MILE
Internet-Draft
Intended status: Standards Track
Expires: August 12, 2018

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

February 8, 2018

Using XMPP for Security Information Exchange
draft-ietf-mile-xmpp-grid-05

Abstract

This document describes how to use the Extensible Messaging and Presence Protocol (XMPP) to collect and distribute security-relevant information between network-connected devices. To illustrate the principles involved, this document describes such a usage for the Incident Object Description Exchange Format (IODEF).

Table of Contents

1. Introduction

This document describes "XMPP-Grid": a method for using the Extensible Messaging and Presence Protocol (XMPP) [RFC6120] to collect and distribute security-relevant information among network platforms, endpoints, and any other network-connected device. 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].

2. Terminology

This document uses XMPP terminology defined in [RFC6120] and [XEP-0060] as well as Security Automation and Continuous Monitoring (SACM) terminology defined in [I-D.ietf-sacm-terminology]. Because the intended audience for this document is those who implement and deploy security reporting systems, in general the SACM terms are used (however, mappings are provided for the benefit of XMPP developers and operators).

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

Broker Flow: In SACM, a method by which security-related information is 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: In SACM, an entity that contains functions to receive information from other components; as used here, the term refers to an XMPP publish-subscribe Subscriber.

Controller: In SACM, 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-related data.

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

Publisher: The XMPP term for a Provider.

Publish-Subscribe Service: The XMPP term for the kind Broker discussed here.

Subscriber: The XMPP term for a Consumer.

Topic: A contextual information channel created on a Broker at 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) 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Architecture

The following figure illustrates the architecture of XMPP-Grid.

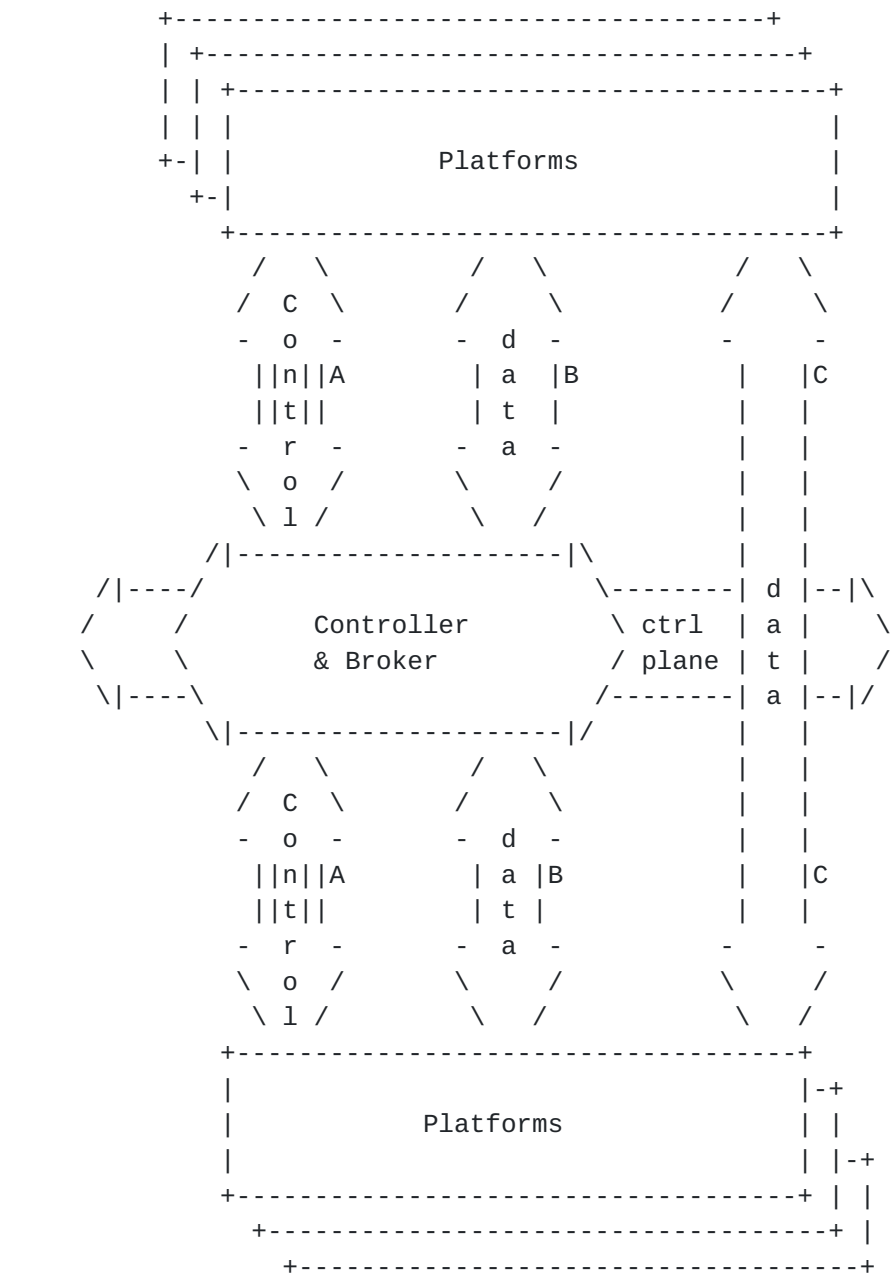

Figure 1: XMPP-Grid Architecture

Platforms connect to the Controller (XMPP server) to authenticate and then establish appropriate authorizations and relationships (e.g., Provider or Consumer) at the Broker. The control plane messaging is established through XMPP and shown as "A" (control plane interface) in Figure 1. Authorized nodes can then share data either thru 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

A typical XMPP-Grid workflow is as follows:

a. A Platform with a source of security data requests connection to the XMPP-Grid via a Controller (XMPP server).

b. The Controller authenticates the Platform.

c. The Platform establishes authorized privileges (e.g. privilege to publish and/or subscribe to security data Topics) with a Broker.

d. The Platform can publish security-related data to a Topic, subscribe to a Topic, query a Topic, or 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 Topics published to the Grid (as permitted by 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:

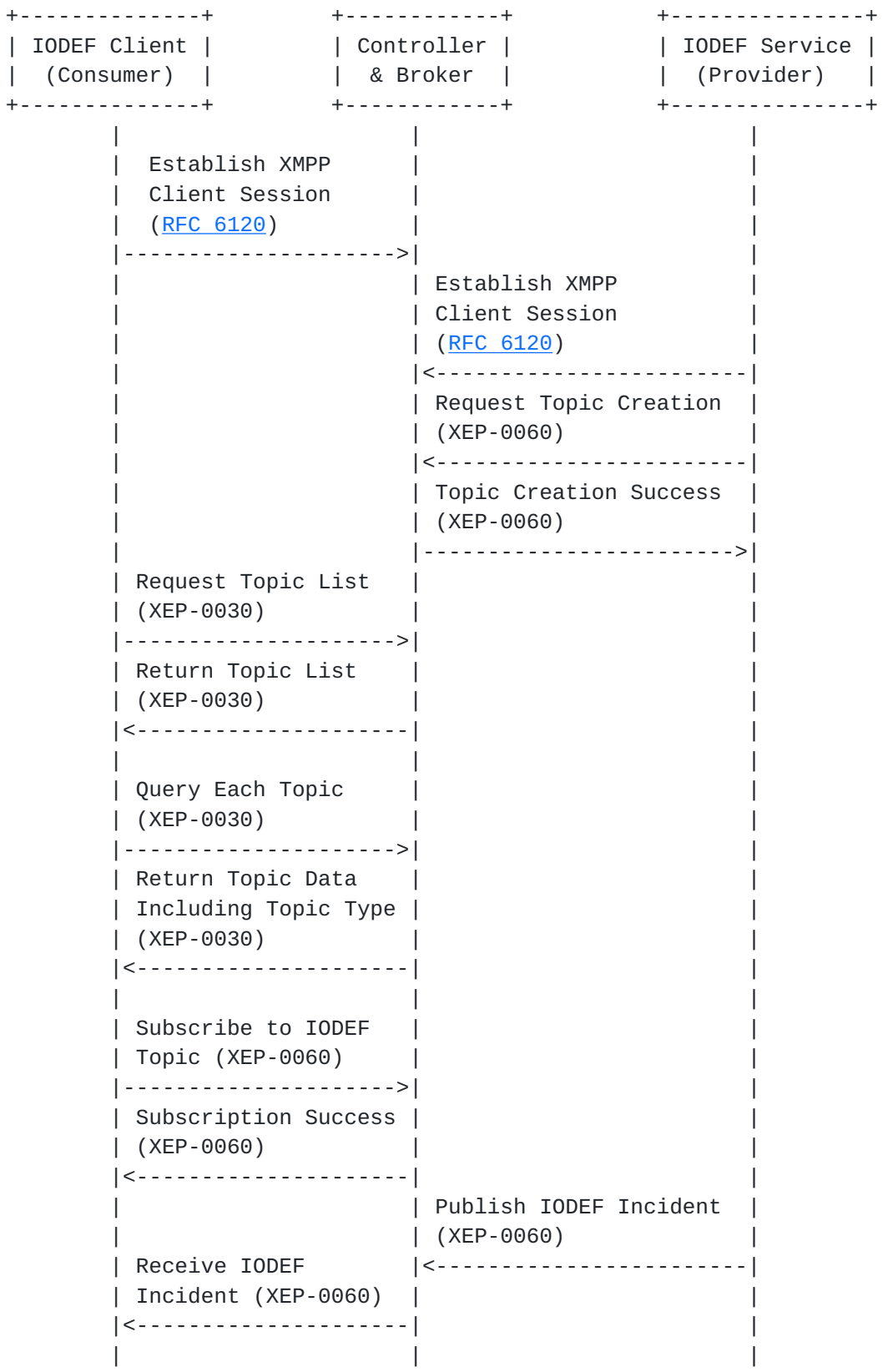

Figure 2: IODEF Example Workflow

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. As an example, the Controller at 'security-grid.example' might provide a Broker at 'broker.security-grid.example' hosting 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 SHOULD include an XMPP Data Form [XEP-0004] including 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>
```

6. Publish-Subscribe

Using the XMPP publish-subscribe extension [XEP-0030], 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 minimal 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.

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 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 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='8bh1g27skbga47fh9wk7'>
        <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 subscribe 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='iah37s61s964gquqy47aksbx9453ks77'>
        <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>
```

7. IANA Considerations

This document has no actions for IANA.

8. Security Considerations

An XMPP-Grid Controller serves as an controlling broker for XMPP-Grid Platforms 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 Platforms 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 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, 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.

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:

- Perform best effort delivery of network traffic

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

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

8.1.2. XMPP-Grid Platforms

Authorized XMPP-Grid Platforms are trusted to:

- 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:

- Broker requests for data and enforce authorization of access to this data throughout its lifecycle
- Perform service requests in a timely and accurate manner
- Create and maintain accurate operational attributes

- Only reveal data to and accept service requests from authorized parties

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

- Verify the truth (correctness) of data

8.1.4. Certification Authority

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

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

The CA is not expected (trusted) to:

- 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.

- Network traffic can be passively monitored to glean information from any unencrypted traffic

- Even if all traffic is encrypted, valuable information can be gained by traffic analysis (volume, timing, source and destination addresses, etc.)

- Network traffic can be modified in transit

- Previously transmitted network traffic can be replayed

- New network traffic can be added

- Network traffic can be blocked, perhaps selectively

- A "Man In The Middle" (MITM) attack can 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

- Resist attacks (including denial of service and other attacks from XMPP-Grid Platforms)

- 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 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 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; because the XMPP-Grid Platform is running buggy software (which can fail in a state that floods the network with traffic); or because 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:

- 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

- Omit important actions (such as posting incriminating data), resulting in incorrect access
- Use confidential information obtained from the XMPP-Grid Controller to enable further attacks (such as using endpoint health check results to exploit vulnerable endpoints)
- Advertise data crafted to exploit vulnerabilities in the XMPP-Grid Controller or in other XMPP-Grid Platforms, with a goal of compromising those systems
- 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
- Establish a communication channel using another XMPP-Grid Platform's session-id

Dependencies of 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 deserves to be ignored.

8.2.3. XMPP-Grid Controllers

An unauthorized XMPP-Grid Controller (one which is not trusted by XMPP-Grid Platforms) 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 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, because the XMPP-Grid Controller is running buggy software (which can 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 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:

- Expose different data to different XMPP-Grid Platforms to mislead investigators or cause inconsistent behavior
- Mount an even more effective denial of service attack than a single XMPP-Grid Platform could
- Obtain and cache 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
- 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 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:

- 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.
- 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.
- 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.

- 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.

- 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.

- 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 [RFC5246]) as described in [RFC6120] and updated by [RFC7590]. 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. The XMPP-Grid Controller MUST authenticate the XMPP-Grid Platform either using the SASL EXTERNAL mechanism or using the SASL SCRAM mechanism (with the SCRAM-SHA-256-PLUS variant being preferred over the SCRAM-SHA-256 variant and SHA-256 variants [RFC7677] being preferred over SHA-1 varients [RFC5802]). XMPP-Grid Platforms and XMPP-Grid Controllers using mutual certificate-based authentication SHOULD each verify the revocation status of the other party's certificate. All XMPP-Grid Controllers and XMPP-Grid Platforms MUST implement both SASL EXTERNAL and SASL SCRAM. The selection of which 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 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 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 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. Another way to detect attacks from XMPP-Grid Platforms is to create fake entries in the available data (honeytokens) which normal XMPP-Grid Platforms will not attempt to access. The XMPP-Grid Controller SHOULD include auditable logs of XMPP-Grid Platform activities.

To avoid compromise of XMPP-Grid Platform, XMPP-Grid Platform 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). XMPP-Grid Platforms 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 can have been compromised should be reissued.

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 100,000s 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. 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 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 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.

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 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.

9. Privacy Considerations

XMPP-Grid Platforms can publish information about endpoint health, network access, events (which can 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 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 purposes of correlation by XMPP-Grid Platforms for intrusion detection, could be misused by a broader set of XMPP-Grid Platforms which hitherto have been performing specific roles with 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.

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.

10. 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.

11. References

11.1. Normative References

[I-D.ietf-sacm-terminology]
Birkholz, H., Lu, J., Strassner, J., and N. Cam-Winget, "Secure Automation and Continuous Monitoring (SACM) Terminology", draft-ietf-sacm-terminology-14 (work in progress), December 2017.

[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>.

[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>.

[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>.

[XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007.

[XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, July 2010.

[XEP-0060] Millard, P., Saint-Andre, P., and R. Meijer, "Publish-Subscribe", XSF XEP 0060, December 2017.

11.2. Informative References

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

[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>.

[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>.

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: syam1@cisco.com

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

Email: scottp@cisco.com

Peter Saint-Andre
Mozilla

Email: stpeter@mozilla.com