

MILE
Internet-Draft
Intended status: Standards Track
Expires: October 8, 2016

N. Cam-Winget, Ed.
S. Appala
S. Pope
Cisco Systems
April 6, 2016

XMPP Protocol Extensions for Use with IODEF
draft-ietf-mile-xmpp-grid-00

Abstract

This document describes the extensions made to Extensible Messaging and Presence Protocol (XMPP) [RFC6120] that enables the use of XMPP as a transport protocol for collecting and distributing any security telemetry information between and among network platforms, endpoints, and most any network connected device. Specifically, this document will focus on how these extensions can be used to transport the Incident Object Description Exchange Format (IODEF) information.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 8, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Glossary of Terms	3
1.2. What is XMPP-Grid?	5
1.3. Overview of XMPP-Grid	6
1.4. Benefits of XMPP-Grid	9
1.5. Example Workflow	10
2. XMPP-Grid Architecture	11
2.1. XMPP Overview	12
2.2. XMPP-Grid Protocol Extensions to XMPP	13
2.3. XMPP-Grid Controller Protocol Flow	13
2.4. XMPP-Grid Node Connection Protocol Flow	16
2.4.1. Authentication	16
2.4.2. Registration	16
2.4.3. Authorization	19
2.5. XMPP-Grid Topics Protocol Flow	22
2.5.1. Topic Versioning	23
2.5.2. Topic Discovery	23
2.5.3. Subtopics and Message Filters	23
2.6. XMPP-Grid Protocol Details	26
3. XMPP-Grid Compatibility with IODEF	32
4. IANA Considerations	33
5. Security Considerations	33
5.1. Trust Model	33
5.1.1. Network	34
5.1.2. XMPP-Grid Nodes	34
5.1.3. XMPP-Grid Controller	34
5.1.4. Certification Authority	34
5.2. Threat Model	35
5.2.1. Network Attacks	35
5.2.2. XMPP-Grid Nodes	36
5.2.3. XMPP-Grid Controllers	37
5.2.4. Certification Authority	38
5.3. Countermeasures	39
5.3.1. Securing the XMPP-Grid Transport Protocol	39
5.3.2. Securing XMPP-Grid Nodes	40
5.3.3. Securing XMPP-Grid Controllers	41
5.3.4. Limit on search result size	41
5.3.5. Cryptographically random session-id and authentication checks for ARC	42
5.3.6. Securing the Certification Authority	42
5.4. Summary	42
6. Privacy Considerations	43

7. Acknowledgements	43
8. References	44
8.1. Normative References	44
8.2. Informative References	44
Authors' Addresses	45

1. Introduction

XMPP-Grid is a set of standards-based XMPP [RFC6120] messages with extensions. It is intended for use as a secure transport and communications protocol ecosystem for devices and organizations to interconnect, forming an information grid for the exchange of formatted data (e.g. XML, JSON, etc). This document describes the extensions made to XMPP [RFC6120] that enables use of XMPP as a transport protocol for securely collecting and distributing security telemetry information between and among network platforms, endpoints, and most any network connected device.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.1. Glossary of Terms

AAA

Authentication, Authorization and Accounting.

CA

Certification Authority.

Capability Provider

Providers who are capable of sharing information on XMPP-Grid.

CMDB

Configuration Management Database.

IDS

Intrusion Detection System.

IPS

Intrusion Prevention System.

JID

Jabber Identifier, native address of an XMPP entity.

MDM

Mobile Device Management.

NAC

Network Admission Control.

PDP

Policy Decision Point.

PEP

Policy Enforcement Point.

Presence

XMPP-Grid node availability and online status on XMPP-Grid.

Publisher

A capability provider sharing content information to other devices participating on XMPP-Grid.

SIEM

Security Information and Event Management.

Subscriber

A device participating in XMPP-Grid and subscribing or consuming information published by Publishers on XMPP-Grid.

Sub-Topics

Topic created by XMPP-Grid Controller under a capability provider's topic based on message filter criteria expressed by subscribers.

Topics

Contextual information channel created on XMPP-Grid where a published message by the Publisher will be propagated by XMPP in real-time to a set a subscribed devices.

VoIP

Voice over IP.

XMPP-Grid

Set of standards-based XMPP messages with extensions, intended for use as a transport and communications protocol framework between devices forming an information grid for sharing information.

XMPP-Grid Controller

Centralized component of XMPP-Grid responsible for managing all control plane operations.

XMPP-Grid Connection Agent

XMPP-Grid client library that a XMPP-Grid node implements to connect and exchange information with other vendor devices on XMPP-Grid.

XMPP-Grid Node

Platform or device that implements XMPP-Grid Connection Agent to connect to XMPP-Grid and share or consume security data.

1.2. What is XMPP-Grid?

XMPP-Grid is a set of standards-based XMPP messages with extensions. It is intended for use as a transport and communications protocol framework for devices that interconnect with each other, forming a secure information grid.

XMPP-Grid enables secure, bi-directional multi-vendor exchange of contextual information between IT infrastructure platforms such as security monitoring and detection systems, network policy platforms, asset and configuration management, identity and access management platforms. XMPP-Grid can serve to securely exchange any contextual information. XMPP-Grid is built on top of XMPP [RFC6120], [RFC6121] which is an open IETF standard messaging routing protocol used in commercial platforms such as Google Voice, Jabber IM, Microsoft Messenger, AOL IM and a variety of IoT and XML message routing services. XMPP is also being considered as a means to transport IODEF [RFC5070]. XMPP-Grid is designed for orchestration of data

sharing between security platforms on a many-to-many basis for millions of end systems.

XMPP-Grid provides a security data sharing framework that enables multiple vendors to integrate to XMPP-Grid once, then both share and consume data bi-directionally with many IT infrastructure platforms and applications from a single consistent framework akin to a network-wide information bus. This reduces the need to develop to explicit, multiple platform-specific interfaces, thereby increasing the breadth of platforms that can interface and share security data. XMPP-Grid is also configurable thereby enabling partners to share only security data they want to share and consume only information relevant to their platform or use-case and to customize information shared without revising the interfaces. XMPP-Grid is data-agnostic enabling it to operate with virtually any data type such as IODEF [RFC5070].

1.3. Overview of XMPP-Grid

XMPP-Grid employs publish/subscribe/query operations brokered by a controller, which enforces access control in the system. This architecture controls what platforms can connect to the "grid" to share ("publish") and/or consume ("subscribe" or "query") contextual information ("Topics") (described in Section 3.3 and 3.5) such as security data needed to support MILE. The control of publish/subscribe/query operations is architecturally distinct from the actual sharing of the contextual information. Control functions are split into a logical control plane, whereas information exchange is considered a logical data plane. This separation enables scalability and customizability.

XMPP-Grid defines an infrastructure protocol that hides the nuances of the XMPP data plane protocol and makes the information sharing models extensible with simple intuitive interfaces. XMPP-Grid Nodes connect to the Grid using the XMPP-Grid Protocol. The XMPP-Grid Protocol makes use of the XMPP transport protocol and introduces an application layer protocol leveraging XML and XMPP extensions to define the protocol.

The components of XMPP-Grid are:

- o XMPP-Grid Controller (Controller): The Controller manages the control plane of XMPP-Grid operations. As such it authenticates and authorizes platforms connecting to the data exchange grid and controls whether or not they can publish, subscribe or query Topics of security data.

- o XMPP-Grid Connection Agent (Connection Agent): The Connection Agent enables the adopting Node to communicate with the Controller and other vendor platforms that have adopted XMPP-Grid. Through this communication privileges of the connecting platform--authorization to connect, publish, subscribe, query--are established. The Connection Agent is typically implemented as a client library.
- o XMPP-Grid Node (Node): A Node is a platform that has implemented the Connection Agent so that it can connect to an XMPP-Grid deployment to share and/or consume security data.
- o Data Repository: This is the source of security data available on the Grid and may be a network security platform, management console, endpoint, etc. XMPP-Grid does not mandate a specific information model, but instead remains open to transport structured or unstructured data. Data may be supplied by the security platform itself or by an external information repository.
- o Topic: An XMPP-Grid Topic defines a type of security data that a platform wants to share with other platform(s).

The operations carried out by XMPP-Grid to exchange security data are:

- o Grid Connect: This is a Controller operation that authenticates a Node that has implemented the Connection Agent to establish a connection with the XMPP-Grid. Once authenticated, authorization policies on the Controller establish a Node's privileges on the XMPP-Grid such as the right to undertake publish, subscribe or query operations explained below.
- o Publish Topic: Security information is made available when a XMPP-Grid enabled platform "publishes" a "Topic". This operation is authorized by the Controller and communicated to the connecting platform via the Connection Agent.
- o Topic Discovery: Nodes on a XMPP-Grid discover Topics of security data relevant to them by searching the Topic directory available within the XMPP-Grid deployment. The Controller maintains such a Topic directory for every instance of XMPP-Grid.
- o Subscribe to Topic: A Node seeking to consume security information "subscribes" to a Topic that provides the security information it seeks to serve its use-case. This operation has its authorization checked by the Controller and communicated with the connecting platform via the Connection Agent.

- o Query: This operation enables a Node to request a specific set of security data regarding a specific asset (such as a specific user endpoint) or bulk output history from a Topic over a specific span of time. Such queries can be carried out node-to-node or by querying a central data repository. Query structure is adaptable to match the information model in use.

XMPP-Grid is used to exchange security context data between systems on a 1-to-1, 1-to-many, or many-to-many basis. Security data shared between these systems may use pre-negotiated non-standard/native data formats or may utilize an optional common information repository with a standardized data format, such as IODEF. XMPP-Grid is data format agnostic and accommodates transport of whatever format the end systems agree upon.

XMPP-Grid can operate in the following deployment architectures:

- o Broker-Flow: An XMPP-Grid control plane brokers the authorization and redirects the Topic subscriber to Topic publisher directly. In this architecture, the Controller only manages the connection; the security data flow is directly between Nodes using data formats negotiated out-of-band.
- o Centralized Data-Flow: An XMPP-Grid maintains the data within its optional centralized database. In this architecture, the Controller provides a common information structure for use in formatting and storing security context data, such as IODEF, and directly responds to Node publish and Subscribe requests.
- o Proxy-Flow: An XMPP-Grid is acting as proxy, collecting the data from the publisher(s) and presenting it to the subscriber directly. This is used for ad-hoc queries.

Within the deployment architecture, XMPP-Grid may be used in any combination of the following data exchange modes. The flexibility afforded by the different modes enables security information to be exchanged between systems in the method most suitable for serving a given use-case.

- o Continuous Topic update stream: This mode delivers in real-time any data published to a Topic to the Nodes that are subscribed to that Topic.
- o Directed query: This mode enables Nodes to request a specific set of security information regarding a specific asset, such as a specific user endpoint.

- o Bulk historic data query: This mode enables Nodes to request transfer of past output from a Topic over a specific span of time.

1.4. Benefits of XMPP-Grid

Benefits of XMPP-Grid can be summarized on two fronts: 1) end-user benefits, 2) benefits for adopting vendors.

Benefits for end-users deploying security services based on XMPP-Grid security context information sharing capabilities are derived from the results that come with standardization including:

- o Consolidating relevant security event data from multiple systems to the "right console at the right time".
- o Cross-vendor interoperability out-of-the-box, when using a standard data format.
- o Coordinated security response across multiple products from multiple vendors, ranging from endpoint security to AAA, NAC, IDS/IPS, Data Loss Prevention, firewalls to infrastructure such as SIEM, CMDB, physical access control systems.
- o Customer product choice and flexibility. No need to buy all security products from one vendor.

Adopting XMPP-Grid security data sharing capabilities provides a number of benefits for adopting vendors, especially when compared to proprietary interfaces, such as:

- o Integrate the XMPP-Grid Connection Agent once to interface with many platforms, simultaneously by subscribing or publishing relevant security data
- o Security information shared is configurable (via Topics) based on relevance to specific use-cases and platforms
- o Only sharing relevant data enables both publishing and subscribing platforms to scale their security data sharing by eliminating excess, irrelevant data
- o Integrated authorization and security ensures only appropriate XMPP-Grid operations are executed by permitted platforms
- o Ability to share security data in native or structured formats enables data model flexibility for adopting vendors

- o Flexibility, adaptability to evolve to address new use cases over time. Utilize data-agnostic transport protocol or the extensible schema that allows for easy support for vendor-specific data.

1.5. Example Workflow

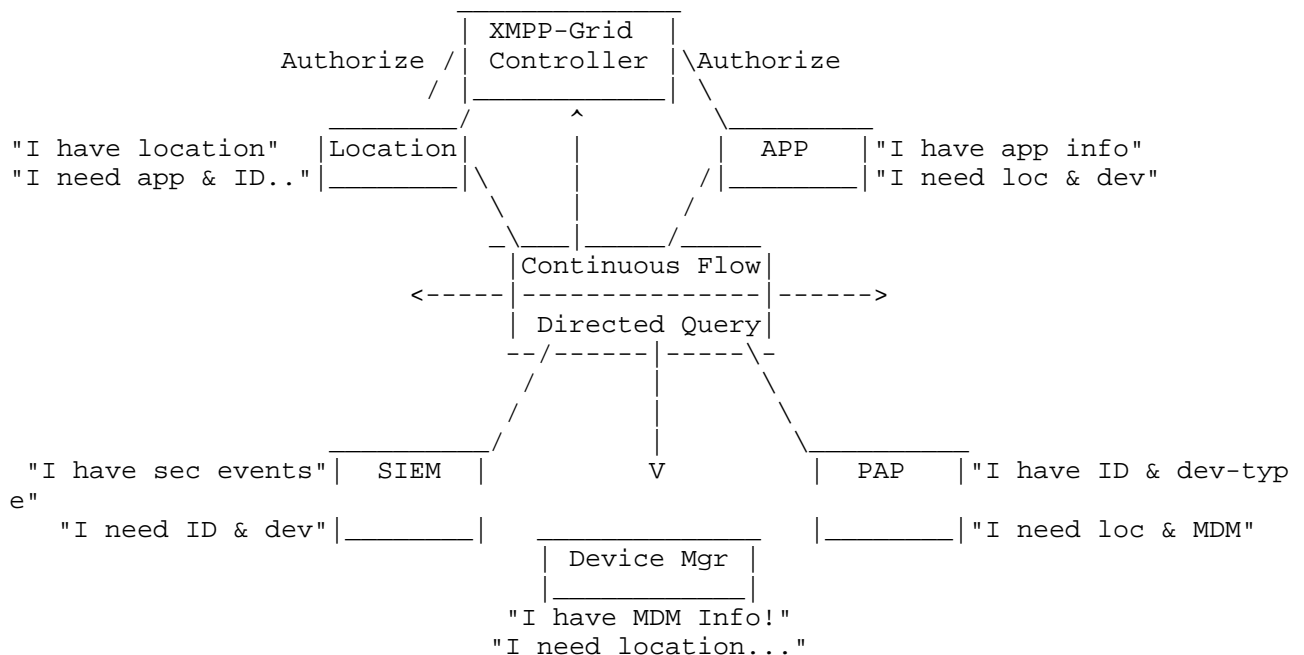


Figure 1: Typical XMPP-Grid Workflow

- XMPP-Grid Controller establishes a grid for platforms wanting to exchange security data.
- A platform (Node) with a source of security data requests connection to the Grid.
- Controller authenticates and establishes authorized privileges (e.g. privilege to publish and/or subscribe to security data Topics) for the requesting Node.
- Node may either publish a security data Topic, subscribe to a security data Topic, query a Node or Topic, or any combination of these operations.
- Publishing Nodes unicast Topic updates to the Grid in real-time. The Grid handles replication and distribution of the Topic to

subscribing Nodes. A Node may publish multiple Topics, thereby allowing for customized relevance of the security data shared.

- f. Subscribing Nodes receive continuous real-time stream of updates to the Topic to which they are subscribed.
- g. Any Node on the Grid may subscribe to any Topics published to the Grid (as permitted by authorization policy), thereby allowing for one-to-one, one-to-many and many-to-many meshed security data sharing between Nodes.

2. XMPP-Grid Architecture

XMPP-Grid is a communication fabric that facilitates secure sharing of information between network elements and networked applications connected to the fabric both in real time and on demand.

XMPP-Grid uses XMPP servers that operate as a cluster with message routing between them, for data plane communication. XMPP-Grid uses a control plane element, the XMPP-Grid Controller, that is an external component of XMPP for centralized policy-based control plane.

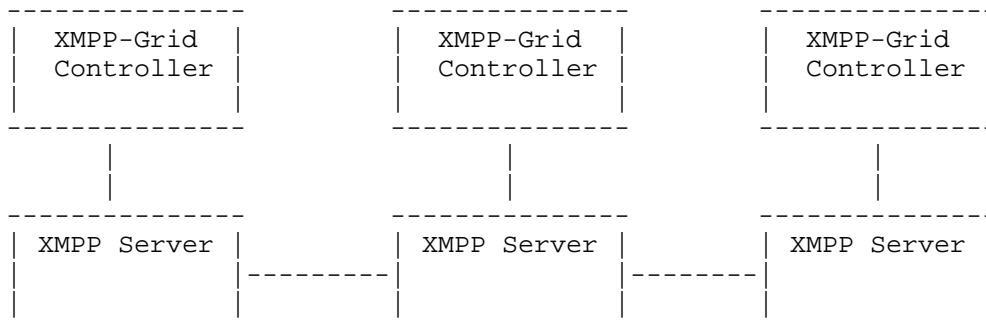


Figure 2: XMPP Server and XMPP-Grid Cluster Architecture

The connected Nodes, with appropriate authorization privileges, can:

- o Receive real-time events of the published messages from the publisher through Topic subscriptions
- o Make directed queries to other Nodes in the XMPP-Grid with appropriate authorization from the Controller
- o Negotiate out-of-band secure file transfer channel with the peer

This model enables flexible API usage depending on the Nodes' contextual and time-sensitivity needs of security information.

2.1. XMPP Overview

XMPP is used as the foundation message routing protocol for exchanging security data between systems across XMPP-Grid. XMPP is a communications protocol for message-oriented middleware based on XML. Designed to be extensible, the protocol uses de-centralized client-server architecture where the clients connect to the servers securely and the messages between the clients are routed through the XMPP servers deployed within the cluster. XMPP has been used extensively for publish-subscribe systems, file transfer, video, VoIP, Internet of Things, Smart Grid Software Defined Networks (SDN) and other collaboration and social networking applications. The following are the 4 IETF specifications produced by XMPP working group:

- o [RFC6120] Extensible Messaging and Presence Protocol (XMPP): Core
- o [RFC6121] Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence
- o [RFC3922] Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)
- o [RFC3923] End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)

XMPP offers several of the following salient features for building a security data interexchange protocol:

- o Open - standards-based, decentralized and federated architecture, with no single point of failure
- o Security - Supports domain segregations and federation. Offers strong security via Simple Authentication and Security Layer (SASL) [RFC4422] and Transport Layer Security (TLS) [RFC5246].
- o Real-time event management/exchange - using publish, subscribe notifications
- o Flexibility and Extensibility - XMPP is XML based and is easily extensible to adapt to new use-cases. Custom functionality can be built on top of it.
- o Multiple information exchanges - XMPP offers multiple information exchange mechanisms between the participating clients -

- o
 - * Real-time event notifications through publish and subscribe.
 - * On-demand or directed queries between the clients communicated through the XMPP server
 - * Facilitates out-of-band, direct communication between participating clients
- o Bi-directional - avoids firewall tunneling and avoids opening up a new connection in each direction between client and server.
- o Scalable - supports cluster mode deployment with fan-out and message routing
- o Peer-to-peer communications also enables scale - directed queries and out-of-band file transfer support
- o XMPP offers Node availability, Node service capability discovery, and Node presence within the XMPP network. Nodes ability to detect the availability, presence and capabilities of other participating nodes eases turnkey deployment.

The XMPP extensions used in XMPP-Grid are now part (e.g. publish/subscribe) of the main XMPP specification [RFC6120] and the presence in [RFC6121]. A full list of XMPP Extension Protocols (XEPs) [RFC6120] can be found in <http://xmpp.org/extensions/xep-0001.html> .

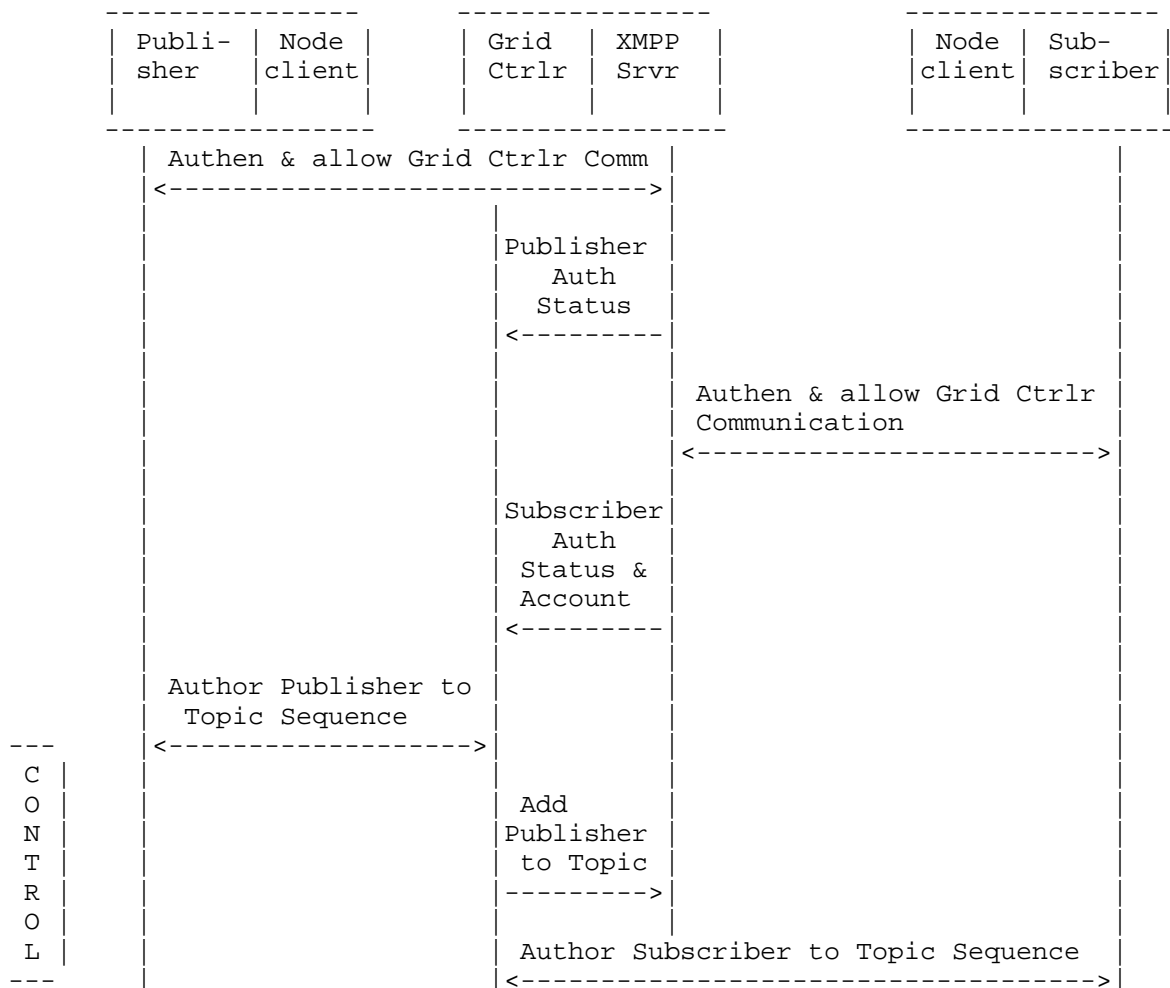
2.2. XMPP-Grid Protocol Extensions to XMPP

XMPP-Grid defines an infrastructure protocol that hides the nuances of the XMPP data plane protocol and makes the information sharing models extensible with simple intuitive APIs. XMPP-Grid Nodes connect to the Grid using the XMPP-Grid Protocol. The XMPP-Grid Protocol makes use of the XMPP transport protocol and introduces an application layer protocol leveraging XML and XMPP extensions to define the protocol. The capability providers on the Grid extend the XMPP-Grid Protocol infrastructure model and define capability specific models and schemas, allowing a cleaner separation of infrastructure and capabilities that can run on the infrastructure.

2.3. XMPP-Grid Controller Protocol Flow

At the heart of the XMPP-Grid network, the XMPP-Grid Controller serves as the centralized policy-based control plane element managing all Node authentications, authorizations, capabilities/Topics and their subscription list. XMPP-Grid Controller manages all control

aspects of the Node communication (including management) with the XMPP-Grid and other participating Nodes with mutual trust and authorizations' enforcement. XMPP-Grid Controller is a component of XMPP server and programs the data plane XMPP server with Node accounts, account status, XMPP Topics that are dynamically created and Topic subscriptions. This is analogous to File Transfer Protocol (FTP) that has control and data plane communication phases. Once the Node requests are authenticated and authorized in the control plane phase by the Controller, the Controller removes itself from the data flow. All data plane communication then occurs between the Nodes, publishers and subscribers of XMPP Topics happen at the XMPP data plane layer.



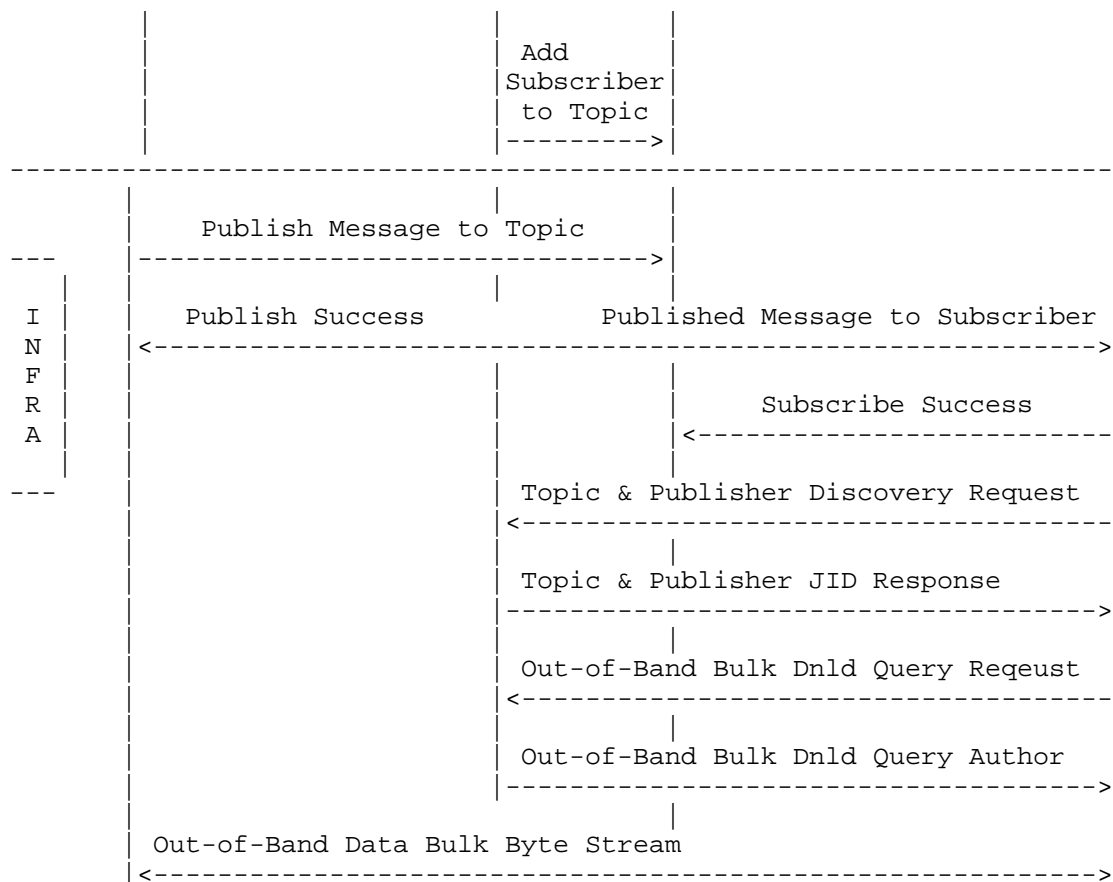


Figure 3: XMPP Controller Message Flow

Through a centralized authorization model, XMPP-Grid Controller provides -

- o Visibility into "who is connecting", "who is accessing what"
- o Node account management with provisions to add, delete or disable accounts, and with provisions to auto or manual approve Node account approval requests during the Node registration phase
- o Centralized, policy-based authorization, providing "who can do what" for publish-subscribe, directed peer-to-peer queries or for bulk out-of-band transfers between participating Nodes

- o Topics and subscription list management with provision to enable or disable Topics
- o Dynamic creation of sub-Topics within the main Topic depending on attributes of interest from the requesting Node
- o Ability to perform message filters on the published messages

2.4. XMPP-Grid Node Connection Protocol Flow

Nodes connecting to XMPP-Grid go through the phases of authentication, registration and authorization before they can participate in information exchange on XMPP-Grid.

2.4.1. Authentication

The communication between the Node and the XMPP-Grid Controller is cryptographically encrypted using TLS. XMPP-Grid uses X.509 certificate-based mutual authentication between the Nodes and Controller. Internally, XMPP uses Simple Authentication and Security Layer (SASL)[RFC4422] External mechanism to authenticate and establish secure tunnel with the Nodes, allowing the XMPP-Grid Controller to rely on this capability offered by XMPP. If the Node certificate does not pass the validation process, the connection establishment is terminated with the error messages defined by the XMPP standard. On successful authentication, XMPP SASL component extracts the Node certificate and Node username to the Controller for registration.

2.4.2. Registration

Once a Node has been authenticated and a secure tunnel has been successfully established, the Nodes will register their accounts with the Controller and Nodes provide their username to the Controller as part of the registration request. XMPP-Grid supports manual registration (requires explicit approval of the Node account) and mutual authentication trust-based auto-approval registration in order to provide additional trust and usability options to the administrator. The administrator may map the Nodes to the Node groups to add additional level of validation and trust, and enforce Node group based authorization. This allows the certificate-username-group trust to get uniquely establishment for each Node and duplicate registration requests using the same username will be rejected.

During the registration process, the Controller restricts all Node communication with the XMPP-Grid and only Node to Controller communication is allowed. Once the Node is successfully registered,

the Controller lifts the restriction and allows the Nodes to communicate on XMPP-Grid after it passes the authorization phase. It should be noted that the registered and authorized Nodes could publish, subscribe or query to multiple XMPP Topics between login and logout to XMPP-Grid. Multiple Node applications running on a Node could use one XMPP-Grid Node to connect to XMPP-Grid. The XMPP-Grid Node should support Node applications' subscription to Topics and should multiplex messages on its connection to XMPP-Grid. If a Node application wants to be identified explicitly on XMPP-Grid, a new XMPP-Grid Node connection to XMPP-Grid is required.

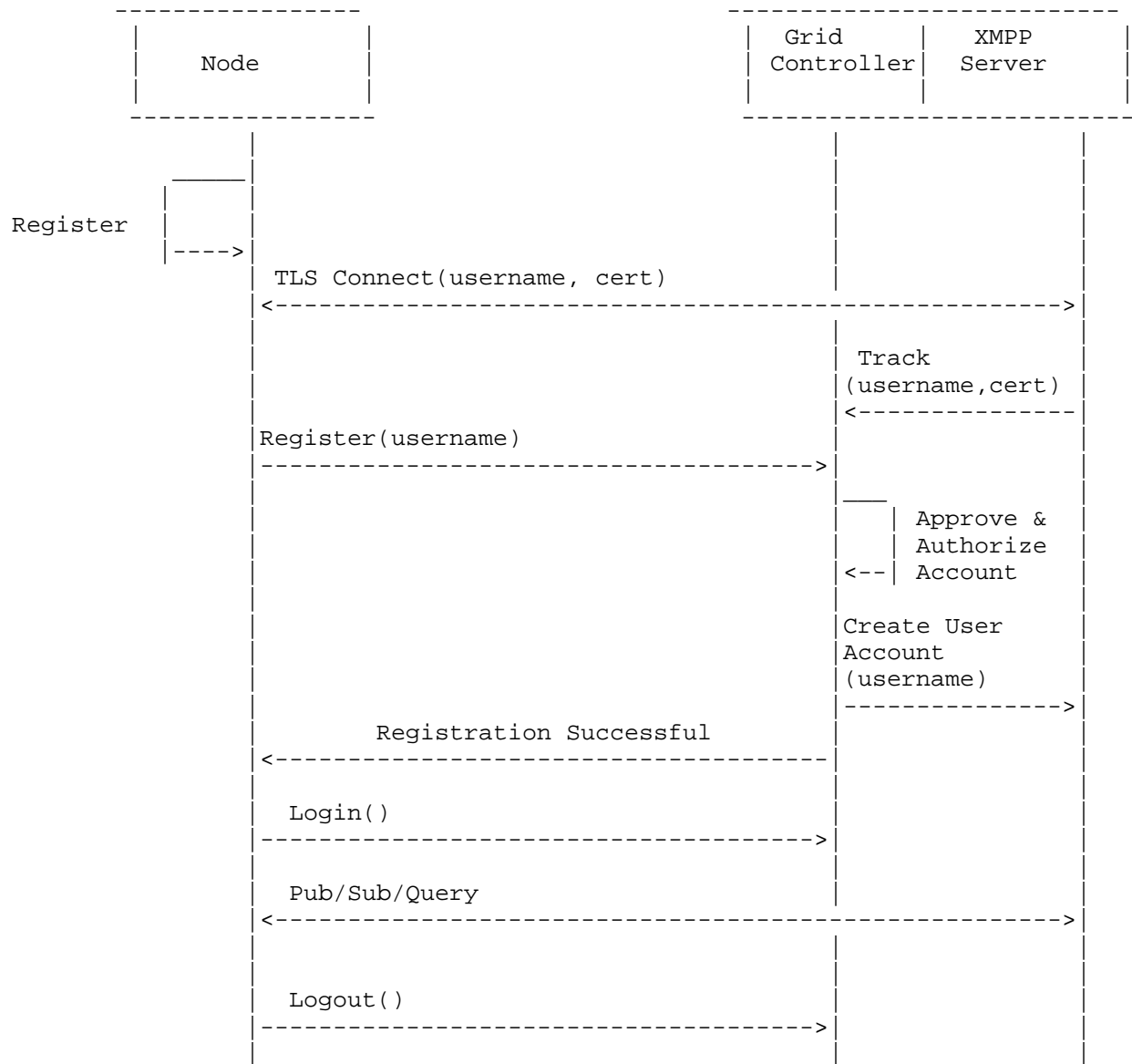


Figure 4: XMPP-Grid Node Registration

2.4.3. Authorization

The registered Nodes send subscription requests to the Controller. The Controller, depending on the defined authorization privileges, grants permissions to subscribe and/or publish to a Topic at the registration time. The Controller updates the XMPP data plane server with the new subscriber information and its capability. Node identity extracted from the request, group to which the Node is assigned during account approval and Topic/capability to which the permission is sought could be some of the ways to authorize Nodes and their requests in XMPP-Grid. Similarly, the Controller authorizes directed peer-to-peer or out-of-band requests from a requesting peer. The destination peer has options to query back the Controller to retrieve and enforce granular authorizations such as read-only, write-only, read/write.

In a Query Authorization flow, the capability provider responding to the query is responsible for enforcing the authorization decision. It retrieves "is authorized" from the XMPP-Grid Controller. Based on the result, the service either allows or disallows the flow from continuing.

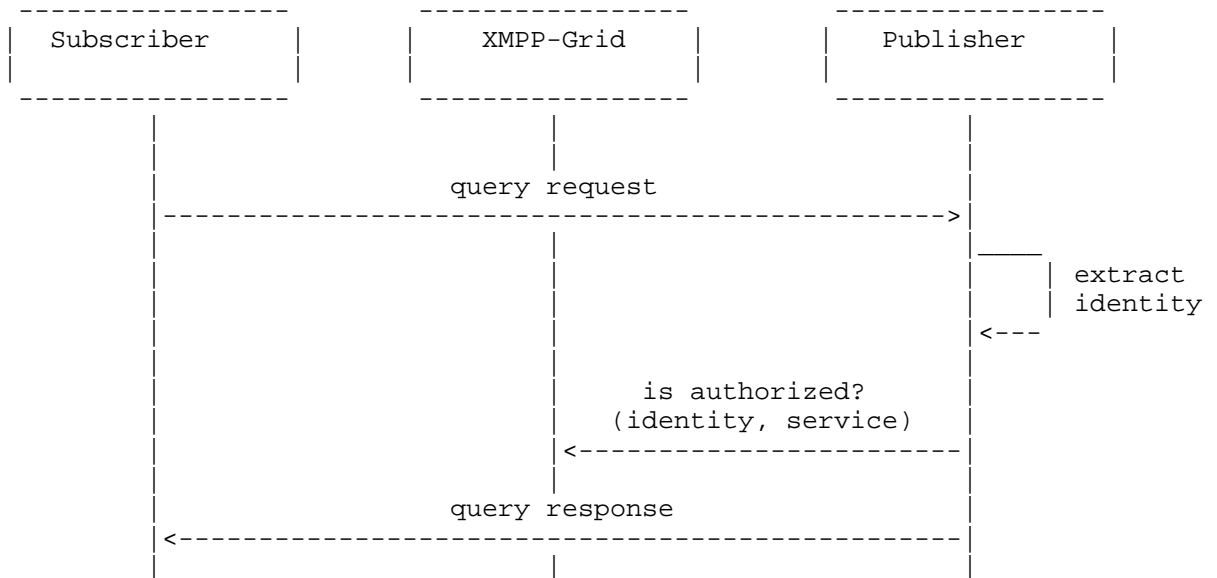


Figure 5: Node Query Authorization Flow

For Publish Authorization, prior to allowing a publish request by a user, the XMPP-Grid Controller calls the rule evaluation engine directly for "is authorized". Based this result, the Controller either allows or disallowed the flow from continuing.

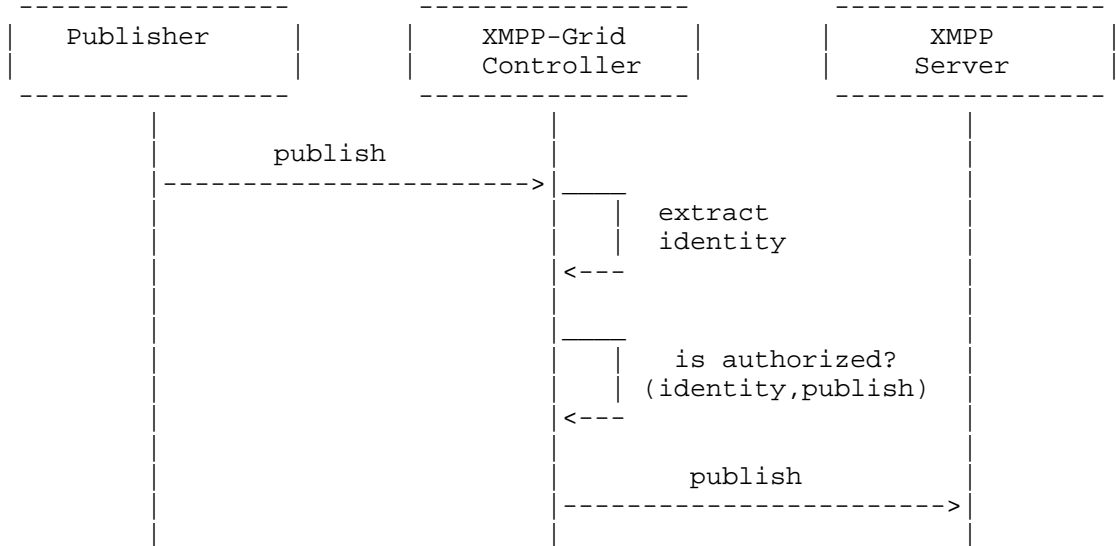


Figure 6: Node Publish Authorization Flow

For Subscribe Authorization, prior to allowing a subscribe request by a user, the XMPP-Grid Controller calls the rule evaluation engine directly for "is authorized". Based this result, the Controller either allows or disallowed the flow from continuing.

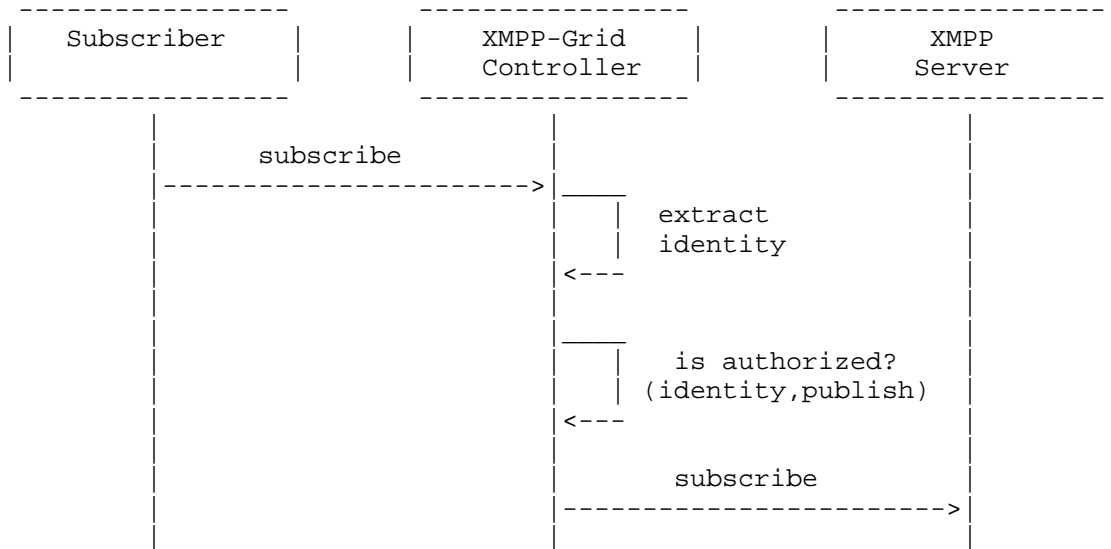


Figure 7: Node Subscribe Authorization Flow

Bulk Data Query differs from other data transfer modes. Unlike with other modes of communication that operate in-band with the XMPP-Grid, bulk downloads occur out-of-band (over a different protocol, outside of the connection that was established with the XMPP-Grid Controller). Previously discussed authorization mechanisms are therefore not appropriate in this context.

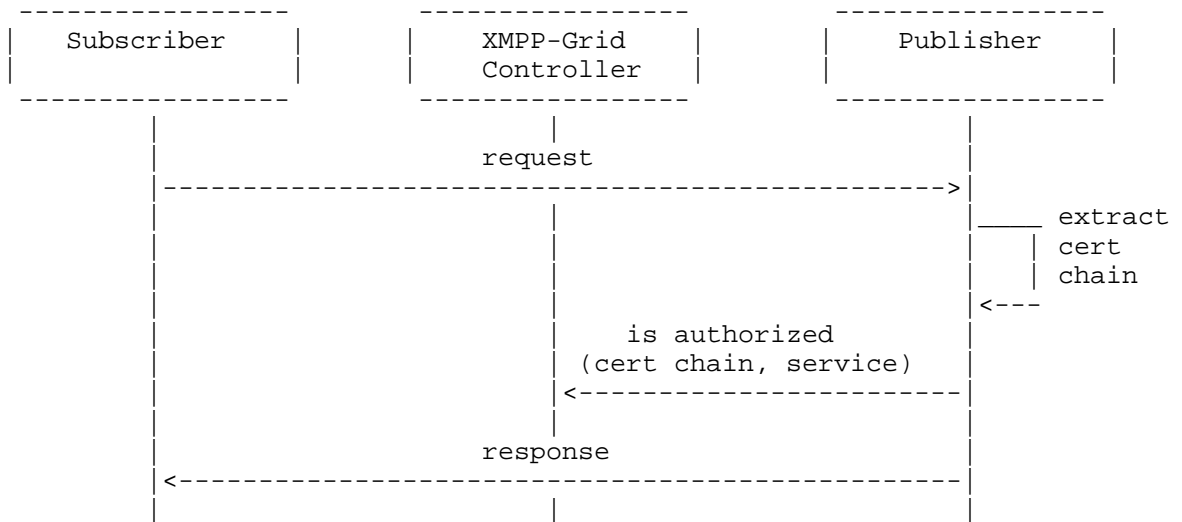


Figure 8: Node Bulk Data Query Flow

Instead the bulk download service sends the certificate chain used by a Node in the TLS connection to the XMPP-Grid Controller for purposes of authenticating and authorizing the Node. Upon receiving a request with a certificate chain, the Controller checks the issuing certificate against the trust store, looks up the identity associated with the certificate, evaluates the rules, and returns "is authorized" to the service. Then the service can either allow or disallow the flow from continuing.

2.5. XMPP-Grid Topics Protocol Flow

For each capability, XMPP-Grid supports extensibility through XML schemas where the providers (publishers) of the capabilities define the schemas for the data exchanged. The capability provider shall also define the version, the available queries and notifications that it can support. The capability provider publishes the messages to one or more XMPP Topics, that it requests XMPP-Grid to create dynamically, depending on:

- a. If the capability provider has mutually exclusive schemas, different Topics will be created where the capability provider will be a publisher to each Topic with a separate schema.
- b. For a given Topic, if the subscribers wants to receive filtered attributes or attribute values in capability provider's published data, XMPP-Grid Controller creates sub Topics to the main Topic

based on the message filters expressed. XMPP-Grid Controller enrolls the capability provider as the publisher and the requesting subscribers based on the message filter criteria they express. The capability provider will be the publisher to both the main Topic and the sub-Topics.

- c. In the case mentioned in (b) above, it is possible for the capability provider to just publish on the main Topic and have the XMPP-Grid Controller filter the published messages on the Controller-side and deliver attributes and attribute values of interest to the subscribers. Controller-side message filter application and the specify mechanisms such as XPATH that can be used for parsing the messages is beyond the scope of this specification.

2.5.1. Topic Versioning

XMPP-Grid supports versioning to support forward and backward compatible information models. The providers of capability include the version number in the messages they publish and the receiving Nodes can interpret the Topic version and process the attributes accordingly. The expectation is any new version of a capability must be of additive updates only. In other words, existing elements and attributes cannot be changed, only new elements or attributes can be added. This will enable nodes with older capability be able to process newer version. The extra new elements or attributes will be ignored. Instead of using the same Topic for all versions, it is possible in XMPP-Grid to programmatically create separate Topics for each version and allow them to be discovered and subscribed by the Nodes.

In XMPP-Grid, versioning support applies equally to both publish/subscribe, directed and out-of-band queries.

2.5.2. Topic Discovery

The Nodes connected to XMPP-Grid can query the Controller and get the list of all capabilities/Topics running on XMPP-Grid. The XML samples provided in XMPP-Grid Protocol section above provide illustrations of Capability Query and Capability Provider Query.

2.5.3. Subtopics and Message Filters

XMPP-Grid supports semantic message filtering for Topics. The content being published by a provider can be semantically grouped into categories based on domain, location of endpoints for example. The provider of a capability specifies whether it supports semantic

filtering or not to the Controller at the subscribe time to the Topic under consideration.

XMPP-Grid subscribers query the Controller and obtain the filtering options available for each capability, and express their message filtering criteria at subscription time. The Controller, for each unique filter criteria specified by the subscribers, creates a new sub Topic under the main capability Topic. All the subscribers with the same filtering criteria will be subscribed to the Subtopic. The set of filter criteria for a capability will be predefined by the capability provider and could be based on the well-defined attributes of the message.

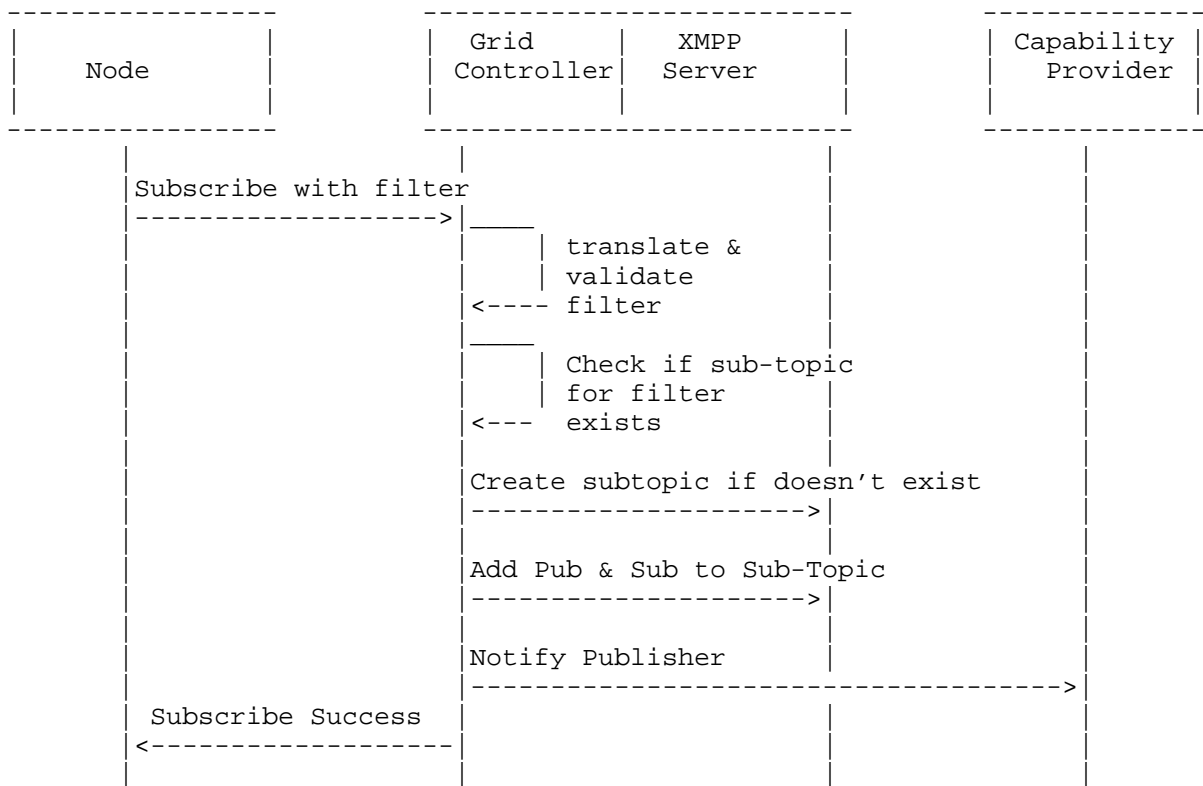


Figure 9: Subtopics and Information Filter Subscribe Operations Flow

The publisher will be responsible for applying the filter on the message and publishing the message on the Topic and the Subtopic based on the filter criteria. Filtering logic will be on the

publisher, as the publisher understands the message content. XMPP-Grid fabric is oblivious to the message content.

To avoid proliferation of new Subtopics, the capability provider could express the configurable limit on the number of Subtopics that can be created for its capability at registration time. The XMPP-Grid Controller will perform periodic cleanup of Subtopics whenever their subscription list reduces to 0.

In XMPP-Grid, message filters are provided to all APIs i.e. publish/subscribe and directed query.

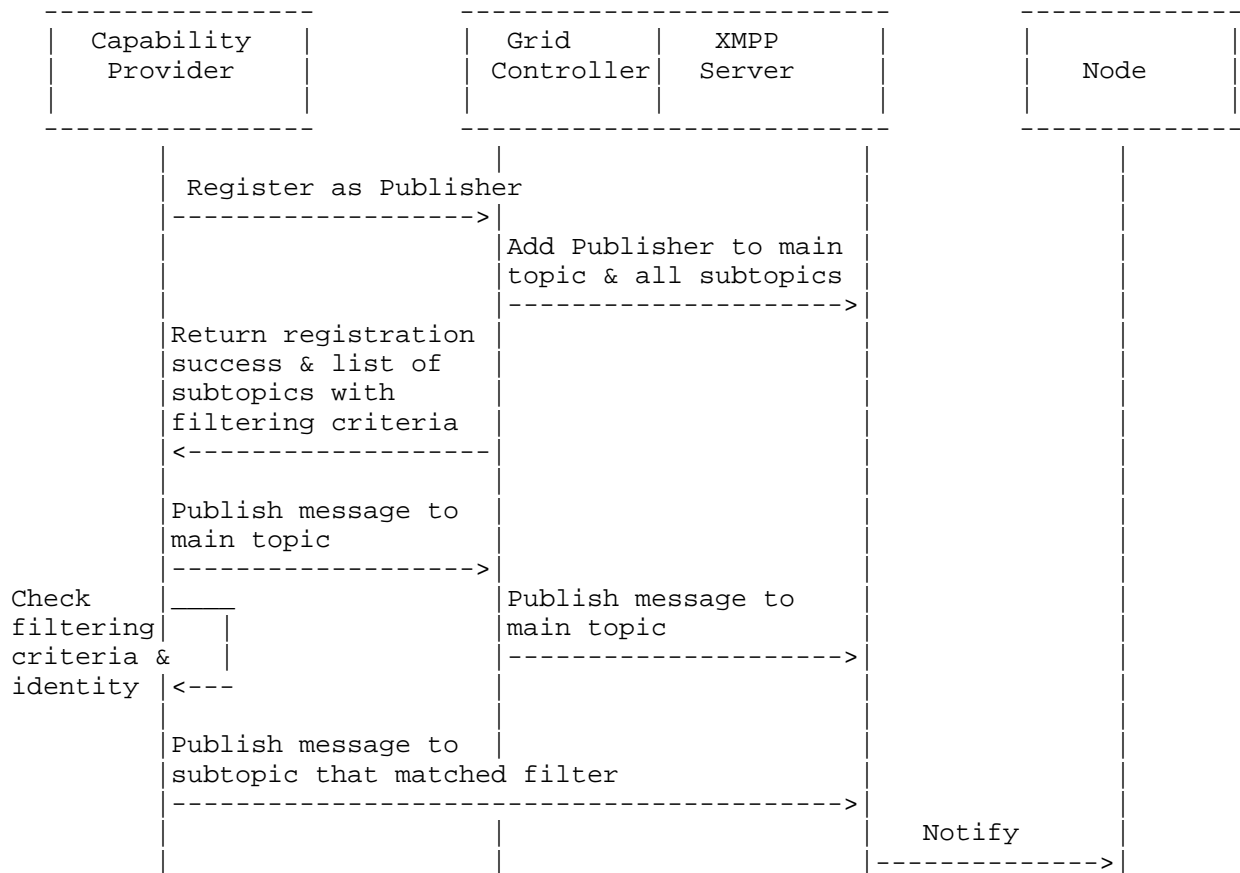


Figure 10: Subtopic Publish Operations Flow

2.6. XMPP-Grid Protocol Details

The XMPP-Grid Protocol provides an abstraction layer over and above XMPP messages with the intent to provide intuitive interfaces to the Nodes connecting to XMPP-Grid. Nodes connecting to XMPP-Grid use the following interfaces (provided as XML samples) offered by XMPP-Grid protocol to connect and participate in information exchange on XMPP-Grid:

o Register the Node to XMPP-Grid: Node identified as "Node2@domain.com/mac" sends the following Registration request to XMPP-Grid controller.

```
<iq id="ay0tK-4" to="grid_Controller.jabber"
  from="Node2@domain.com/syam-mac" type="get">
  <grid xmlns='gi' type='request'>
    <AccountQuery xmlns='com.domain.gi.gcl.Controller'>
      <register></register></AccountQuery>
    </grid>
  </iq>
```

o Node login to XMPP-Grid: The following XML sample shows the Login request from Node "Node2@domain.com/mac" to XMPP-Grid controller and Login response returned by the XMPP-Grid controller to the Node.

```
// Request
<iq id="ay0tK-5" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns='gi' type='request'>
    <AccountQuery xmlns='com.domain.gi.gcl.Controller'>
      <login></login>
    </AccountQuery>
  </grid>
</iq>

// Response
<iq xmlns="jabber:client" to=" Node2@domain.com/mac"
  from="grid_Controller.jabber" type="result" id="ay0tK-5">
  <grid xmlns="gi" type="response">
    <AccountQuery xmlns="com.domain.gi.gcl.Controller">
      <login xmlns="">
        <value xmlns:ns2="gi" xmlns:xsi=" xsi:nil="true" />
      </login>
    </AccountQuery>
  </grid>
</iq>
```

o Node logout from XMPP-Grid: The following XML sample shows the Logout request sent by Node "Node2@domain.com/mac" to XMPP-Grid controller.

```
<iq id="o47m2-8" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns='gi' type='request'>
    <AccountQuery xmlns='com.domain.gi.gcl.Controller'>
      <logout></logout>
    </AccountQuery>
  </grid>
</iq>
```

o Capability Discovery Query: The following XML sample shows the Capability Discovery query request from Node "Node2@domain.com/mac" to XMPP-Grid controller. The XMPP-Grid controller returns the list of capabilities supported by XMPP-Grid and their versions as a response to the Node's request.

// Request

```
<iq id="tVKqm-6" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns="xgrid" type="request">
    <ns2:getCapabilityListRequest xmlns:ns2="  xmlns:ns4="
      xmlns:ns3=" xmlns:ns5=" xmlns:ns6=" xmlns:ns7=" />
  </grid>
</iq>
```

// Response

```
<iq from="grid_Controller.jabber" id="tVKqm-6"
  to="Node2@domain.com/mac" type="result" xmlns="jabber:client">
  <grid type="response" xmlns="xgrid">
    <ns2:getCapabilityListResponse xmlns:ns2=" xmlns:ns3="
      xmlns:ns4=" xmlns:ns5=" xmlns:ns6=" xmlns:ns7=">
    <ns2:capability xmlns:xsi="
      " xsi:type="ns5:TrustSecMetaDataCapability">
      <ns2:id>0</ns2:id>
      <ns2:name>TrustSecMetaDataCapability-1.0</ns2:name>
      <ns2:version>1.0</ns2:version>
    </ns2:capability>
    <ns2:capability
      xmlns:xsi=" xsi:type="ns5:EndpointProfileMetaDataCapability">
      <ns2:id>0</ns2:id>
      <ns2:name>
        EndpointProfileMetaDataCapability-1.0</ns2:name>
      <ns2:version>1.0</ns2:version>
    </ns2:capability>
```

```

<ns2:capability xmlns:xsi=
    " xsi:type="ns5:IdentityGroupCapability">
    <ns2:id>0</ns2:id>
    <ns2:name>IdentityGroupCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
<ns2:capability xmlns:ns9=" xmlns:xsi="
    xsi:type="ns9:TDAAnalysisServiceCapability">
    <ns2:id>0</ns2:id>
    <ns2:name>TDAAnalysisServiceCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
<ns2:capability xmlns:xsi=" xsi:type="
    ns7:NetworkCaptureCapability">
    <ns2:id>0</ns2:id>
    <ns2:name>NetworkCaptureCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
<ns2:capability xmlns:xsi=
    " xsi:type="ns6:EndpointProtectionServiceCapability"
>
    <ns2:id>0</ns2:id>
    <ns2:name>
        EndpointProtectionServiceCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
<ns2:capability xmlns:xsi=
    " xsi:type="ns4:GridControllerAdminServiceCapability">
    <ns2:id>0</ns2:id>
    <ns2:name>
        GridControllerAdminServiceCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
<ns2:capability xmlns:xsi=
    " xsi:type="ns5:SessionDirectoryCapability">
    <ns2:id>0</ns2:id>
    <ns2:name>SessionDirectoryCapability-1.0</ns2:name>
    <ns2:version>1.0</ns2:version>
</ns2:capability>
</ns2:getCapabilityListResponse>
</grid>
</iq>

```

o Specific Capability Provider Query: The following XML sample shows the Capability Provider hostname query request from Node "Node2@domain.com/mac" to XMPP-Grid controller. XMPP-Grid controller returns the hostname of the specific Capability Provider as a response to the Node's request.

```
// Request
<iq id="996IL-8" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns='gi' type='request'>
    <DiscoveryQuery xmlns='com.domain.gi.gcl.Controller'>
      <find><param xsi:type="xs:string" xmlns:ns2="gi" xmlns:xs
        =" xmlns:xsi=">com.domain.ise.session.SessionQuery
      </param></find>
    </DiscoveryQuery>
  </grid>
</iq>

// Response
<iq from='grid_Controller.jabber' id='996IL-8'
  to='Node2@domain.com/mac' type='result'
  xmlns='jabber:client'>
  <grid type='response' xmlns='gi'>
    <DiscoveryQuery xmlns='com.domain.gi.gcl.Controller'>
      <find xmlns=''><value xmlns:ns3='http://jaxb.dev.java.net/array'
        xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
        xsi:type='ns3:stringArray'>
        <item>ise@syam-06.domain.com/syam-mac</item></value></find>
    </DiscoveryQuery>
  </grid>
</iq>
```

o Register as a publisher to the Topic: The following XML sample shows the Register as a Publisher request from a Node "Node2@domain.com/mac" to XMPP-Grid controller.

```
<iq id="fD65a-6" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns="xgrid" type="request">
    <ns2:initPublishRequest xmlns:ns2=" " xmlns:ns4="
      " xmlns:ns3=" " xmlns:ns5=" " xmlns:ns6=" " xmlns:ns7=" ">
      <ns2:capability xsi:type="ns5:SessionCapability"
        xmlns:xsi=" ">
        <ns2:id>0</ns2:id>
        <ns2:version>1.0</ns2:version>
      </ns2:capability>
    </ns2:initPublishRequest>
  </grid>
</iq>
```

o Register as a subscriber to the Topic: The following XML sample shows a subscription request made by Node "Node2@domain.com/mac" for "SessionCapability" Topic to XMPP-Grid controller. On success, determined by the Node's authorization privilege, XMPP-Grid controller returns the Topic name, version and the Publishers' hostname as a response to the Node's request.

```
// Subscribe Request
<iq id="lQJIT-6" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns="xgrid" type="request">
    <ns2:subscribeRequest xmlns:ns2=" " xmlns:ns4=" " xmlns:ns3
      =" " xmlns:ns5=" " xmlns:ns6=" " xmlns:ns7=" ">
      <ns2:capability xsi:type="ns5:SessionCapability"
        xmlns:xsi=" ">
        <ns2:id>0</ns2:id>
        <ns2:version>1.0</ns2:version>
      </ns2:capability>
    </ns2:subscribeRequest>
  </grid>
</iq>

// Subscribe Response
<iq from="grid_Controller.jabber" id=" lQJIT-6"
  to="Node2@domain.com/mac" type="result" xmlns="jabber:client">
  <grid type="response" xmlns="xgrid">
    <ns2:subscribeResponse xmlns:ns2="
      " xmlns:ns3=" " xmlns:ns4=" " xmlns:ns5="
        xmlns:ns6=" " xmlns:ns7=" ">
    <ns2:topicName>SessionCapability-1.0</ns2:topicName>
    <ns2:xmppDetails>
      <ns2:jid>ise-mnt-XMPP-Grid-004@xgrid.domain.com/gcl
    </ns2:jid>
      <ns2:jid>ise-mnt-XMPP-Grid-005@xgrid.domain.com/gcl
    </ns2:jid>
    </ns2:xmppDetails>
  </ns2:subscribeResponse>
</grid>
</iq>
```

o Peer-to-Peer Directed Query: The following XML sample shows a peer-to-peer directed query request made by Node "Node2@domain.com/mac" to other XMPP-Grid participating Node "grid_Controller.jabber", seeking identity group information for a specific user "user1". "grid_Controller.jabber" returns the list of identity groups "user1" belongs as a response to the request.

```
// Query Request
<iq id="kR0YY-8" to="grid_Controller.jabber"
  from="Node2@domain.com/mac" type="get">
  <grid xmlns="xgrid" type="request">
    <ns5:getIdentityGroupRequest xmlns:ns2=" xmlns:ns4="
      xmlns:ns3=" xmlns:ns5=" xmlns:ns6=" xmlns:ns7=">
      <ns5:user>
        <ns2:name>user1</ns2:name>
      </ns5:user>
    </ns5:getIdentityGroupRequest>
  </grid>
</iq>

// Query Response
<iq from="grid_Controller.jabber"
  id=" kR0YY-8" to="Node2@domain.com/mac" type="result">
  <grid type="response" xmlns="xgrid">
    <ns5:getIdentityGroupResponse xmlns:ns2=" xmlns:ns3="
      " xmlns:ns4=" xmlns:ns5=" xmlns:ns6=" xmlns:ns7=">
    <ns5:user>
      <ns2:name>user1</ns2:name>
      <ns3:groupList>
        <ns3:object>
          <ns2:name>User Identity Groups:Employee
          </ns2:name>
          <ns3:type>Identity</ns3:type>
        </ns3:object>
      </ns3:groupList>
    </ns5:user>
  </ns5:getIdentityGroupResponse>
</grid>
</iq>
```

3. XMPP-Grid Compatibility with IODEF

The Incident Object Description and Exchange Format (IODEF) [RFC5070] defines a common data format and common exchange procedures for sharing incidents and related data between CSIRTs. RFC5070 provides the information and data model for IODEF specified with XML schema.

XEP-0268 (<http://xmpp.org/extensions/xep-0268.html>), Incident Handling, defines ways for XMPP server deployments to share incident reports with each other using the IODEF format and handle attacks on the servers in real-time.

Providers of incident reports, across administrative domains, could participate as publishers to an XMPP topic (for example: IODEF). Trust is achieved through authentication, authorization and account approval as defined in Section 2.4. The providers could expose IODEF incident attributes such as Authority as message filter criteria for the topic in order for subscribing systems to subscribe to incident reports from administrative domains of interest. The providers could further expose other IODEF attributes such as Assessment, Method, Attacker etc as message filter criteria for subscribers to selectively choose events of interest that are published from administrative domain(s). Privacy and regulatory requirements of information shared across administrative domains is beyond the scope of this document.

4. IANA Considerations

IANA Considerations to be determined

5. Security Considerations

A XMPP-Grid Controller serves as an controlling broker for XMPP-Grid Nodes such as Enforcement Points, Policy Servers, CMDBs, and Sensors, using a publish-subscribe-search model of information exchange and lookup. By increasing the ability of XMPP-Grid Nodes to learn about and respond to security-relevant events and data, XMPP-Grid can improve the timeliness and utility of the security system. However, this integrated security system can also be exploited by attackers if they can compromise it. Therefore, strong security protections for XMPP-Grid are essential.

This section provides a security analysis of the XMPP-Grid transport protocol and the architectural elements that employ it, specifically with respect to their use of this protocol. Three subsections define the trust model (which elements are trusted to do what), the threat model (attacks that may be mounted on the system), and the countermeasures (ways to address or mitigate the threats previously identified).

5.1. Trust Model

The first step in analyzing the security of the XMPP-Grid transport protocol is to describe the trust model, listing what each architectural element is trusted to do. The items listed here are

assumptions, but provisions are made in the Threat Model and Countermeasures sections for elements that fail to perform as they were trusted to do.

5.1.1. Network

The network used to carry XMPP-Grid messages is trusted to:

- o Perform best effort delivery of network traffic

The network used to carry XMPP-Grid messages is not expected (trusted) to:

- o Provide confidentiality or integrity protection for messages sent over it
- o Provide timely or reliable service

5.1.2. XMPP-Grid Nodes

Authorized XMPP-Grid Nodes are trusted to:

- o Preserve the confidentiality of sensitive data retrieved via the XMPP-Grid Controller

5.1.3. XMPP-Grid Controller

The XMPP-Grid Controller is trusted to:

- o Broker requests for data and enforce authorization of access to this data throughout its lifecycle
- o Perform service requests in a timely and accurate manner
- o Create and maintain accurate operational attributes
- o Only reveal data to and accept service requests from authorized parties

The XMPP-Grid Controller is not expected (trusted) to:

- o Verify the truth (correctness) of data

5.1.4. Certification Authority

The Certification Authority (CA) that issues certificates for the XMPP-Grid Controller and/or XMPP-Grid Nodes (or each CA, if there are several) is trusted to:

- o Protect the confidentiality of the CA's private key
- o Ensure that only proper certificates are issued and that all certificates are issued in accordance with the CA's policies
- o Revoke certificates previously issued when necessary
- o Regularly and securely distribute certificate revocation information
- o Promptly detect and report any violations of this trust so that they can be handled

The CA is not expected (trusted) to:

- o Issue certificates that go beyond name constraints or other constraints imposed by a relying party or a cross-certificate

5.2. Threat Model

To secure the XMPP-Grid transport protocol and the architectural elements that implement it, this section identifies the attacks that can be mounted against the protocol and elements.

5.2.1. Network Attacks

A variety of attacks can be mounted using the network. For the purposes of this subsection the phrase "network traffic" should be taken to mean messages and/or parts of messages. Any of these attacks may be mounted by network elements, by parties who control network elements, and (in many cases) by parties who control network-attached devices.

- o Network traffic may be passively monitored to glean information from any unencrypted traffic
- o Even if all traffic is encrypted, valuable information can be gained by traffic analysis (volume, timing, source and destination addresses, etc.)
- o Network traffic may be modified in transit
- o Previously transmitted network traffic may be replayed
- o New network traffic may be added
- o Network traffic may be blocked, perhaps selectively

- o A "Man In The Middle" (MITM) attack may be mounted where an attacker interposes itself between two communicating parties and poses as the other end to either party or impersonates the other end to either or both parties
- o Resist attacks (including denial of service and other attacks from XMPP-Grid Nodes)
- o Undesired network traffic may be sent in an effort to overload an architectural component, thus mounting a denial of service attack

5.2.2. XMPP-Grid Nodes

An unauthorized XMPP-Grid Nodes (one which is not recognized by the XMPP-Grid Controller or is recognized but not authorized to perform any actions) cannot mount any attacks other than those listed in the Network Attacks section above.

An authorized XMPP-Grid Node, on the other hand, can mount many attacks. These attacks might occur because the XMPP-Grid Node is controlled by a malicious, careless, or incompetent party (whether because its owner is malicious, careless, or incompetent or because the XMPP-Grid Node has been compromised and is now controlled by a party other than its owner). They might also occur because the XMPP-Grid Node is running malicious software; because the XMPP-Grid Node is running buggy software (which may fail in a state that floods the network with traffic); or because the XMPP-Grid Node has been configured improperly. From a security standpoint, it generally makes no difference why an attack is initiated. The same countermeasures can be employed in any case.

Here is a list of attacks that may be mounted by an authorized XMPP-Grid Node:

- o Cause many false alarms or otherwise overload the XMPP-Grid Controller or other elements in the network security system (including human administrators) leading to a denial of service or disabling parts of the network security system
- o Omit important actions (such as posting incriminating data), resulting in incorrect access
- o Use confidential information obtained from the XMPP-Grid Controller to enable further attacks (such as using endpoint health check results to exploit vulnerable endpoints)

- o Advertise data crafted to exploit vulnerabilities in the XMPP-Grid Controller or in other XMPP-Grid Nodes, with a goal of compromising those systems
- o Issue a search request or set up a subscription that matches an enormous result, leading to resource exhaustion on the XMPP-Grid Controller, the publishing XMPP-Grid Node, and/or the network
- o Establish a communication channel using another XMPP-Grid Node's session-id

Dependencies of or vulnerabilities of authorized XMPP-Grid Nodes may be exploited to effect these attacks. Another way to effect these attacks is to gain the ability to impersonate a XMPP-Grid Node (through theft of the XMPP-Grid Node's identity credentials or through other means). Even a clock skew between the XMPP-Grid Node and XMPP-Grid Controller can cause problems if the XMPP-Grid Node assumes that old XMPP-Grid Node data should be ignored.

5.2.3. XMPP-Grid Controllers

An unauthorized XMPP-Grid Controller (one which is not trusted by XMPP-Grid Nodes) cannot mount any attacks other than those listed in the Network Attacks section above.

An authorized XMPP-Grid Controller can mount many attacks. Similar to the XMPP-Grid Node case described above, these attacks might occur because the XMPP-Grid Controller is controlled by a malicious, careless, or incompetent party (either a XMPP-Grid Controller administrator or an attacker who has seized control of the XMPP-Grid Controller). They might also occur because the XMPP-Grid Controller is running malicious software, because the XMPP-Grid Controller is running buggy software (which may fail in a state that corrupts data or floods the network with traffic), or because the XMPP-Grid Controller has been configured improperly.

All of the attacks listed for XMPP-Grid Node above can be mounted by the XMPP-Grid Controller. Detection of these attacks will be more difficult since the XMPP-Grid Controller can create false operational attributes and/or logs that imply some other party created any bad data.

Additional XMPP-Grid Controller attacks may include:

- o Expose different data to different XMPP-Grid Nodes to mislead investigators or cause inconsistent behavior

- o Mount an even more effective denial of service attack than a single XMPP-Grid Node could
- o Obtain and cache XMPP-Grid Node credentials so they can be used to impersonate XMPP-Grid Nodes even after a breach of the XMPP-Grid Controller is repaired
- o Obtain and cache XMPP-Grid Controller administrator credentials so they can be used to regain control of the XMPP-Grid Controller after the breach of the XMPP-Grid Controller is repaired

Dependencies of or vulnerabilities of the XMPP-Grid Controller may be exploited to obtain control of the XMPP-Grid Controller and effect these attacks.

5.2.4. Certification Authority

A Certification Authority trusted to issue certificates for the XMPP-Grid Controller and/or XMPP-Grid Nodes can mount several attacks:

- o Issue certificates for unauthorized parties, enabling them to impersonate authorized parties such as the XMPP-Grid Controller or a XMPP-Grid Node. This can lead to all the threats that can be mounted by the certificate's subject.
- o Issue certificates without following all of the CA's policies. Because this can result in issuing certificates that may be used to impersonate authorized parties, this can lead to all the threats that can be mounted by the certificate's subject.
- o Fail to revoke previously issued certificates that need to be revoked. This can lead to undetected impersonation of the certificate's subject or failure to revoke authorization of the subject, and therefore can lead to all of the threats that can be mounted by that subject.
- o Fail to regularly and securely distribute certificate revocation information. This may cause a relying party to accept a revoked certificate, leading to undetected impersonation of the certificate's subject or failure to revoke authorization of the subject, and therefore can lead to all of the threats that can be mounted by that subject. It can also cause a relying party to refuse to proceed with a transaction because timely revocation information is not available, even though the transaction should be permitted to proceed.

- o Allow the CA's private key to be revealed to an unauthorized party. This can lead to all the threats above. Even worse, the actions taken with the private key will not be known to the CA.
- o Fail to promptly detect and report errors and violations of trust so that relying parties can be promptly notified. This can cause the threats listed earlier in this section to persist longer than necessary, leading to many knock-on effects.

5.3. Countermeasures

Below are countermeasures for specific attack scenarios to the XMPP-Grid infrastructure.

5.3.1. Securing the XMPP-Grid Transport Protocol

To address network attacks, the XMPP-Grid transport protocol described in this document requires that the XMPP-Grid messages **MUST** be carried over TLS (minimally TLS 1.2 [RFC5246]) as described in [RFC2818]. The XMPP-Grid Node **MUST** verify the XMPP-Grid Controller's certificate and determine whether the XMPP-Grid Controller is trusted by this XMPP-Grid Node before completing the TLS handshake. The XMPP-Grid Controller **MUST** authenticate the XMPP-Grid Node either using mutual certificate-based authentication in the TLS handshake or using Basic Authentication as described in IETF RFC 2617. XMPP-Grid Controller **MUST** use Simple Authentication and Security Layer (SASL), described in [RFC4422], to support the aforesaid authentication mechanisms. SASL offers authentication mechanism negotiations between the XMPP-Grid Controller and XMPP-Grid node during the connection establishment phase. XMPP-Grid Nodes and XMPP-Grid Controllers using mutual certificate-based authentication **SHOULD** each verify the revocation status of the other party. All XMPP-Grid Controllers and XMPP-Grid Nodes **MUST** implement both mutual certificate-based authentication and Basic Authentication. The selection of which XMPP-Grid Node authentication technique to use in any particular deployment is left to the administrator.

An XMPP-Grid Controller **MAY** also support a local, configurable set of Basic Authentication userid-password pairs. If so, it is implementation dependent whether a XMPP-Grid Controller ends a session when an administrator changes the configured password. Since Basic Authentication has many security disadvantages (especially the transmission of reusable XMPP-Grid Node passwords to the XMPP-Grid Controller), it **SHOULD** only be used when absolutely necessary. Per the HTTP specification, when basic authentication is in use, a XMPP-Grid Controller **MAY** respond to any request that lacks credentials with an error code similar to HTTP code 401. A XMPP-Grid Node **SHOULD** avoid this code by submitting basic auth credentials with every

request when basic authentication is in use. If it does not do so, a XMPP-Grid Node MUST respond to this code by resubmitting the same request with credentials (unless the XMPP-Grid Node is shutting down).

As XMPP uses TLS as the transport and security mechanisms, it is understood that best practices such as those in [I-D.ietf-uta-tls-bcp] are followed.

These protocol security measures provide protection against all the network attacks listed in the above document section except denial of service attacks. If protection against these denial of service attacks is desired, ingress filtering, rate limiting per source IP address, and other denial of service mitigation measures may be employed. In addition, a XMPP-Grid Controller MAY automatically disable a misbehaving XMPP-Grid Node.

5.3.2. Securing XMPP-Grid Nodes

XMPP-Grid Nodes may be deployed in locations that are susceptible to physical attacks. Physical security measures may be taken to avoid compromise of XMPP-Grid Nodes, but these may not always be practical or completely effective. An alternative measure is to configure the XMPP-Grid Controller to provide read-only access for such systems. The XMPP-Grid Controller SHOULD also include a full authorization model so that individual XMPP-Grid Nodes may be configured to have only the privileges that they need. The XMPP-Grid Controller MAY provide functional templates so that the administrator can configure a specific XMPP-Grid Node as a DHCP server and authorize only the operations and metadata types needed by a DHCP server to be permitted for that XMPP-Grid Node. These techniques can reduce the negative impacts of a compromised XMPP-Grid Node without diminishing the utility of the overall system.

To handle attacks within the bounds of this authorization model, the XMPP-Grid Controller MAY also include rate limits and alerts for unusual XMPP-Grid Node behavior. XMPP-Grid Controllers SHOULD make it easy to revoke a XMPP-Grid Node's authorization when necessary. Another way to detect attacks from XMPP-Grid Nodes is to create fake entries in the available data (honeytokens) which normal XMPP-Grid Nodes will not attempt to access. The XMPP-Grid Controller SHOULD include auditable logs of XMPP-Grid Node activities.

To avoid compromise of XMPP-Grid Node, XMPP-Grid Node SHOULD be hardened against attack and minimized to reduce their attack surface. They SHOULD go through a TNC handshake to verify the integrity of the XMPP-Grid Node, and SHOULD, if feasible, utilize a Trusted Platform Module (TPM) for identity and/or integrity measurements of the XMPP-

Grid Node within a TNC handshake. They should be well managed to minimize vulnerabilities in the underlying platform and in systems upon which the XMPP-Grid Node depends. Personnel with administrative access should be carefully screened and monitored to detect problems as soon as possible.

5.3.3. Securing XMPP-Grid Controllers

Because of the serious consequences of XMPP-Grid Controller compromise, XMPP-Grid Controllers SHOULD be especially well hardened against attack and minimized to reduce their attack surface. They SHOULD go through a regular TNC handshake to verify the integrity of the XMPP-Grid Controller, and SHOULD utilize a Trusted Platform Module (TPM) for identity and/or integrity measurements of the XMPP-Grid Node within a TNC handshake. They should be well managed to minimize vulnerabilities in the underlying platform and in systems upon which the XMPP-Grid Controller depends. Network security measures such as firewalls or intrusion detection systems may be used to monitor and limit traffic to and from the XMPP-Grid Controller. Personnel with administrative access should be carefully screened and monitored to detect problems as soon as possible. Administrators should not use password-based authentication but should instead use non-reusable credentials and multi-factor authentication (where available). Physical security measures SHOULD be employed to prevent physical attacks on XMPP-Grid Controllers.

To ease detection of XMPP-Grid Controller compromise should it occur, XMPP-Grid Controller behavior should be monitored to detect unusual behavior (such as a reboot, a large increase in traffic, or different views of an information repository for similar XMPP-Grid Nodes). XMPP-Grid Nodes should log and/or notify administrators when peculiar XMPP-Grid Controller behavior is detected. To aid forensic investigation, permanent read-only audit logs of security-relevant information (especially administrative actions) should be maintained. If XMPP-Grid Controller compromise is detected, a careful analysis should be performed of the impact of this compromise. Any reusable credentials that may have been compromised should be reissued.

5.3.4. Limit on search result size

While XMPP-Grid is designed for high scalability to 100,000s of Nodes, an XMPP-Grid Controller MAY establish a limit to the amount of data it is willing to return in search or subscription results. This mitigates the threat of a XMPP-Grid Node causing resource exhaustion by issuing a search or subscription that leads to an enormous result.

5.3.5. Cryptographically random session-id and authentication checks for ARC

A XMPP-Grid Controller SHOULD ensure that the XMPP-Grid Node establishing an ARC is the same XMPP-Grid Node as the XMPP-Grid Node that established the corresponding SSRC. The XMPP-Grid Controller SHOULD employ both of the following strategies:

- o session-ids SHOULD be cryptographically random
- o The HTTPS transport for the SSRC and the ARC SHOULD be authenticated using the same credentials. SSL session resumption MAY be used to establish the ARC based on the SSRC SSL session.

5.3.6. Securing the Certification Authority

As noted above, compromise of a Certification Authority (CA) trusted to issue certificates for the XMPP-Grid Controller and/or XMPP-Grid Nodes is a major security breach. Many guidelines for proper CA security have been developed: the CA/Browser Forum's Baseline Requirements, the AICPA/CICA Trust Service Principles, etc. The CA operator and relying parties should agree on an appropriately rigorous security practices to be used.

Even with the most rigorous security practices, a CA may be compromised. If this compromise is detected quickly, relying parties can remove the CA from their list of trusted CAs, and other CAs can revoke any certificates issued to the CA. However, CA compromise may go undetected for some time, and there's always the possibility that a CA is being operated improperly or in a manner that is not in the interests of the relying parties. For this reason, relying parties may wish to "pin" a small number of particularly critical certificates (such as the certificate for the XMPP-Grid Controller). Once a certificate has been pinned, the relying party will not accept another certificate in its place unless the Administrator explicitly commands it to do so. This does not mean that the relying party will not check the revocation status of pinned certificates. However, the Administrator may still be consulted if a pinned certificate is revoked, since the CA and revocation process are not completely trusted.

5.4. Summary

XMPP-Grid's considerable value as a broker for security-sensitive data exchange distribution also makes the protocol and the network security elements that implement it a target for attack. Therefore, strong security has been included as a basic design principle within the XMPP-Grid design process.

The XMPP-Grid transport protocol provides strong protection against a variety of different attacks. In the event that a XMPP-Grid Node or XMPP-Grid Controller is compromised, the effects of this compromise have been reduced and limited with the recommended role-based authorization model and other provisions, and best practices for managing and protecting XMPP-Grid systems have been described. Taken together, these measures should provide protection commensurate with the threat to XMPP-Grid systems, thus ensuring that they fulfill their promise as a network security clearing-house.

6. Privacy Considerations

XMPP-Grid Nodes may publish information about endpoint health, network access, events (which may include information about what services an endpoint is accessing), roles and capabilities, and the identity of the end user operating the endpoint. Any of this published information may be queried by other XMPP-Grid Nodes and could potentially be used to correlate network activity to a particular end user.

Dynamic and static information brokered by a XMPP-Grid Controller, ostensibly for purposes of correlation by XMPP-Grid Nodes for intrusion detection, could be misused by a broader set of XMPP-Grid Nodes which hitherto have been performing specific roles with strict well-defined separation of duties.

Care should be taken by deployers of XMPP-Grid to ensure that the information published by XMPP-Grid Nodes does not violate agreements with end users or local and regional laws and regulations. This can be accomplished either by configuring XMPP-Grid Nodes to not publish certain information or by restricting access to sensitive data to trusted XMPP-Grid Nodes. That is, the easiest means to ensure privacy or protect sensitive data, is to omit or not share it at all.

Another consideration for deployers is to enable end-to-end encryption to ensure the data is protected from the data layer to data layer and thus protect it from the transport layer.

7. Acknowledgements

The authors would like to acknowledge the contributions, authoring and/or editing of the following people: Joseph Salowey, Lisa Lorenzin, Clifford Kahn, Henk Birkholz, Jessica Fitzgerald-McKay, Steve Hanna, and Steve Venema.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3922] Saint-Andre, P., "Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)", RFC 3922, DOI 10.17487/RFC3922, October 2004, <<http://www.rfc-editor.org/info/rfc3922>>.
- [RFC3923] Saint-Andre, P., "End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)", RFC 3923, DOI 10.17487/RFC3923, October 2004, <<http://www.rfc-editor.org/info/rfc3923>>.
- [RFC4422] Melnikov, A., Ed. and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, DOI 10.17487/RFC4422, June 2006, <<http://www.rfc-editor.org/info/rfc4422>>.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <<http://www.rfc-editor.org/info/rfc6120>>.
- [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, DOI 10.17487/RFC6121, March 2011, <<http://www.rfc-editor.org/info/rfc6121>>.

8.2. Informative References

- [I-D.ietf-uta-tls-bcp] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of TLS and DTLS", draft-ietf-uta-tls-bcp-11 (work in progress), February 2015.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <<http://www.rfc-editor.org/info/rfc2818>>.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <<http://www.rfc-editor.org/info/rfc5070>>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.

Authors' Addresses

Nancy Cam-Winget (editor)
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
USA

Email: ncamwing@cisco.com

Syam Appala
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
USA

Email: syaml@cisco.com

Scott Pope
Cisco Systems
5400 Meadows Road
Suite 300
Lake Oswego, OR 97035
USA

Email: scottp@cisco.com