ISSN:0975-9646

Two Way Authentication For Android Client Against Active Directory

Prof. Sunita M. Dol, Varsha Kulkarni, Vipula Moholkar

Walchand Institute of Technology, Solapur,
Maharashtra, India

Abstract: There are several systems for dealing with two way mobile authentication which may differ in delivering the password to the authorized user or a different entity based on the security constraints. One of the disadvantage of the one factor authentication systems is the ease with which they can be cracked. This threat has increased over time as new and modern methods can be used for guessing and cracking authentication systems. The need for better and more secure systems has given rise to the concept of the two factor authentication system. In system mentioned here, two factor authentication for android client is provided. Android client is accessing different resources such as users files in a network. One factor is username and password which user gets when they become user of active directory and the second factor is OTP which is generated at client side and encrypted using RSA algorithm and at server side it is decrypted using same algorithm. OAuth is used to authorize the android client and providing it limited access in network. Client gets access token and only by using that client can access user's resources.

Key words: OTP (one time password), RSA algorithm, OAuth, Android

1. INTRODUCTION

One of the major shortfalls of the one factor authentication systems is the ease with which they can be cracked. This threat has increased over time as new and modern methods can be used for guessing and cracking authentication systems. The need for better and more secure systems has given rise to the concept of the two factor authentication system. Sometimes third party application need to access user resources on behalf of user. At that time client credentials such as username and password need to shared with third party app .To avoid this, OAuth is used.

Two factor authentication is becoming need for security. In this system, we are providing two factor authentication for android client. Android client is accessing different resources such as users files in a network. One factor is username and password which user gets when they become user of active directory and the second factor is OTP which is generated at client side and encrypted using RSA algorithm and at server side it is decrypted using same algorithm. OAuth is used to authorize the android client and providing it limited access in network. Client gets access token and only by using that client can access user's resources.

2.SYSTEM ARCHITECTURE

Fig. 1 shows the overall system architecture. At client side OTP is generated then client is redirected to Server login page by OAuth client .Then client enters generated OTP ,username and password .Then OAuth server will check whether user is active directory user or not. If user is valid then client is authorized and client will get access token. Only by using access token client will access resources.

A sequence diagram shows object interactions arranged in time sequence. Figure 2 shows sequence diagram of project. Here client initiates the process by generating OTP. Then client makes authorize request to apigee proxy. The apigee proxy then redirects to server login page. Client then enters the username, password and OTP .Then server authenticates client and gives authorization code. Then client request for access token. Server returns access token and then client can access resources.

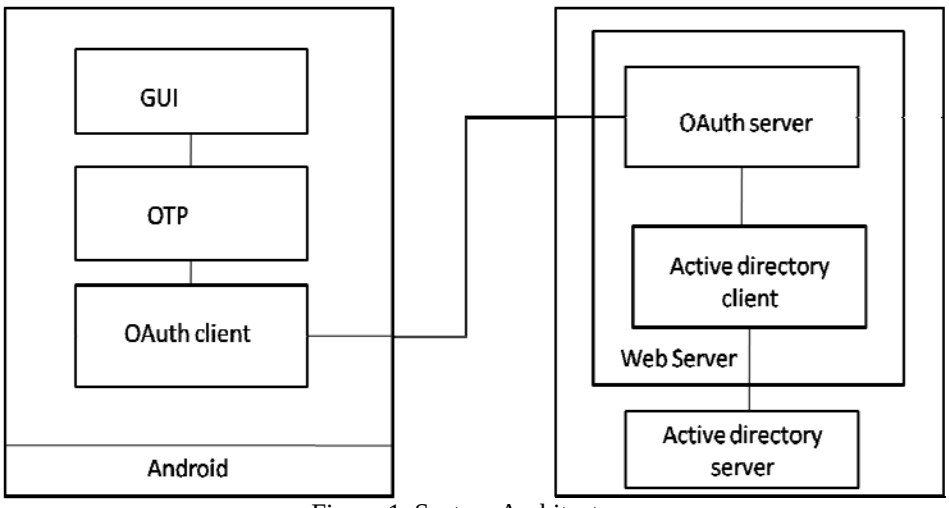

Figure 1. System Architecture

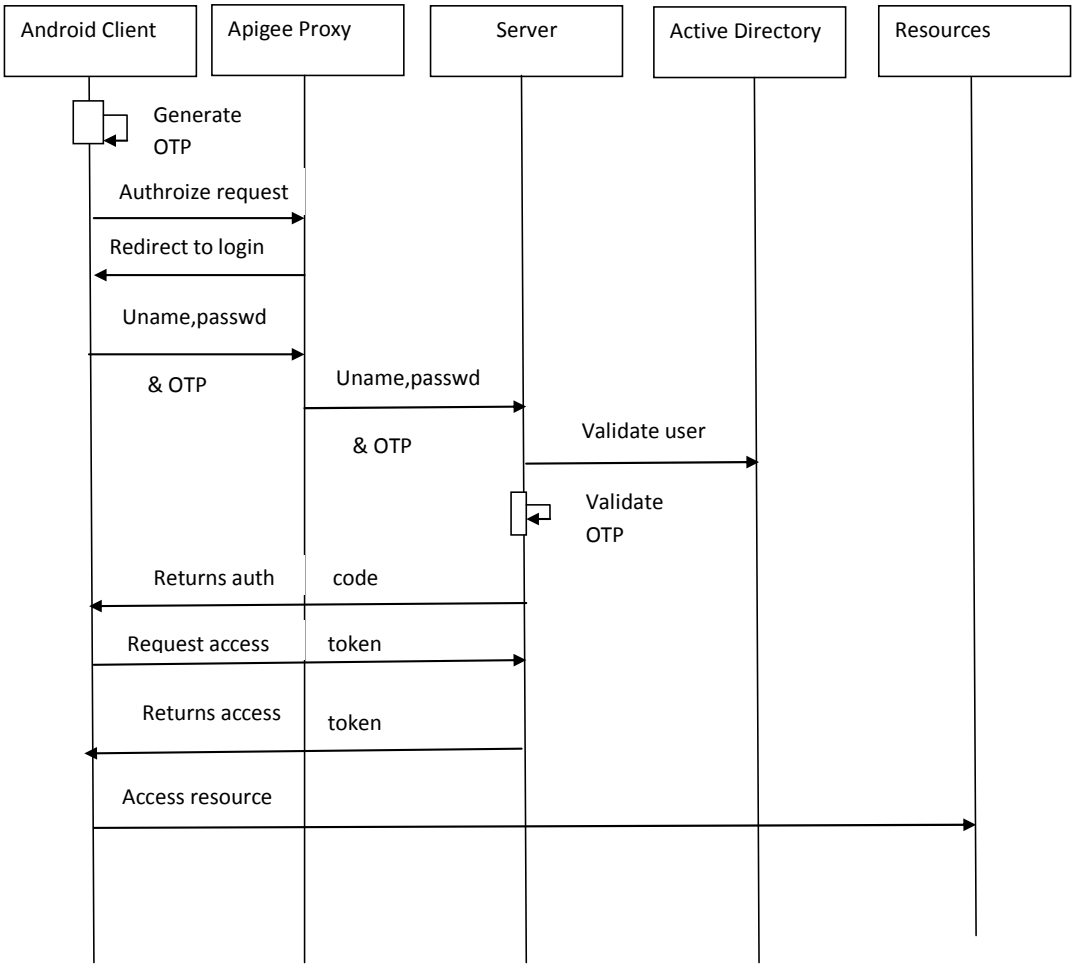

Figure 2. Sequence Diagram

3.TECHNOLOGY USED

3.1 Android

Android, as a system, is a Java-based operating system that runs on the Linux 2.6 kernel. The system is very lightweight and full featured. Android applications are developed using Java and can be ported rather easily to the new platform. Other features of Android include an accelerated 3-D graphics engine, database support powered by SQLite, and an integrated web browser. Third-party applications including those that are home grown are executed with the same system priority as those that are bundled with the core system.

3.2 Apigee Edge

Apigee Edge, which is built on Java, enables you to provide secure access to your services with a well-defined API that is consistent across all of your services, regardless of service implementation. A consistent API:

- ✓ Makes it easy for app developers to consume your services.
- ✓ Enables you to change the backend service implementation without affecting the public API.
- ✓ Enables you to take advantage of the analytics, monetization, developer portal, and other features built into Edge.

By creating an API proxy you let Edge handle the security and authorization tasks required to protect your services, as well as to analyze, monitor, and monetize those services. Because app developers make HTTP requests to an API proxy, rather than directly to your services, developers do not need to know anything about the implementation of your services.

4.SOFTWARE USED

4.1 Net Beans

NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others.The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The NetBeans IDE Bundle for Web & Java EE provides complete tools for all the latest Java EE 6 standards, including the new Java EE 6 Web Profile, Enterprise Java Beans (EJBs), servlets, Java Persistence API, web services, and annotations. NetBeans also supports the JSF 2.0 (Facelets), JavaServer Pages (JSP), Hibernate, Spring, and Struts frameworks, and the Java EE 5 and J2EE 1.4 platforms.

4.2 Active Directory

Active Directory (AD) is a directory service implemented by Microsoft for Windows domain networks. It is included in most Windows Server operating systems, as a set of processes and services.

An AD domain controller authenticates and authorizes all users and computers in a Windows domain type network—assigning and enforcing security policies for all computers and installing or updating software. For example, when a user logs into a computer that is part of a Windows domain, Active Directory checks the submitted password and determines whether the user is a system administrator or normal user.

Active Directory makes use of Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Microsoft's version of Kerberos, and DNS. The command used to install active directory is dcpromo. Once the directory is installed we can create active directory users and groups by opening active directory users and group wizard.

5.IMPLEMENTATION

There are three main module:

5.1 OAuth Proxy
5.2 Client module
5.3 Server module

5.1 OAuth Proxy

The OAuth proxy is the module which covers the work of OAuth in project. This module is implemented using Apigee Edge. There are four main parts of this module:

5.1.1 OAuthLoginApp Proxy :

OAuth login app proxy implements 3-legged oauth. In 3-legged oauth, first client request for authorization code. After receiving authorization code, client request for access token using authorization code. Then client gets the access token and expiry time. Main purpose of this proxy is to implement all above functionality of oauth. For this purpose following policy are added to this proxy :

- ✓ Generate Authorization Code : This is the OAuth2 policy that generates authorization code . This policy requires client_id , scope, and response type as parameter.
- ✓ Generate Access Token : This is also OAuth2 policy that generates access token. This policy requires code i.e. authorization code ,response type as parameter.
- ✓ Redirect to Login App : This is raise fault type policy that redirects the client to server's login page on request to authorize.
- ✓ Traffic management policies : The API Platform supports two traffic management policy which are SpikeArrest and Quota.

5.1.2 Creating product

The users in an organization create one or more API products, where an API product is a bundle of API proxies combined with a service plan. That service plan can set access limits on API proxies, provide security, allow monitoring and analytics, and provide additional features. While creating api product we need to specify access level , environment ,allowed oauth scope. We also need to specify api proxy and resources.

5.1.3 Creating Developer

An organization contains one or more developers who build the apps that consume the APIs (assembled into API products) defined by your organization. Developers consume APIs but cannot create APIs or perform any other actions in the organization.

5.1.4 Creating Developer Apps

Developers create one or more client apps that consume your APIs. For creating developer app we need to specify name of app, callback url ,developer. After creating developer app ,apigee provides consumer key and consumer secret. This consumer key is used while requesting for authorization code.

5.2 Client Module

Client module is android client. Android client first generate OTP and then request the apigee proxy to authorize client and get access token from server. Once the access token received then client can access resources. Following are the important parts in client module:

5.2.1 Generating OTP :

At client side OTP is generated using RSA public key encryption algorithm. The current system time is encrypted and from that OTP is generated. This OTP is displayed to user at starting as shown in following figure:

5.2.2 Redirect to Login App

After that client request to apigee proxy as follows:

https://varsha7-test.apigee.net/oauth/authorize?response_type=code&client_id=CS6MA6U4CZeE6AGfQ0dIpqPgALU4c6p&redirect_uri=http://192.168.56.101:8080/Demo/GetCode.jsp&scope=READ&state=foobar

5.2.3 Home Page

After successfully login, the server will authenticate the user and authorize the client. Server will return access token. When client gets access token only then it can access resource. Hence when client gets access token only then app redirect to home page. From home page user can download files from server, user can view existing file.

5.2.4 Download Files

After clicking Download Files button on home page, the location of File and name can be specified to save the downloaded file.

5.2.5 View Existing

After clicking view existing button on home page , view existing page displayed as shown below. This page shows all files which are already downloaded by user. It also categorize the file according to type.

5.2.6 Advance Search

Advance search provides two advance search options:

- Search by starting character
- Search by date

5.3 Server Module

Server module consist of working at server side. Main task of server are mentioned below :

✓ Authentication :

When client logins using username, password and OTP then server authenticates the user against active directory. That is server checks whether user is active directory user or not. Also here OTP send by user is decrypted and difference between current system time and time encrypted in OTP is calculated, if difference less than 60s and

username and password are correct, server will give access token to user.

- ✓ Providing Resource :

When client request resources i.e. in our case file server should give files. Server should also respond to advance search request.

6 . CONCLUSION

In this paper, how two factor authentication for android client against active directory have implemnetd is expalined. One Time Password as second factor for authentication is used. OTP is the cheapest and most efficient way to be used for second factor.

OAuth provides client applications a 'secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Apigee Edge makes it easy for user to start with oauth, by configuring and enforcing oauth using policies without requiring user to write any code.

REFERENCES

[1] Thesis on two way mobile authentication system by Harish Dinne and Kartik Mandava
[2] The OAuth 2.0 Authorization Framework : http://tools.ietf.org/html/draft-ietf-OAuth-v2-31
[3] The OAuth 2.0 Authorization Framework :Bearer Token Usage: http://tools.ietf.org/html.draft-ietf-OAuth-v2-bearer-22
[4] OAuth : http://en.wikipedia.org/wiki/OAuth
[5] Apigee Documentation : http://apigee.com/docs
[6] http://apigee.com/docs/api-services/content/what-apigee-edge
[7] Secure your api using OAuth 2.0 :Client Credentials http://apigee.com/docs/gateway-services/content/secure-calls-your-api-through-oauth-20-client-credentials
[8] Deploying proxies : http://apigee.com/docs/gateway-services/content/deploying-proxies-ui
[9] Active Directory : en.wikipedia.org/wiki/Active_Directory
[10] Android SDK developer.android.com/sdk