. the circumstances under which access tokens should accompany requests (i.e., content previously proposed for the Bulk Access Implementation Guide). Basically, a risk-management decision to be made by the data holder.

Submit Date	PR#	Specification		Title	Summary
		Authz	Bulk Data		
12/12/18	105		X	requiresAccessToken	Changed reference to "authentication token" to "access token." Also clarified that the Note was an example and not a rule.
12/14/18	106		X	Access Token Consistency	The specification was inconsistent regarding the requirement to present an access token with a request -- the language was inconsistent, and in one case (Delete Request) missing. For consistency and to eliminate any ambiguity, used the same wording for every operation: "The request MUST include a valid access token in the Authorization header (i.e., Authorization: Bearer {{token}}). See the Security Considerations section above." NOTE: This language was not incorporated into the Bulk Data Guide, as it was more appropriately placed in the Authorization Guide. Also, the wording of the description of the "requiresAccessToken" field seemed to minimize the need to include an access token for what is arguably the most security-critical operation -- downloading PHI. So I added wording that clarifies that if the download is RESTful (e.g., SMART), an access token is required, while acknowledging that some download options (e.g., signed URLs to Amazon S3 buckets) use alternative means of assuring that accesses are authorized.
12/18/18	107	X		Added Presentation of Access Token	Per discussions, added token presentation as final section in this profile. Also, for increased clarity, moved the Scopes section to immediately follow "Obtaining an Access Token," where scopes are first introduced; and made "Signature Verification" the first subsection under "Server Obligations," with "Issuing Access Tokens" as 2nd subsection.

Submit Date	PR#	Specification		Title	Summary
		Authz	Bulk Data		
01/08/19	111		X	accessTokenRequired Definition	Revised wording to clarify when this field MUST hold a value of "true" and when it MAY be "false. Note slight change in the first sentence from that proposed in the Report, in order to clarify the context of "require OAuth 2.0 bearer tokens." Also added "bearer" in the first sentence of the definition.
01/09/19	112		X	File Server Threat	Incorporated content relating to the need to protect files persisted in the file server. Includes reference to on-going work.