Internet Draft
Document: draft-zander-ipfix-diameter-eval-00.txt
Expires: February 2003

Sebastian Zander
Fraunhofer FOKUS

September 2002

Evaluation of Diameter Protocol against IPFIX Requirements

<draft-zander-ipfix-diameter-eval-00.txt>

Status of this Memo

Copyright Notice

Abstract

This document provides an evaluation of the applicability of the Diameter protocol [DIAMETER] as an IPFIX protocol. It compares the properties and capabilities of the Diameter protocol to the IPFIX requirements [IPFIX-REQ].

Table of Contents

1. Introduction

This document provides an evaluation of the applicability of the Diameter protocol as an IPFIX protocol. First, the general Diameter architecture is introduced and its application to the communication between an IPFIX exporting process and an IPFIX collecting process is discussed in Section 2. Section 3 discusses in detail, to which degree requirements stated in [IPFIX-REQ] are met.

This document uses the terminology defined in [IPFIX-REQ].

The Diameter protocol is the successor of the RADIUS protocol and was developed to overcome several limitations of RADIUS. The Diameter protocol was developed for the purpose of Authentication, Authorization and Accounting (AAA) and is standardized by the IETF Authentication, Authorization and Accounting Working Group (AAA WG). The Diameter base protocol is specified in draft-ietf-aaa-diameter-12.txt [DIAMETER]. The Diameter base protocol provides a AAA framework. Specific applications require extensions to the base protocol and are called Diameter applications. Currently applications are being standardized by the AAA WG for dial-up and mobile IPv4 services. These specific applications are considered out of scope for the purpose of this document. However there are two companion drafts to the Diameter base protocol which need to be considered here: the strong end-to-end security extension [DIAM_CMS] and the transport recommendations for the Diameter base protocol [AAA_TRANS].

1.1. Diameter Standardization

The Diameter base protocol has passed Working Group last call after thorough review. The draft will now be reviewed by the IESG and afterwards become a Proposed Standard. The transport recommendations draft is very mature while the CMS draft needs more work.

The Diameter protocol will become an open IETF standard and is not protected by any patents. The IETF copyright can be found at the end of [DIAMETER] or at the end of this draft.

Currently there exist approximately 5 implementations from different vendors. One of the implementations is even freely available as binary [DIA_SUN]. Furthermore an open source project has been started to create an open source implementation of the protocol [DIA_OS].

1.2. Diameter Deployment and Evolution

The Diameter protocol is already commercially available but not widely used yet because it is not standardized. Also most ISPs will not switch from RADIUS to Diameter for their dial-up services. However if Diameter is used in Release 5 of 3GPP as planned it is expected that the protocol will be widely deployed as part of the 3GPP rollout. A further driver will be ISPs offering mobile IP support. Currently a standardized RADIUS AAA solution for mobile IP does not exist.

The further activities of the AAA WG after standardization of Diameter base protocol and base applications are under discussion. The most likely scenario (advocates opinion) is that the AAA WG will continue to develop more Diameter applications (collaborating with other WGs if needed). Also the base protocol will probably be developed further in case the demand is there.

2. Architectural Considerations

This section introduces the architecture of the Diameter protocol and suggests a way of applying it to the communication between an IPFIX exporting process and an IPFIX collecting process.

The Diameter architecture consists of three different entities: Diameter clients, Diameter servers and Diameter agents.

Diameter clients send requests for Authentication and/or Authorization to Diameter servers. Diameter clients also send

accounting information to Diameter servers. Diameter clients are for example Network Access Servers (NASs) or foreign and home agents (mobile IPv4).

Diameter servers perform authentication and authorization decisions based on Diameter clients requests and return answers to Diameter clients. Diameter servers configure accounting of the Diameter clients and get accounting information send by them. When Diameter servers get messages not destinated to themselves they forward both Authentication and Authorization request and accounting data to the appropriate servers. Diameter servers may also automatically direct the Diameter clients to send accounting information in a particular way.

Four different types of Diameter agents have been identified and are described in Section 2.8 of [DIAMETER]. Diameter agents are used for a number of purposes: grouping of systems (security associations), request concentration, load balancing and value-added processing of requests/responses.

Please note that despite the fact that the terms client and server are used the Diameter protocol is not a client server protocol but a peer-to-peer protocol. Any Diameter peer can start a messages exchange. This will be described in more detail in the next section.

2.1. Diameter Protocol Overview

This section only provides a brief overview of the Diameter base protocol. Therefore it is mandatory to read the base protocol draft [DIAMETER] before evaluating the protocol.

The Diameter base protocol is intended to provide an AAA framework for applications such as network access or mobile IP [DIAMETER]. is a peer-to-peer protocol allowing any peer to start a message exchange.

The Diameter base protocol is never used on its own. It is always extended for a particular application e.g. mobile IPv4. The Diameter protocol is run on top of TCP [TCP] or SCTP [SCTP] which guarantee a reliable and congestion aware transport. Draft [AAA_TRANS] discusses the AAA transport issues in detail.

Diameter has a build-in watchdog algorithm which can pro-actively detect transport failures. Also failback and failover procedures have been defined (section 5.5.4 [DIAMETER] and [AAA_TRANS]).

Diameter supports dynamic peer discovery through different mechanisms e.g. DNS, SLPv2 [SLP] to enable a simpler and more robust deployment. Peers can also be configured manually.

Diameter provides application layer sessions which abstracts from transport connections. Diameter messages for multiple sessions are multiplexed through a single transport connection.

Diameter server and agents route messages to their final destination based on realms.

The data model of Diameter is based on Attribute Value Pairs (AVPs). Each Diameter message consists of a fixed header and a number of AVPs carrying data. The fixed header contains a 16 bit command code which identifies the type of message. A number of data types for AVPs have been defined already. AVPs can be grouped into logical structures. Diameter can be extended by defining new AVP values, new AVPs and new applications (which includes definition of new command codes).

The Diameter protocol supports capability negotiation between peers. This enables a more simpler and robust deployment.

Diameter supports applications which support Authentication, Authorization and Accounting. It also supports applications which only make use of accounting (see section 8 [DIAMETER]).

Diameter uses hop-by-hop security [DIAMETER] and a Diameter extension exists which supports end-to-end security [DIAM_CMS] (see security considerations for more details).

2.2. General Applicability

The Diameter architecture consists of three different entities (client, server and agent) while the IPFIX architecture is comprised of: observation point, metering process, exporting process and collecting process. From the viewpoint of the IPFIX protocol only the exporting process and collection process are relevant because these are the entities communicating via the IPFIX protocol. However the IPFIX protocol must carry data generated by the metering process.

The process of exporting IP flow information is similar to the accounting part of Diameter: The Diameter client sending accounting information is simlar to the IPFIX exporting process sending IP flow information. The Diameter server receiving accounting information is similar to the collecting process receiving IP flow information. Diameter Relay and Proxy Agents are similar to an IPFIX proxy (a

combination of collecting and exporting process) as described in section 9 [IPFIX]. The Diameter Translation Agent is similar to the Protocol Converter as described in section 9 [IPFIX]. The metering process and observation point are similar to entities co-located with the Diameter client which collect the accounting information.

The IPFIX protocol could be implemented with Diameter similar as a new Diameter accounting application can be defined. [DIAMETER] explicitly describes how new accounting applications can be defined in section 1.2.4.

2.3. Architectural Differences

The Diameter architecture has been developed for providing a AAA framework while IPFIX is to be developed for exporting IP flow information. There are no major differences between both architectures. The Diameter architecture is more generic than IPFIX. It supports a number of different proxy types and message routing. Furthermore the functionality of the first two As is not useful for IPFIX. However IPFIX could be viewed as a subset of Diameter similar to the Diameter accounting functionality. The Diameter standard explicitly supports accounting-only Diameter applications.

3. Item Level Compliance Evaluation

This section evaluates the compliance of the Diameter protocol with the IPFIX requirements item by item. Requirements are addressed by their section numbers and item numbers in [IPFIX-REQ]. For each requirement it is explained to what degree protocol Diameter meets the requirement and how this is achieved. The degree of compliancy is explicitly stated using five grades:

-T Total compliance: The requirement is met completely by the protocol specification without any extensions required.

-E Extension required for total compliance: The protocol is prepared to be extended and it is possible to extend it in a way that it meets the requirement. This grade is onLY applicable to protocols that are explicitly open to externally defined extensions, such as SNMP is extended by MIB modules or DIAMETER is extended by application modules. It is not applicable to protocols, where the protocol specification itself needs to be extended in order to comply with the requirement.

-P Partial compliance: The requirement is met partially by the protocol specification.

-U Upcoming compliance: The requirement is not met or met partially by the protocol specification, but there is a concrete plan for an upcoming version of the protocol.

-F Failed compliance: The requirement is not met by the protocol specification.

Requirement 4 (Distinguishing Flows): -E

The following requirements assumes that Diameter is not only used for exporting IP flow information but also for configuration of the export. In case a different protocol is used for the configuration these requirements do not apply.

As Diameter has not been developed for the purpose of IP flow export most of the attributes are not yet defined in the existing data model. To comply to the IPFIX protocol there are several possible options. Either all of the attributes are defined as new AVPs from the defined AVP types (and grouped where appropriate) (see Section 4.3,4.4 [DIAMETER]). Alternatively the existing QoS/IPFilterRule data types can be used and grouped with additional AVPs containing the attributes not yet defined in Diameter (see Section 4.4 [DIAMETER]). A further option is to extend the QoS/IPFilterRule types directly because most of the required attributes are covered already.

Requirement 4.1 (Interfaces): -E

Incoming interface and outgoing interface can be defined as AVPs.

Requirement 4.2 (IP Header): -E

Source IP and destination IP address can be defined as new AVPs. Note that Diameter supports both IPv4 and IPv6 addresses. Protocol Type and IP version number can be defined as AVPs.

The existing IPFilterRule data type supports source, destination IP, protocol type. It also supports prefix matches of the addresses via masks.

Requirement 4.3 (Transport Header): -T/E

Source and destination port number can be defined as AVPs. The

existing IPFilterRule data type supports ports, list of ports or ranges.

Requirement 4.4 (MPLS): -E

The MPLs label can be defined as AVP.

Requirement 4.5 (DSCP): -T/E

The Diffserv Code Point can be defined AVP. The existing QoSFilterRule supports a DSCP attribute.

Requirement 4.6 (Header Compression): n/a

This imposes no further requirements on the protocol

Requirement 5.1 (Reliability): n/a

This requirement does not apply to the protocol.

Requirement 5.2 (Sampling): -E

This requirement is a metering process requirement. However the IPFIX protocol must support to export some information about sampling configuration. New (grouped) AVPs can be defined for carrying the sampling configuration.

Requirement 5.3 (Overload Behavior): -E

This requirement is a metering process requirement. However some information must be send via the protocol e.g. to indicate overload behavior changes. New AVPs can be defined to signal changes of metering behaviour to the collecting process.

Requirement 5.4 (Timestamps): -E

This requirement is a metering process requirement. However a candidate protocol must support a proper timestamp format. Diameter does only support the Time data type which is UTC time in seconds. To meet the requirement a new AVP data type must be defined. Alternatively a Time AVP can be used for the seconds and grouped with an additional AVP for the centiseconds.

Requirement 5.5 (Time Synchronization): n/a

This requirement does not apply to the protocol.

Requirement 5.6 (Flow Expiration): n/a

This requirement does not apply to the protocol.

Requirement 5.7 (Multicast Flows): n/a

This requirement does not apply to the protocol.

Requirement 5.8 (Ignore Port Copy): n/a

This requirement does not apply to the protocol.

Requirement 6.1 (Information Model): -E

For most of the attributes required there currently exist no defined AVPs. But for all attributes listed AVPs can be easily derived from the base data types. Instead of using existing data types new data types could be defined.

Requirement 6.2 (Data Model): -T

The data model of Diameter is based on the Attribute Value Pair (AVP) concept. An attribute is identified uniquely by a numeric AVP code and AVP length. A number of base types for AVPs have been defined in [DIAMETER]. The data model of Diameter is extensible because new AVPs and new AVP types can be defined. Diameter supports grouping of AVPs and nesting of grouped AVPs to create more complex structure.

The data model is flexible because each Diameter message only has a small fixed size header. After the header arbitrary AVPs (as defined for a message) follow.

The data model is independent of the transport (TCP or SCTP are used).

Requirement 6.3.1 (Congestion Awareness): -T

Diameter uses TCP or SCTP as transport. Both protocols are congestion aware.

Requirement 6.3.2 (Reliability): -T

Diameter is an application layer protocol which uses TCP [TCP] or SCTP [SCTP] as transport protocols which are both reliable. To support application layer reliability the protocol supports application layer ACKs and error messages.

A watch dog mechanism has been defined to detect transport problems and failover and failback procedures have been defined. Diameter also supports capability negotiation between peers which assures that both peers have the same capabilities.

Requirement 6.3.3 (Security): -T

Diameter provides end-to-end as well as hop-by-hop authentication, integrity and encryption. Some mechanisms are provided by underlying security protocols used such as IPSec or TLS. Since [DIAMETER] specifies how to use them this requirement is considered to be met by the protocol. Please read the security considerations section for more details.

Requirement 6.3.4 (Push/Pull Mode): -T

Diameter is a peer to peer protocol. The current Diameter accounting model uses the push mode (a Diameter client sends accounting information to a Diameter server). However a Diameter application could be defined which supports a pull mode as well.

Requirement 6.4 (Regular Report Interval): -T

Diameter can send accounting information in regular intervals.

Requirement 6.5 (Event Notification): -T

A Diameter peer can send messages at times of events. The events and messages must be defined for the specific application. A Diameter configuration message could configure when to send specific event messages. Currently the Diameter base protocol sends accounting messages at the start and end of a session.

Requirement 6.6 (Anonymization): n/a

Diameter does not support anonymization. However anonymization is not a protocol specific function and therefore the requirement does not apply to the protocol. A function can be integrated into a Diameter peer which anonymizes certain data before it is exported in AVPs.

Requirement 7 (Metering Process Configuration): -E

These requirements only apply if Diameter is used for configuration of the metering process. Since Diameter is not a protocol for exporting IP flow information the listed attributes are not specified yet. Since the data model is flexible all attributes can

be specified as AVPs.

Requirement 8.1 (Openness): -T

The Diameter base protocol is open and has an extensibility concept specified in the standard. The flexible AVP model allows to support any information model. New Diameter applications can be created which can define application specific messages and message exchange.

Requirement 8.2 (Scalability): -T

A Diameter peer is not limited in the number of connections to other peers. In Diameter each peer has a unique identifier which must be present in each message (Origin-Host AVP). Furthermore Diameter uses session IDs to uniquely identify specific sessions.

Requirement 8.3 (Several Collecting Processes): -T

Diameter accounting records are usually only send to the home server. However there is no limitation in the protocol that restricts sending information to only one destination. Diameter supports duplicate data detection over multiple receivers because each accounting message contains client ID, session ID, timestamp and a sequence number in each message.

4. Security Considerations

Security considerations for the IPFIX protocol are covered by the comparison against the specific Security requirements in the IPFIX requirements document [IPFIX-REQ] where they are specifically addressed by sections 6.3.3 and 10.

The Diameter base protocol assumes that messages are secured by using either IPSec or TLS. This security model is acceptable in environments where there is no untrusted third party agents. The use of TLS, IPSEC and considerations of peer-to-peer security issues are discussed in the security considerations of [DIAMETER].

In situations of untrusted third party agents, end-to-end security is needed. [DIAM_CMS] describes how a security association is established by two peers through agents, and how authentication, integrity, confidentiality and data origin authentication are achieved using a mixture of symmetric and asymmetric transforms.

5. Acknowledgements

The authors would like to thank Jari Arkko for his valuable comments on the first version of the draft.

6. References

[IPFIX-REQ] J. Quittek et al., "Requirements for IP Flow Information Export", draft-ietf-ipfix-reqs-05.txt, work in progress, July 2002.

[DIAMETER] P. Calhoun et al. "Diameter Base Protocol", draft-ietf-aaa-diameter-12.txt, work in progress, July 2002.

[AAA_TRANS] B. Aboba at al., "Authentication, Authorization and Accounting (AAA) Transport Profile", draft-ietf-aaa-transport-07.txt, work in progress, April 2002

[DIAM_CMS] P. Calhoun et al., "Diameter CMS Security Application", draft-ietf-aaa-diameter-cms-sec-04.txt, work in progress, March 2002

[TCP] Postel, J., "Transmission Control Protocol", RFC 793, January 1981.

[SCTP] R. Stewart et al., "Stream Control Transmission Protocol", RFC 2960. October 2000.

[SLP] E. Guttman, C. Perkins, J. Veizades, M. Day. "Service Location Protocol, Version 2", RFC 2165, June 1999.

[DIA_SUN] SUN Diameter implementation, http://playground.sun.com/diameter/

[DIA_OS] Diameter Open Source Project, http://sourceforge.net/projects/diameter/

7. Author's Addresses

Sebastian Zander
Fraunhofer Institute for Open Communication Systems (FOKUS)
Kaiserin-Augusta-Allee 31
10589 Berlin
Germany

Phone: +49 30 3463 7287
Email: zander@fokus.fhg.de

8. Full Copyright Statement