

Internet Engineering Task Force (IETF)
Request for Comments: 8600
Category: Standards Track
ISSN: 2070-1721

N. Cam-Winget, Ed.
S. Appala
S. Pope
Cisco Systems
P. Saint-Andre
Mozilla
June 2019

Using Extensible Messaging and Presence Protocol (XMPP)
for Security Information Exchange

Abstract

This document describes how to use the Extensible Messaging and Presence Protocol (XMPP) to collect and distribute security incident reports and other security-relevant information between network-connected devices, primarily for the purpose of communication among Computer Security Incident Response Teams and associated entities. To illustrate the principles involved, this document describes such a usage for the Incident Object Description Exchange Format (IODEF).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 7841](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8600>.

Copyright Notice

Copyright (c) 2019 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 (<https://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
2. Terminology	3
3. Architecture	5
4. Workflow	6
5. Service Discovery	8
6. Publish-Subscribe	10
7. IANA Considerations	13
8. Security Considerations	14
8.1. Trust Model	14
8.2. Threat Model	16
8.3. Countermeasures	20
8.4. Summary	23
9. Privacy Considerations	24
10. Operations and Management Considerations	25
11. References	25
11.1. Normative References	25
11.2. Informative References	27
Acknowledgements	27
Authors' Addresses	28

1. Introduction

This document defines an architecture, i.e., "XMPP-Grid", as a method for using the Extensible Messaging and Presence Protocol (XMPP) [RFC6120] to collect and distribute security incident reports and other security-relevant information among network platforms, endpoints, and any other network-connected device, primarily for the purpose of communication among Computer Security Incident Response Teams and associated entities. In effect, this document specifies an Applicability Statement ([RFC2026], Section 3.2) that defines how to use XMPP for the exchange of security notifications on a controlled-access network among authorized entities.

Among other things, XMPP provides a publish-subscribe service [XEP-0060] that acts as a broker, enabling control-plane functions by which entities can discover available information to be published or consumed. Although such information can take the form of any structured data (XML, JSON, etc.), this document illustrates the principles of XMPP-Grid with examples that use the Incident Object Description Exchange Format (IODEF) [RFC7970]. That is, while other security information formats can be shared using XMPP, this document uses IODEF as one such example format that can be published and consumed using XMPP.

2. Terminology

This document uses XMPP terminology defined in [RFC6120] and [XEP-0060]. Because the intended audience for this document is those who implement and deploy security reporting systems, mappings are provided for the benefit of XMPP developers and operators.

Broker: A specific type of controller containing control-plane functions; as used here, the term refers to an XMPP publish-subscribe service.

Broker Flow: A method by which security incident reports and other security-relevant information are published and consumed in a mediated fashion through a Broker. In this flow, the Broker handles authorization of Consumers and Providers to Topics, receives messages from Providers, and delivers published messages to Consumers.

Consumer: An entity that contains functions to receive information from other components; as used here, the term refers to an XMPP publish-subscribe Subscriber.

Controller: A component containing control-plane functions that manage and facilitate information sharing or execute on security functions; as used here, the term refers to an XMPP server, which provides core message delivery [RFC6120] used by publish-subscribe entities.

Node: The XMPP term for a Topic.

Platform: Any entity that connects to the XMPP-Grid in order to publish or consume security-relevant information.

Provider: An entity that contains functions to provide information to other components; as used here, the term refers to an XMPP publish-subscribe Publisher.

Topic: A contextual information channel created on a Broker on which messages generated by a Provider are propagated in real time to one or more Consumers. Each Topic is limited to a specific type and format of security data (e.g., IODEF namespace) and provides an XMPP interface by which the data can be obtained.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

data either through the Broker (shown as "B" in Figure 1) or, in some cases, directly (shown as "C" in Figure 1). This document focuses primarily on the Broker Flow for information sharing ("direct flow" interactions can be used for specialized purposes, such as bulk data transfer, but methods for doing so are outside the scope of this document).

4. Workflow

Implementations of XMPP-Grid adhere to the following workflow:

- a. A Platform with a source of security data requests connection to the XMPP-Grid via a Controller.
- b. The Controller authenticates the Platform.
- c. The Platform establishes authorized privileges (e.g., privilege to publish and/or subscribe to one or more Topics) with a Broker.
- d. The Platform can publish security incident reports and other security-relevant information to a Topic, subscribe to a Topic, query a Topic, or perform any combination of these operations.
- e. A Provider unicasts its Topic updates to the Grid in real time through a Broker. The Broker handles replication and distribution of the Topic to Consumers. A Provider can publish the same or different data to multiple Topics.
- f. Any Platform on the Grid can subscribe to any Topic published to the Grid (as permitted by the authorization policy) and (as Consumers) will then receive a continual, real-time stream of updates from the Topics to which it is subscribed.

The general workflow is summarized in the figure below.

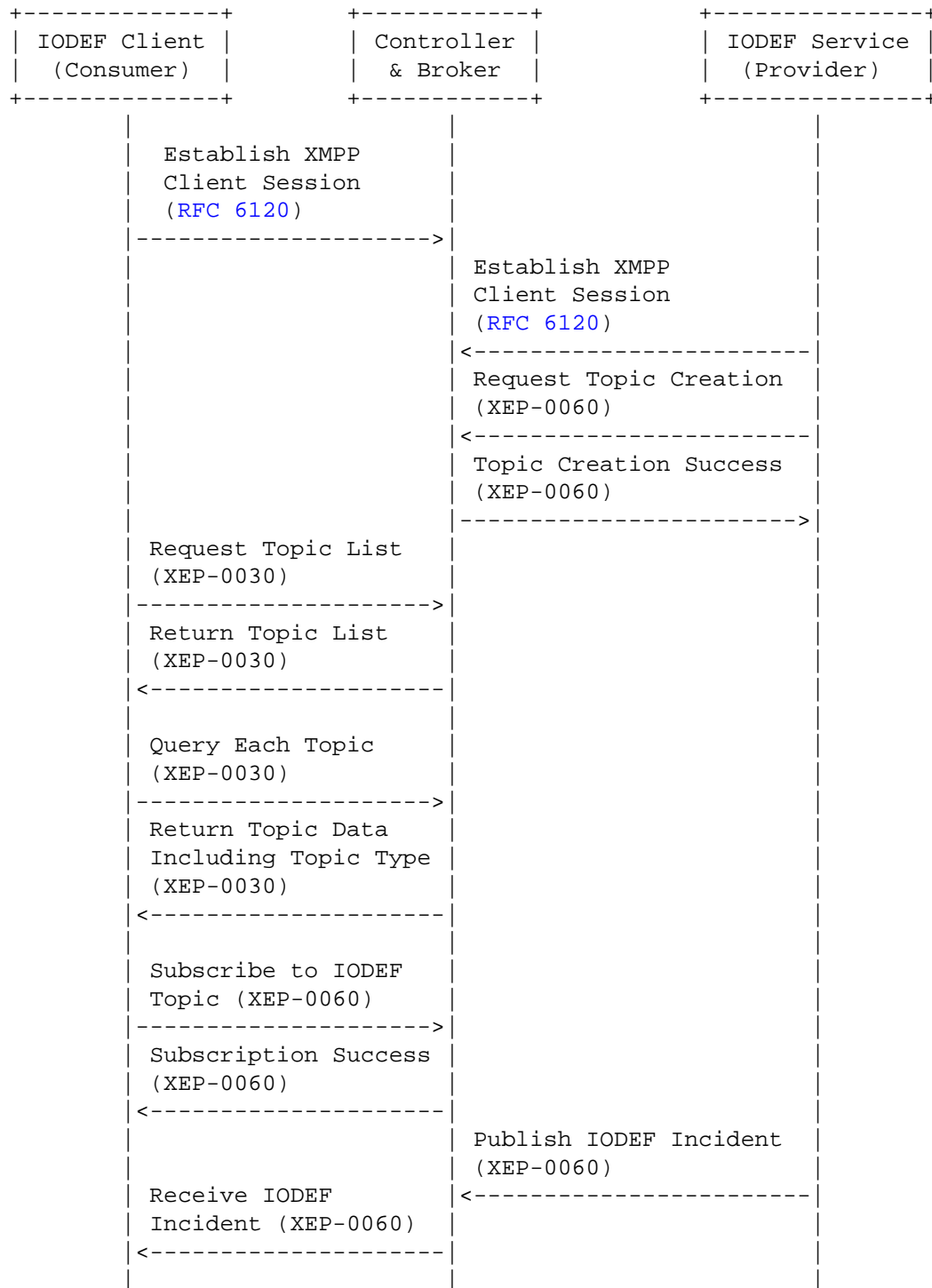


Figure 2: IODEF Example Workflow

XMPP-Grid implementations MUST adhere to the mandatory-to-implement and mandatory-to-negotiate features as defined in [RFC6120]. Similarly, implementations MUST implement the publish-subscribe extension [XEP-0060] to facilitate the asynchronous sharing of information. Implementations SHOULD implement Service Discovery as defined in [XEP-0030] to facilitate the means to dynamically discover the available information and namespaces (Topics) to be published or consumed. Care should be taken with implementations if their deployments allow for a large number of Topics. The result set management as defined in [XEP-0059] SHOULD be used to allow the requesting entity to explicitly request Service Discovery result sets to be returned in pages or a limited size, if the discovery results are larger in size. Note that the control plane may optionally also implement [XEP-0203] to facilitate delayed delivery of messages to the connected consumer as described in [XEP-0060]. Since information may be timely and sensitive, capability providers should communicate to the Controller whether its messages can be cached for delayed delivery during configuration; such a function is out of scope for this document.

The following sections provide protocol examples for the service discovery and publish-subscribe parts of the workflow.

5. Service Discovery

Using the XMPP service discovery extension [XEP-0030], a Controller enables Platforms to discover what information can be consumed through the Broker and at which Topics. Platforms could use [XEP-0059] to restrict the size of the result sets the Controller returns in a Service Discovery response. As an example, the Controller at 'security-grid.example' might provide a Broker at 'broker.security-grid.example', which hosts a number of Topics. A Platform at 'xmpp-grid-client@mile-host.example' would query the Broker about its available Topics by sending an XMPP "disco#items" request to the Broker:

```
<iq type='get'
  from='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  to='broker.security-grid.example'
  id='B3C17F7B-B9EF-4ABA-B08D-805DA9F34626'>
  <query xmlns='http://jabber.org/protocol/disco#items'/>
</iq>
```


The Broker responds with the Topics it hosts:

```
<iq type='result'
  from='broker.security-grid.example'
  to='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  id='B3C17F7B-B9EF-4ABA-B08D-805DA9F34626'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    <item node='NEA1'
      name='Endpoint Posture Information'
      jid='broker.security-grid.example' />
    <item node='MILEHost'
      name='MILE Host Data'
      jid='broker.security-grid.example' />
  </query>
</iq>
```

In order to determine the exact nature of each Topic (i.e., in order to find Topics that publish incidents in the IODEF format), a Platform would send an XMPP "disco#info" request to each Topic:

```
<iq type='get'
  from='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  to='broker.security-grid.example'
  id='D367D4ED-2795-489C-A83E-EAAFA07A0356'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='MILEHost' />
</iq>
```

The Broker responds with the "disco#info" description, which MUST include an XMPP data form [XEP-0004] with a 'pubsub#type' field that specifies the supported namespace (in this example, the IODEF namespace defined in [RFC7970]):

```
<iq type='result'
  from='broker.security-grid.example'
  to='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  id='D367D4ED-2795-489C-A83E-EAAFA07A0356'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='MILEHost'>
    <identity category='pubsub' type='leaf' />
    <feature var='http://jabber.org/protocol/pubsub' />
    <x xmlns='jabber:x:data' type='result'>
      <field var='FORM_TYPE' type='hidden'>
        <value>http://jabber.org/protocol/pubsub#meta-data</value>
      </field>
      <field var='pubsub#type' label='Payload type' type='text-single'>
        <value>urn:ietf:params:xml:ns:iodef-2.0</value>
      </field>
    </x>
  </query>
</iq>
```

The Platform discovers the Topics by obtaining the Broker's response and obtaining the namespaces returned in the "pubsub#type" field (in the foregoing example, IODEF 2.0).

6. Publish-Subscribe

Using the XMPP publish-subscribe extension [XEP-0060], a Consumer subscribes to a Topic, and a Provider publishes information to that Topic, which the Broker then distributes to all subscribed Consumers.

First, a Provider would create a Topic as follows:

```
<iq type='set'
  from='datasource@provider.example/F12C2EFC9BB0'
  to='broker.security-grid.example'
  id='A67507DF-2F22-4937-8D30-88D2F7DBA279'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <create node='MILEHost' />
  </pubsub>
</iq>
```

Note: The foregoing example is the minimum protocol needed to create a Topic with the default node configuration on the XMPP publish-subscribe service specified in the 'to' address of the creation

request stanza. Depending on security requirements, the Provider might need to request a non-default configuration for the node; see [XEP-0060] for detailed examples. To also help with the Topic configuration, the Provider may also optionally include configuration parameters such as:

```
<configure>
  <x xmlns='jabber:x:data' type='submit'>
    <field var='FORM_TYPE' type='hidden'>
      <value>http://jabber.org/protocol/pubsub#node_config</value>
    </field>
    <field var='pubsub#access_model'><value>authorize</value></field>
    <field var='pubsub#persist_items'><value>1</value></field>
    <field var='pubsub#send_last_published_item'>
      <value>never</value>
    </field>
  </x>
</configure>
```

The above configuration indicates the Topic is configured so that the Broker will a) explicitly approve subscription requests, b) persistently store all messages posted to the Topic, and c) not deliver the most recently published when an entity initially subscribes to the Topic. Please refer to [XEP-0060] for a more detailed description of this configuration and other available configuration options.

Unless an error occurs (see [XEP-0060] for various error flows), the Broker responds with success:

```
<iq type='result'
  from='broker.security-grid.example'
  to='datasource@provider.example/F12C2EFC9BB0'
  id='A67507DF-2F22-4937-8D30-88D2F7DBA279' />
```

Second, a Consumer would subscribe as follows:

```
<iq type='set'
  from='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  to='broker.security-grid.example'
  id='9C6EEE9E-F09A-4418-8D68-3BA6AF852522'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscribe node='MILEHost'
      jid='xmpp-grid-client@mile-host.example' />
  </pubsub>
</iq>
```

Unless an error occurs (see [XEP-0060] for various error flows), the Broker makes an appropriate authorization decision. If access is granted, it responds with success:

```
<iq type='result'
  from='broker.security-grid.example'
  to='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  id='9C6EEE9E-F09A-4418-8D68-3BA6AF852522'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <subscription
      node='MILEHost'
      jid='xmpp-grid-client@mile-host.example'
      subscription='subscribed' />
    </pubsub>
  </iq>
```

Third, a Provider would publish an incident to the Broker using the MILEHost Topic as follows:

```
<iq type='set'
  from='datasource@provider.example/F12C2EFC9BB0'
  to='broker.security-grid.example'
  id='2A17D283-0DAE-4A6C-85A9-C10B1B40928C'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='MILEHost'>
      <item id='8bhlq27skbga47fh9wk7'>
        <IODEF-Document version="2.00" xml:lang="en"
          xmlns="urn:ietf:params:xml:ns:iodef-2.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation=
            "http://www.iana.org/assignments/xml-registry/
            schema/iodef-2.0.xsd">
          <Incident purpose="reporting" restriction="private">
            <IncidentID name="csirt.example.com">492382</IncidentID>
            <GenerationTime>2015-07-18T09:00:00-05:00</GenerationTime>
            <Contact type="organization" role="creator">
              <Email>
                <EmailTo>contact@csirt.example.com</EmailTo>
              </Email>
            </Contact>
          </Incident>
        </IODEF-Document>
      </item>
    </publish>
  </pubsub>
</iq>
```

(The payload in the foregoing example is from [RFC7970]; payloads for additional use cases can be found in [RFC8274].)

The Broker would then deliver that incident report to all Consumers who are subscribed to the Topic:

```
<message
  from='broker.security-grid.example'
  to='xmpp-grid-client@mile-host.example/2EBE702A97D6'
  id='37B3921D-4F7F-450F-A589-56119A88BC2E'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='MILEHost'>
      <item id='iah37s6ls964gquqy47aksbx9453ks77'>
        <IODEF-Document version="2.00" xml:lang="en"
          xmlns="urn:ietf:params:xml:ns:iodef-2.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation=
            "http://www.iana.org/assignments/xml-registry/
            schema/iodef-2.0.xsd">
          <Incident purpose="reporting" restriction="private">
            <IncidentID name="csirt.example.com">492382</IncidentID>
            <GenerationTime>2015-07-18T09:00:00-05:00</GenerationTime>
            <Contact type="organization" role="creator">
              <Email>
                <EmailTo>contact@csirt.example.com</EmailTo>
              </Email>
            </Contact>
          </Incident>
        </IODEF-Document>
      </item>
    </items>
  </event>
</message>
```

Note that [XEP-0060] uses the XMPP "<message />" stanza for delivery of content. To ensure that messages are delivered to the Consumer even if the Consumer is not online at the same time that the Publisher generates the message, an XMPP-Grid Controller MUST support "offline messaging" delivery semantics as specified in [RFC6121], the best practices of which are further explained in [XEP-0160].

7. IANA Considerations

This document has no actions for IANA.

8. Security Considerations

An XMPP-Grid Controller serves as a controlling broker for XMPP-Grid Platforms such as enforcement points, policy servers, Configuration Management Databases (CMDBs), and sensors, using a publish-subscribe-search model of information exchange and lookup. By increasing the ability of XMPP-Grid Platforms to learn about and respond to security incident reports and other security-relevant information, 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.

As XMPP is the core of this document, the security considerations of [RFC6120] apply. In addition, as XMPP-Grid defines a specific instance, this section provides a security analysis of the XMPP-Grid data transfer 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 can be mounted on the system), and the countermeasures (ways to address or mitigate the threats previously identified).

8.1. Trust Model

The first step in analyzing the security of the XMPP-Grid transport protocol is to describe the trust model and list what each architectural element is trusted to do. The items listed here are assumptions, but provisions are made in "Threat Model" (Section 8.2) and "Countermeasures" (Section 8.3) for elements that fail to perform as they were trusted to do.

8.1.1. Network

The network used to carry XMPP-Grid messages (i.e., the underlying network transport layer over which XMPP runs) 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

8.1.2. XMPP-Grid Platforms

Authorized XMPP-Grid Platforms are trusted to:

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

8.1.3. XMPP-Grid Controller

The XMPP-Grid Controller (including its associated Broker) 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
- o Preserve the integrity (and confidentiality against unauthorized parties) of the data flowing through it.

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

- o Verify the truth (correctness) of data

8.1.4. Certification Authority

To allow XMPP-Grid Platforms to mutually authenticate with XMPP-Grid Controllers, it is expected that a Certification Authority (CA) is employed to issue certificates. Such a CA (or each CA, if there are several) is trusted to:

- 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 the XMPP-Grid needs or other constraints imposed by a relying party.

8.2. Threat Model

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

8.2.1. Network Attacks

A variety of attacks can be mounted using the network. For the purposes of this subsection, the phrase "network traffic" can be taken to mean messages and/or parts of messages. Any of these attacks can 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 can 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 can be modified in transit
- o Previously transmitted network traffic can be replayed
- o New network traffic can be added
- o Network traffic can be blocked, perhaps selectively
- o A man-in-the-middle (MITM) attack can be mounted where an attacker interposes itself between two communicating parties and impersonates the other end to either or both parties
- o Undesired network traffic can be sent in an effort to overload an architectural component, thus mounting a denial-of-service attack

8.2.2. XMPP-Grid Platforms

An unauthorized XMPP-Grid Platform (one that 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 "Network Attacks" ([Section 8.2.1](#)).

An authorized XMPP-Grid Platform, on the other hand, can mount many attacks. These attacks might occur because the XMPP-Grid Platform is controlled by a malicious, careless, or incompetent party (whether because its owner is malicious, careless, or incompetent or because the XMPP-Grid Platform has been compromised and is now controlled by a party other than its owner). They might also occur because the XMPP-Grid Platform is running malicious software, the XMPP-Grid Platform is running buggy software (which can fail in a state that floods the network with traffic), or the XMPP-Grid Platform 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 can be mounted by an authorized XMPP-Grid Platform:

- 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 parts of the network security system being disabled.
- 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 Platforms with the 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 Platform, and/or the network.
- o Establish a communication channel using another XMPP-Grid Platform's session-id.
- o Advertise false data that leads to incorrect (e.g., potentially attacker-controlled or -induced) behavior of XMPP-Grid Platforms by virtue of applying correct procedures to the falsified input.

Dependencies or vulnerabilities of authorized XMPP-Grid Platforms can be exploited to effect these attacks. Another way to effect these attacks is to gain the ability to impersonate an XMPP-Grid Platform (through theft of the XMPP-Grid Platform's identity credentials or

through other means). Even a clock skew between the XMPP-Grid Platform and XMPP-Grid Controller can cause problems if the XMPP-Grid Platform assumes that old XMPP-Grid Platform data should be ignored.

8.2.3. XMPP-Grid Controllers

An unauthorized XMPP-Grid Controller (one that is not trusted by XMPP-Grid Platforms) cannot mount any attacks other than those listed in "Network Attacks" ([Section 8.2.1](#)).

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

All of the attacks listed for XMPP-Grid Platform 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 can include:

- o Exposing different data to different XMPP-Grid Platforms to mislead investigators or cause inconsistent behavior.
- o Mounting an even more effective denial-of-service attack than a single XMPP-Grid Platform could; some mechanisms include inducing many platforms to perform the same operation in an amplification-style attack, completely refusing to pass any traffic at all, or sending floods of traffic to (certain) platforms or other targets.
- o Obtaining and caching XMPP-Grid Platform credentials so they can be used to impersonate XMPP-Grid Platforms even after a breach of the XMPP-Grid Controller is repaired. Some Simple Authentication and Security Layer (SASL) mechanisms (including the mandatory-to-implement SCRAM and EXTERNAL with TLS mutual certificate-based authentication) do not admit this class of attack, but others (such as PLAIN) are susceptible.

- o Obtaining and caching 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.
- o Eavesdropping, injecting, or modifying the data being transferred between Provider and Consumer.

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

8.2.4. Certification Authority

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

- o Issue certificates for unauthorized parties, enabling them to impersonate authorized parties such as the XMPP-Grid Controller or an XMPP-Grid Platform. 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 can 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 can 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.

8.3. Countermeasures

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

8.3.1. Securing the XMPP-Grid Data Transfer Protocol

To address network attacks, the XMPP-Grid data transfer protocol described in this document requires that the XMPP-Grid messages **MUST** be carried over TLS (minimally TLS 1.2 and preferably TLS 1.3 [RFC8446]) as described in [RFC6120] and updated by [RFC7590]. The XMPP-Grid Controller and XMPP-Grid Platforms **SHOULD** mutually authenticate. The XMPP-Grid Platform **MUST** verify the XMPP-Grid Controller's certificate and determine whether the XMPP-Grid Controller is trusted by this XMPP-Grid Platform before completing the TLS handshake. To ensure interoperability, implementations **MUST** implement at least either the SASL EXTERNAL mechanism [RFC4422] or the SASL SCRAM mechanism. When using the SASL SCRAM mechanism, the SCRAM-SHA-256-PLUS variant **SHOULD** be preferred over the SCRAM-SHA-256 variant, and SHA-256 variants [RFC7677] **SHOULD** be preferred over SHA-1 variants [RFC5802]). XMPP-Grid Platforms and XMPP-Grid Controllers using certificate-based authentication **SHOULD** each verify the revocation status of the other party's certificate. The selection of the XMPP-Grid Platform authentication technique to use in any particular deployment is left to the administrator.

These protocol security measures provide protection against all the network attacks listed in [Section 8.2](#) 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 can be employed. In addition, an XMPP-Grid Controller **MAY** automatically disable a misbehaving XMPP-Grid Platform.

8.3.2. Securing XMPP-Grid Platforms

XMPP-Grid Platforms can be deployed in locations that are susceptible to physical attacks. Physical security measures can be taken to avoid compromise of XMPP-Grid Platforms, but these are not always 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 Platforms can 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 Platform as a DHCP [RFC2131] server and authorize only the operations and metadata types needed by a DHCP server to be permitted for that XMPP-Grid Platform. These techniques can reduce the negative impacts of a compromised XMPP-Grid Platform 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 Platform behavior. XMPP-Grid Controllers SHOULD make it easy to revoke an XMPP-Grid Platform's authorization when necessary. The XMPP-Grid Controller SHOULD include auditable logs of XMPP-Grid Platform activities.

To avoid compromise of XMPP-Grid Platforms, they SHOULD be hardened against attack and minimized to reduce their attack surface. They should be well managed to minimize vulnerabilities in the underlying platform and in systems upon which the XMPP-Grid Platform depends. Personnel with administrative access should be carefully screened and monitored to detect problems as soon as possible.

8.3.3. Securing XMPP-Grid Controllers

Because of the serious consequences of XMPP-Grid Controller compromise, XMPP-Grid Controllers need to be especially well hardened against attack and minimized to reduce their attack surface. They need to 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 can be used to monitor and limit traffic to and from the XMPP-Grid Controller. Personnel with administrative access ought to 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 ought to 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 Platforms). It is a matter of local policy whether XMPP-Grid Platforms log and/or notify administrators when peculiar XMPP-Grid Controller behavior is detected and whether read-only audit logs of security-relevant information (especially administrative actions) are maintained; however, such behavior is encouraged to aid in forensic analysis.

Furthermore, if compromise of an XMPP-Grid Controller is detected, a careful analysis should be performed, and any reusable credentials that could have been compromised should be reissued.

To address the potential for the XMPP-Grid Controller to eavesdrop, modify or inject data, it would be desirable to deploy end-to-end encryption between the Provider and the Consumer(s). Unfortunately, because there is no standardized method for encryption of one-to-many messages within XMPP, techniques for enforcing end-to-end encryption are out of scope for this specification.

8.3.4. Broker Access Models for Topics

The XMPP publish-subscribe specification [XEP-0060] defines five access models for subscribing to Topics at a Broker: open, presence, roster, authorize, and whitelist. The first model allows uncontrolled access, and the next two models are appropriate only in instant-messaging applications. Therefore, a Broker SHOULD support only the authorize model (under which the Topic owner needs to approve all subscription requests and only subscribers can retrieve data items) and the whitelist model (under which only preconfigured Platforms can subscribe or retrieve data items). In order to ease the deployment burden, subscription approvals and whitelist management can be automated (e.g., the Topic "owner" can be a policy server). The choice between "authorize" and "whitelist" as the default access model is a matter for local service policy.

8.3.5. Limit on Search Result Size

While XMPP-Grid is designed for high scalability to hundreds of thousands of Platforms, an XMPP-Grid Controller MAY establish a limit to the amount of data it is willing to return in search or subscription results. Platforms could use [XEP-0059] to restrict the size of the result sets the Controller returns in search or subscription results or topics' service discovery. This mitigates the threat of an XMPP-Grid Platform causing resource exhaustion by issuing a search or subscription that leads to an enormous result.

8.3.6. 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 Platforms is a major security breach. Many guidelines for proper CA security have been developed: the CA/Browser Forum's Baseline Requirements, the American Institute of Certified Public Accountants (AICPA) / Canadian Institute of Chartered Accountants (CICA) Trust

Service Principles, the IETF's Certificate Transparency [RFC6962], etc. The CA operator and relying parties should agree on appropriately rigorous security practices to be used.

Even with the most rigorous security practices, a CA can 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 can still be consulted if a pinned certificate is revoked, since the CA and revocation process are not completely trusted. By "pinning" one or a small set of certificates, the relying party has identified the effective XMPP-Grid Controller(s) authorized for connection.

8.3.7. End-to-End Encryption of Messages

Because it is expected that there will be a relatively large number of Consumers for every Topic, for the purposes of content discovery and scaling, this document specifies a "one-to-many" communications pattern using the XMPP Publish-Subscribe extension. Unfortunately, there is no standardized technology for end-to-end encryption of one-to-many messages in XMPP. This implies that messages can be subject to eavesdropping, data injection, and data modification attacks within a Broker or Controller. If it is necessary to mitigate against such attacks, implementers would need to select a messaging pattern other than [XEP-0060], most likely the basic "instant messaging" pattern specified in [RFC6121] with a suitable XMPP extension for end-to-end encryption. The description of such an approach is out of scope for this document.

8.4. 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 data transfer protocol provides strong protection against a variety of different attacks. In the event that an XMPP-Grid Platform or XMPP-Grid Controller is compromised, the effects of this compromise are 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 clearinghouse.

9. Privacy Considerations

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

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

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

Similarly, care must be taken by deployers and XMPP-Grid Controller implementations as they implement the appropriate auditing tools. In particular, any information, such as logs, must be sensitive to the type of information stored to ensure that the information does not violate privacy and agreements with end users or local and regional laws and regulations.

Another consideration for deployers is to enable end-to-end encryption to ensure the data is protected while in transit between data layers and thus protected from the transport layer. The means to achieve end-to-end encryption is beyond the scope of this document.

10. Operations and Management Considerations

In order to facilitate the management of Providers and the onboarding of Consumers, it is helpful to generate the following ahead of time:

- o Agreement between the operators of Provider services and the implementers of Consumer software regarding identifiers for common Topics (e.g., these could be registered with the XMPP Software Foundation's registry of well-known nodes for service discovery and publish-subscribe, located at <<https://xmpp.org/registrar/nodes.html>>).
- o Security certificates (including appropriate certificate chains) for Controllers, including identification of any Providers associated with the Controllers (which might be located at subdomains).
- o Consistent and secure access control policies for publishing and subscribing to Topics.

These matters are out of scope for this document but ought to be addressed by the XMPP-Grid community.

11. References

11.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4422] Melnikov, A., Ed. and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", [RFC 4422](#), DOI 10.17487/RFC4422, June 2006, <<https://www.rfc-editor.org/info/rfc4422>>.
- [RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", [RFC 5802](#), DOI 10.17487/RFC5802, July 2010, <<https://www.rfc-editor.org/info/rfc5802>>.

- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 6120](#), DOI 10.17487/RFC6120, March 2011, <<https://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, <<https://www.rfc-editor.org/info/rfc6121>>.
- [RFC7590] Saint-Andre, P. and T. Alkemade, "Use of Transport Layer Security (TLS) in the Extensible Messaging and Presence Protocol (XMPP)", [RFC 7590](#), DOI 10.17487/RFC7590, June 2015, <<https://www.rfc-editor.org/info/rfc7590>>.
- [RFC7677] Hansen, T., "SCRAM-SHA-256 and SCRAM-SHA-256-PLUS Simple Authentication and Security Layer (SASL) Mechanisms", [RFC 7677](#), DOI 10.17487/RFC7677, November 2015, <<https://www.rfc-editor.org/info/rfc7677>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007, <<https://xmpp.org/extensions/xep-0004.html>>.
- [XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, October 2017, <<https://xmpp.org/extensions/xep-0030.html>>.
- [XEP-0059] Paterson, I., Saint-Andre, P., Mercier, V., and J. Segueineau, "Result Set Management", XSF XEP 0059, September 2006, <<https://xmpp.org/extensions/xep-0059.html>>.
- [XEP-0060] Millard, P., Saint-Andre, P., and R. Meijer, "Publish-Subscribe", XSF XEP 0060, January 2019, <<https://xmpp.org/extensions/xep-0060.html>>.
- [XEP-0203] Saint-Andre, P., "Delayed Delivery", XSF XEP 0203, September 2009, <<https://xmpp.org/extensions/xep-0203.html>>.

11.2. Informative References

- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", [RFC 6962](#), DOI 10.17487/RFC6962, June 2013, <<https://www.rfc-editor.org/info/rfc6962>>.
- [RFC7970] Danyliw, R., "The Incident Object Description Exchange Format Version 2", [RFC 7970](#), DOI 10.17487/RFC7970, November 2016, <<https://www.rfc-editor.org/info/rfc7970>>.
- [RFC8274] Kampanakis, P. and M. Suzuki, "Incident Object Description Exchange Format Usage Guidance", [RFC 8274](#), DOI 10.17487/RFC8274, November 2017, <<https://www.rfc-editor.org/info/rfc8274>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [RFC 8446](#), DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [XEP-0160] Saint-Andre, P., "Best Practices for Handling Offline Messages", XSF XEP 0160, October 2016, <<https://xmpp.org/extensions/xep-0160.html>>.

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. In addition, we want to thank Takeshi Takahashi, Panos Kampanakis, Adam Montville, Chris Inacio, and Dave Cridland for reviewing and providing valuable comments.

Authors' Addresses

Nancy Cam-Winget (editor)
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
United States of America

Email: ncamwing@cisco.com

Syam Appala
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
United States of America

Email: syaml@cisco.com

Scott Pope
Cisco Systems
5400 Meadows Road
Suite 300
Lake Oswego, OR 97035
United States of America

Email: scottp@cisco.com

Peter Saint-Andre
Mozilla

Email: stpeter@mozilla.com