Diameter Single Sign On – Secure and Personalized Service Provision via Authentication and Authorization Mechanisms

Robayet Nasim

Faculty of Science, Engineering and Technology,
University of Science and Technology Chittagong, Chittagong
robayet@kth.se

Abstract. Network Services universally rely upon Authentication and Authorization mechanisms to ensure secure and personalized service provision. Protocols, such as Diameter provides a reliable framework for efficient access control to network services utilized by network devices. This framework can also encompass application level services e.g. web applications accessed via web browsers [1]. On the other hand, the prevalence of Internet based services and applications have brought about the burden of identity management among distributed security domains, an issue not specifically addressed by protocols such as Diameter. Efforts such as OpenID alleviate this difficulty by proposing an application level framework based on open standards to realize single sign on/off [2] semantics with regard to application level services. However, these technologies do not build upon existing security infrastructure, require significant investment in terms of technology adoption and have yet to receive industry wide acceptance and support. This paper presents Diameter Single Sign On – a framework that provides single sign on/off semantics in the context of network and application level services by harnessing the strengths of existing and proven authentication and authorization infrastructure. Because of combination of the Diameter protocol with Single Sign On and OpenID the proposed architecture overcomes the problem of identity management and also builds on existing security infrastructure.

Keywords: Diameter, OpenID, Authentication, Authorization, Single Sign On.

1 Introduction

As Internet becomes a popular medium of doing everyday works, ensuring security to the users to access a network or application level resources is a raising question in the modern world. Furthermore, providing privileges to the users for accessing resources as well as storing users history about their usage of the resources is also taken the concern. AAA (Authentication, Authorization, Accounting) security services provide the primary framework through which a network administrator or a service provider can set up access control on network points of entry or network access servers, or set up an access control on applications [10]. Authentication checks the identity of the

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 573–581, 2012.

users, Authorization confirms the proper access control rights of the users, and Accounting declares the history of the users [11]. The core elements [11] of AAA are: *Clients* for authentication (itself or another user); *Policy Enforcement Point* (Authenticator), provides the constraints for a client access to the resources; *Policy Information Point* (PIP), stores information about devices or user access requests and helps to make the access decision; *Policy Decision Point* (AAA Server), takes final decision about a resource access and takes the access request from the clients through PEP and queries for relevant information to the PIPs for decision making; *Accounting and Reporting System,* records usage of the resources with details information, such as – who are using the resources now, from where the resources are accessed, who are granted to access the resources, etc.

In recent times each and every user has a large number of accounts for using different types of web applications or network resources. It is quite natural that a single user may not have the same user ID in all of these accounts and that's why, it is difficult for that user to manage all of his IDs (combination of both user name and password). To solve this identity management problem an open decentralized, true framework is introduced named OpenID, [12] which offers the way of authentication to a user for several services by providing his OpenID password only once.

Diameter [3] is a network protocol that provides AAA to the end users. Because of the flexibility of this protocol it can be used efficiently for the basic purpose of the AAA realm. Although this protocol has reliable framework, efficient access control and support for the application level services, it suffers from lack of identity management capabilities. However, Single sign On [6] with OpenID solve this problem of identity management by authenticate users by providing their passwords only once. But it does not build upon existing security infrastructure and requires significant investment in terms of technology adoption.

Motivated by identifying these problems I proposed a framework *Diameter Single Sign On* for secure and personalized service provision via authentication and authorization in this paper. It provides single sign on/off semantics in the context of network and application level services by harnessing the strengths of existing and proven authentication and authorization infrastructure.

The remainder of the paper is organized as follows: In Section 2, I present the related work. Section 3 represents the detailed description of my solution and Section 4 illustrates possible future research directions. Finally, section 5 presents the conclusions of this paper by including the achievements.

2 Related Work

2.1 Network Service Authentication and Authorization with Diameter (RADIUS X 2)

The Diameter [3] is a network protocol for centralized Authentication, Authorization and Accounting. It is an application layer protocol that handles the communication between clients and servers through reliable transport. It is an upgrade to the RADIUS [4] (Remote Authentication Dial in User Service) protocol that is in wide use for access control to various network services including local area networks, wireless networks and the Internet. Fig. 1 illustrates a typical Diameter deployment scenario.

In a typical Authorization and Authentication transaction [5], the following sequence of events takes place.

1. User via User Agent (UA) initiates authentication procedure with the Network Access Server (NAS) to access a network service.
2. NAS Diameter client forwards Access-Request with user credentials to the Diameter Server.
3. Diameter Server checks the authenticity of user credentials and responds with Accept or Reject.
4. Based on the type of response and associated attribute value pairs, the Diameter client provides appropriate services to the User.

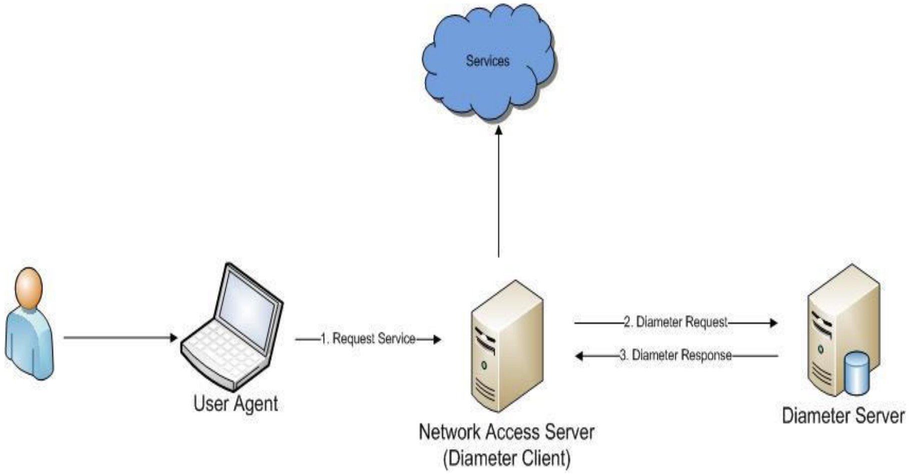


Fig. 1. Network Service Authentication & Authorization with Diameter

Although Diameter provides an efficient access control to network and application level services, the issue of identity management in Internet-based applications is not addressed properly.

2.2 Single Sign On with Open ID

Single sign on [6] refers to the facility that allows a user to a number of application services by providing his/her credentials only once. OpenID realizes Single Sign On / Off [7] by allowing a user to authenticate with multiple security domains using a single identity such that the security domains trust certain authentication authority. Fig. 2 illustrates a higher level view of Single Sign On with OpenID.

After the user has established a trust relationship with an OpenID Identity Provider / Server (e.g.*provider.com* by registering an OpenID Identifier (an unique URL e.g. *bob.provider. com*), the following sequence of steps takes place while signing in.

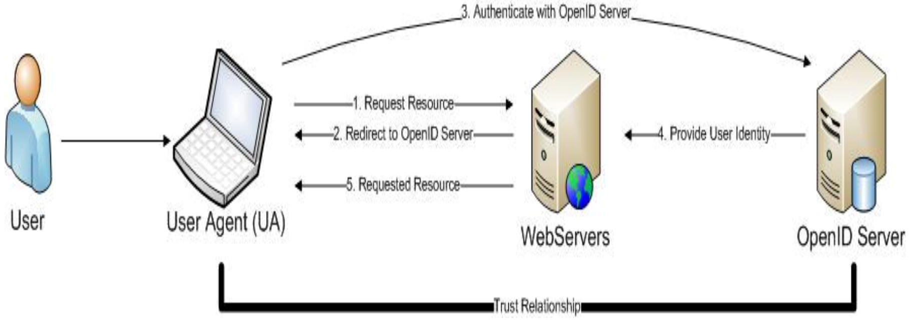


Fig. 2. Single Sign On with Open ID

1. UA requests a resource hosted by a relying party Web Server and presents an OpenID identifier.
2. The Web Server determines the OpenID Identity Provider from the presented Open ID Identifier and determines its Service URL. The Web Server and the Identity Provider establish a shared secret. The Web Server then redirects UA to the Identity Provider.
3. The User via UA provides authentication credentials to the Identity Provider.
4. Having successfully authenticated the user, the Identify Provider forwards users identity credentials to the Web Server with the user's consent.
5. The Web Server verifies the authenticity and integrity of received identity information using the shared secret established earlier. Upon successful verification, the requested resource is presented to the user.

However, Open ID does not build on existing security infrastructure and therefore, requires an industry wide accepted standard infrastructure.

3 Solution

Single Sign On Service that allows manageable and end user friendly access control for network and application level services can benefit from the reliability and mass deployment of Diameter / RADIUS infrastructure. In the proposed framework, end user security credentials and associated information including preferences as well as access rights are stored in a data store (LDAP, RDBMS) connected to the Diameter Server. Each user is uniquely identified by an identifier of the form *User@SecurityDomain*, where SecurityDomain represents the globally unique identifier (URI) of the Diameter Server and User represents a unique user name within the security domain. Thus, a trust relationship is established between the user and the Diameter Server. Fig. 3 depicts an envisioned deployment of my proposed architecture. Among other benefits, the proposed architecture provides a uniform mechanism for gaining access to both network and application level services. Below, I detail the flow of events associated with access control of network and application level resources.

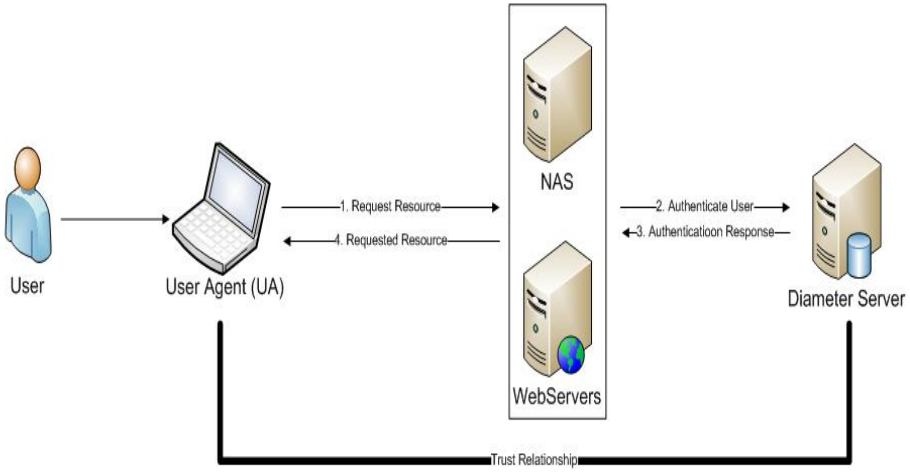


Fig. 3. Diameter Single Sign On

3.1 Network Service Single Sign on

1. User via UA communicates with the NAS to access a network service.
2. Diameter client on NAS initiates a Challenge Handshake Authentication protocol with the Diameter Server on behalf of UA.
3. Depending upon result of authentication process, the Diameter Server responds with Accept or Reject.
4. Based on the type of response and associated attribute value pairs, the Diameter client provides appropriate services to the User.

3.2 Application Service Single Sign On

1. User via UA requests a resource hosted by a Web Server and presents his/her unique identifier.
2. The Web Server determines the Diameter Server from the suffix of the presented Identifier and determines its Service URL and initiates a Challenge Handshake Authentication Protocol with the Diameter Server on behalf of UA.
3. Having successfully authenticated the user, the Diameter Server forwards user's identity credentials to the Web Server.
4. The Web Server presents the requested resource to the user.

Furthermore, I define two modes of Single Sign On for the Diameter Server i.e. Active and Passive. In active mode, the Diameter Server actively signs in the user to a set of previously specified services. In this case, the user may access all services from the set by only specifying user's unique identifier to the Service Provider. In passive mode, the Diameter Server provides the user's identity information to a Service provider only when the user chooses to request for the service. In both cases,

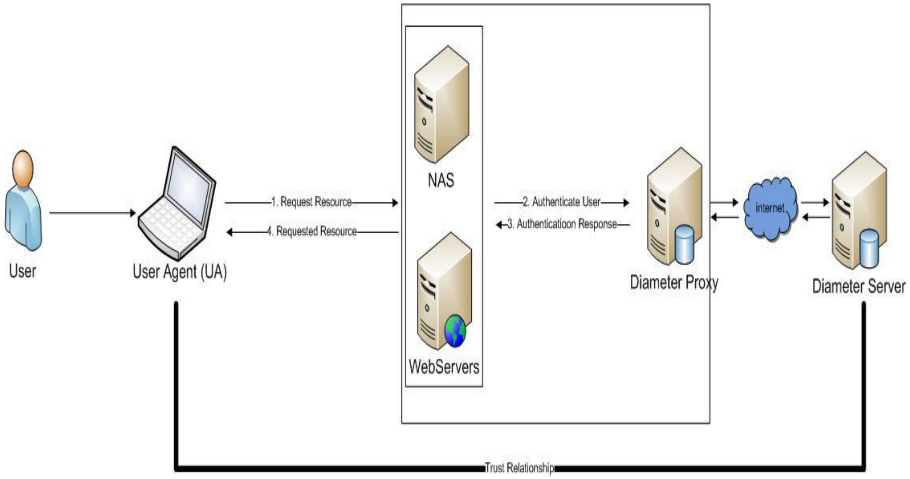


Fig. 4. Diameter Single Sign On Roaming

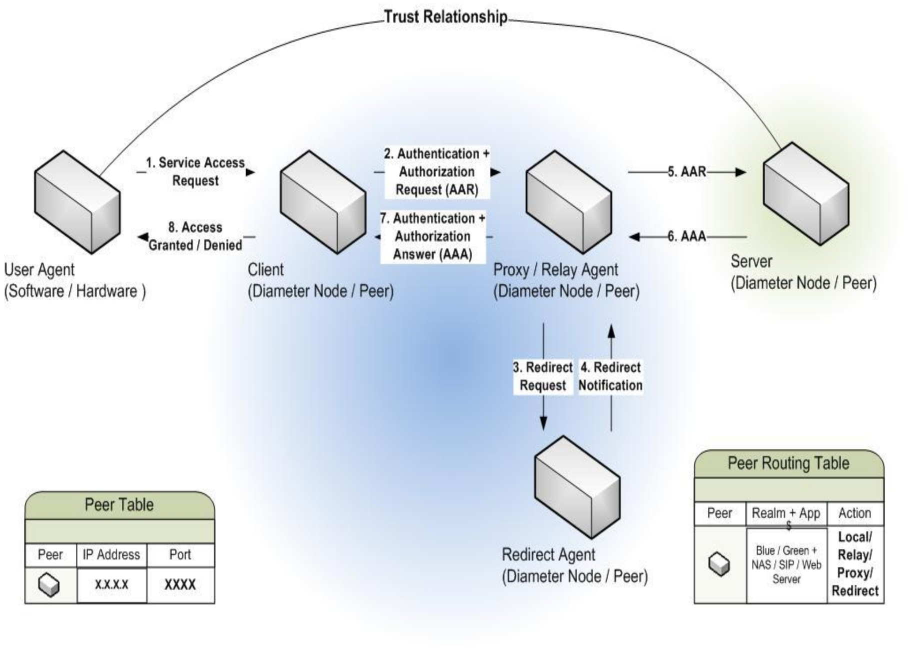
the Diameter Server and Service providers must ensure that only the authenticated user may access the available services by reliable and appropriate techniques such as checking for IP Address or Certificates.

3.3 Architecture

The Diameter provides a rich framework sufficient to implement my proposed architecture. Being a Peer-to-Peer architecture [8], a Diameter node can create, send, forward, modify and receive Diameter protocol messages. It is of interest for the purpose of this text to signify the following Diameter node / agent types and concepts.

- A *Relay Agent* forwards messages.
- A *Proxy Agent* forwards and optionally modifies messages.
- A *Redirect Agent* assists with routing messages.
- A *Peer Table* is kept at every Diameter node and lists particulars such as address and capabilities of all known Diameter nodes.
- A *Peer Routing Table* is kept at every Diameter node and specifies the correct processing for a received message, that is, forward, modify, redirect or process locally.

I illustrate, in Fig. 5, a basic yet comprehensive scenario that depicts the feasibility of the Diameter framework for implementing my proposed architecture. I aggregate the various possible User Agent entities into one entity specified as *User Agent* which could be software or a device. Similarly, various Diameter clients providing a variety of services have been collectively represented as *Client.* The scenario depicts the basic interaction between a User Agent and a Service Provider, whereby the User Agent is interested in a privileged resource available at the Service Provider. The Service Provider ensures authenticated and authorized access to the resource by taking on the role of a Diameter Client node as outlined below.


Trust Relationship
1. Service Access Request
8. Access Granted / Denied
2. Authentication + Authorization Request (AAR)
7. Authentication + Authorization Answer (AAA)
5. AAR
6. AAA
3. Redirect Request
4. Redirect Notification
User Agent (Software / Hardware)
Client (Diameter Node / Peer)
Proxy / Relay Agent (Diameter Node / Peer)
Server (Diameter Node / Peer)
Redirect Agent (Diameter Node / Peer)

Peer Table

Peer	IP Address	Port
⬡	X.X.X.X	XXXX

Peer Routing Table

Peer	Realm + App	Action
⬡	Blue / Green + NAS / SIP / Web Server	Local/ Relay/ Proxy/ Redirect

Fig. 5. Diameter Interaction Scenario

1. User Agent (UA) requests access to a service *S* along with its unique identifier *UID* (and security credentials encrypted with server's Public Key if PKI is used).
2. Client creates an Authentication and Authorization Request (AAR) on behalf of the UA and sends the AAR to the Diameter node indicated by the Peer Routing Table for the Realm and Application indicated by the requested service *S* and User Agent's *UID*.
3. The Next Hop of the AAR, in this case a Proxy / Relay Agent, might determine that the AAR is to be forwarded to another realm and that it requires routing information from a Redirect Agent to accurately forward the message.
4. The Redirect Agent would return redirection information sufficient for accurate routing of the AAR.
5. Having received the necessary routing information, the AAR would be relayed to the appropriate Diameter Server.
6. After having recognized that the message should be processed locally, the Server node would return the appropriate Authentication and Authorization Answer (AAA), based on the received AAR, to the Relay/ Proxy agent.
7. The Proxy / Relay Agent would then forward the AAR to the Client.
8. The Client would inspect the AAR and grant or deny access to service accordingly.

The sequence of steps outlined above presents a bare bones interaction among participating entities and a number of interactions involving connection and session set-up, capabilities negotiations and session tear down have been omitted for sake of simplicity.

3.2 Security Considerations

Diameter Base Protocol [9] mandates support for TLS and IPsec at Diameter Servers and IPsec at all Diameter nodes thereby ensuring ample on the wire security for Diameter protocol messages. Though the security mechanism (CHAP and PAP) provided by Diameter Base Protocol for securing user credentials are sufficient, usage of Public Key Infrastructure is recommended in order to reduce the complexity of interactions among User Agent and various Diameter nodes.

4 Future Work

The framework presented in this paper is the first step to build an Authentication and Authorization framework to provide single Sign On/off for network and application level services. Therefore, as a future work, I plan to focus on a detailed evaluation of the proposed architecture against different security threats to compare robustness of the architecture against contemporary schemes.

5 Conclusions

The demand for an Authentication and Authorization framework that leverages the capabilities of exiting AAA infrastructure to provide single sign on/off for network and application level services has been widely felt. The Diameter protocol provides ample support to serve as the foundation for a candidate architecture that meets these criteria. The proposed architecture benefits from the elements of the Diameter framework including Diameter Agents, Protocol Messages as well as Security Mechanisms.

References

1. Neumann, N., Fu, X.: Diameter WebAuth: An AAA-based Identity Management Framework for Web Applications. In: 51th Annual IEEE Global Telecommunications Conference, Computer and Communications Network Security Symposium, New Orleans, LA, USA, pp. 86–88. IEEE Press, New York (2008)
2. Build and Implement a single sign-on solution, http://www.ibm.com/developerworks/web/library/wa-singlesign
3. Mehta, N.: Introduction to Diameter protocol, http://blogs.oracle.com/naman/entry/introduction_to_diameter_protocol
4. Remote Authentication Dial in User Services, http://tools.ietf.org/html/rfc2865

5. How Does RADIUS Work?, http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a00800945cc.shtml
6. Introduction to Single Sign-On, http://www.opengroup.org/security/sso/sso_intro.htm
7. Eldon, E.: Single sign-On service OpenID getting more usage, http://venturebeat.com/2009/04/14/single-sign-on-service-openid-getting-more-usage/
8. Liu, J., Jiang, S., Lin, H.: Introduction to Diameter, http://www.ibm.com/developerworks/wireless/library/wi-diameter/
9. Diameter Base Protocol, http://tools.ietf.org/html/rfc3588
10. Authentication, Authorization, and Accounting, http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/aaans_ov.pdf
11. Network Authentication, Authorization, and Accounting: Part One-The internet Protocol Journal 10(1), http://www-fr.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-1/101_aaa-part1.html
12. What is OpenID?, http://openid.net/get-an-openid/what-is-openid