Towards Community Cloud Trustworthiness

By

Ahad Saad Alotaibi

B.A Alkharj University 2010

A thesis submitted to the graduate faculty of the

University of Colorado Colorado Springs

In partial fulfillment of the

Requirement for the degree of

Master of Science

Department of Computer Science

2016
This thesis for Master of Science degree by

Ahad Saad Alotaibi

Has been approved for

Department of Computer Science

By

______________________________

Dr. Edward Chow, Chair

______________________________

Dr. Jia Rao

______________________________

Dr. Jonathon Ventura

June 7th 2016

Date
Alotaibi, Ahad Saad (M.S, Computer science)

Towards Community Cloud Trustworthiness

Thesis directed by Associate Professor Edward Chow.

In this thesis we presented an authorization system for community cloud. We investigated the use of XACML rule for specifying the resource access control and explored the use of OpenAM for its implementation. We also investigated how the ranking information of the new community member can be computed based on the recommended rank values of the selected recommending community members with additional weights based on the ranks of the recommenders. We suggest the use of contribution effort and credibility information to revise the ranking periodically The ccAuth system was implemented as a web application in ASP.net connected to MySQL database to test the proposed design. For the performance we test our servers on the three different configurations: baseline, normal operation on a local server, and that with AWS cloud virtual machine at Oregon region. The results show the maximum time performance with the AWS cloud server is 365 milliseconds, well within 1 seconds, and should be considered as acceptable for most applications.
Acknowledgements

I want to thank My Family members for their support and picking me up when I was down and hopeless. I’m grateful that such positive people surround me.

I would love to thank Dr. Edward Chow for his supervision and support since day one. His encouragement and positive words were an important factor in my thesis. I do appreciate him as my advisor. Thank you so much.

Also, I would like to thank Dr. Jia Rao and Dr. Jonathon Ventura for being interesting in my thesis and providing me with useful feedbacks and alteration.
# Table of Content

## CHAPTER 1

**INTRODUCTION** ............................................................................................................... 1

1.2 Related Work .................................................................................................................... 3

1.3 Social Aspects and Trust ............................................................................................... 5

1.4 Proposed "ccAuth" ........................................................................................................ 7

## CHAPTER 2 ccAuth

**CONCEPTS & DEFINITIONS** .......................................................................................... 11

2.1 Role-Based Access Control (RBAC) Model ...................................................................... 11

2.2 eXtensible Access Control Markup Language (XACML) ............................................. 12

2.2.1 XACML Architecture ............................................................................................ 12

## CHAPTER 3

ccAuth DESIGN ....................................................................................................................... 14

3.1 System Architecture ....................................................................................................... 14

3.1.1 ccAuth web application Design .............................................................................. 15

A web page that allow the user to execute action on the data base such as insert or update. The web page is connected to the MySQL database mb_info to insert new member or update current one. Note this page has limited access only privilege users can execute the action .............................................................................................................. 16

3.1.2 Encoding Rules In XACML Format ....................................................................... 16
3.2.2 Algorithm calculate the Rank from recommenders...............................20
3.2.3 Rank Adjustment Based on Contributions and Credibility Over Time........21

CHAPTER 4
PERFORMANCE EVALUATION........................................................................23

CHAPTER 5
PROBLEMS AND LESSONS LEARNED .............................................................26

CHAPTER 6
IMPLEMENTATION ..........................................................................................30

CHAPTER 7
FUTURE WORK ..............................................................................................42

CHAPTER 8
CONCLUSION .................................................................................................43

CHAPTER 9
REFERENCES ..................................................................................................44

CHAPTER 10
APPENDIX A. INSTALLATION AND DEMO .....................................................47

10.1 Configuration of ASP.NET visual studio.......................................................47
10.2 Configuration of ASP.NET core .................................................................51
Table of Figures

Figure 1-1 On-site Community Cloud ................................................................. 2
Figure 1-2 WorkFlow for the rank system .......................................................... 8
Figure 1-3 Workflow for the access management system process for the rank system .... 9
Figure 3-1 ccAuth Home page .............................................................................. 14
Figure 3-2 ccAuth Main page ............................................................................... 15
Figure 3-3 tracker table ......................................................................................... 18
Figure 4.1 Performance of ccAuth system on Insert and Delete operations under three
different configurations. .................................................................................. 22
Table of Tables

Table 4-1 System Configurations .................................................................................................. 23
Table 4-2 Web application commands performance results ...................................................... 24
Table 4-3 Web application performance results without checking rank .............................. 24
Table 4-4 Web application command performance results with AWS network ............... 25
CHAPTER 1
INTRODUCTION

The need of the security services such as confidentiality, integrity, and availability is ever increasing and important nowadays due to the extensive usage of cloud computing. Four deployment models are summarized in NIST report [Liu2011] describing how the computing infrastructure that delivers these services can be shared: private cloud, community cloud, public cloud, and hybrid cloud.

A community cloud serves a group of Cloud Consumers, which have shared concerns such as mission objectives, security, privacy and compliance policy, rather than serving a single organization, as does a private cloud. Similar to private clouds, a community cloud may be managed by the organizations or by a third party, and may be implemented on customer premise (i.e. on-site community cloud) or outsourced to a hosting company (i.e. outsourced community cloud). Figure 1-1 (adapted from NIST [Liu2011]) depicts an on-site community cloud comprised of a number of participant organizations. A cloud consumer can access the local cloud resources, and also the resources of other participating organizations through the connections between the associated organizations.
Community clouds can be used as facilitate for the critical communication with the community members in time of disaster relief. Experiences have shown in many flooding and large scale fires that community clouds can be very useful in alleviating the concerns of citizens by providing effective notifications from government agencies or the emergency response organization, delivering urgent and timely services to the community members under stress.
1.2 Related Work

One of the challenges faced by the community cloud is to verify the identity and evaluate the trustworthiness of the community members based on the information available from other community members, and information registered in governments/local organizations.

Trust

Trust is a complex concept that has been researched from different aspects such as social and economics etc.

Mayer et al.’s model defined trust as” the willingness of a trustor to be vulnerable to the actions of a trustee based on the expectation that the trustee will perform a particular action “ [Maye1995]. Moreover, Rousseau et al defined trust as “a psychological state comprising the intentions to accept vulnerability based on positive expectations of the actions of the trustee “[Rous1998].

Gambetta et al [Gamb2000] defined trust as: “trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action”
Trustworthiness

In the definitions of trust, which provided by Mayer et al. (1995) and Rousseau et al. (1998) both have the expectation that another party will perform a specific action. One reason of that expectation is trustworthiness, as Lewis and Weigert (1985) noted:

First, trust is based on a cognitive process, which discriminates among persons, and institutions that are trustworthy, distrusted, and unknown. In this sense, we cognitively choose whom we will trust in which respects and under which circumstances, and we base the choice on what we take to be “good reasons,” constituting evidence of trustworthiness. (p. 970)

The different variations of trust definitions raise the issue of how it is possible to trust then authenticate a new individual into a community cloud based on the trust relationship among users. The question of trust among users in a group triggered more related questions:

- Is every user has unconditional access?
- Based on what criteria, will the recommender give unconditional access?
- Based on which criteria is the trust ranked?
1.3 Social Aspects and Trust

There are many social aspects that can be used to decide the trustworthiness of a community member, including

- Relationship between a recommender and the community member to be evaluated.
- Information registered in governments/local organizations.

There are various kinds of relationships including friendships, co-worker relationship and neighbor relationship. Each relationship has its nature and differences from the other ones as neighbor. Even two members with neighbor relationship could have conflicts and cannot get along and not willing to verify for the others. In this case conflicts can affect the ranks and to avoid that all the conflict relationships among community members will be recorded so that certain tasks can be organized and assigned to members with less or no conflicts.

Abdul Rahman et al [Abdu2000] have presented a trust supporting model in virtual communities where trust as a concept is divided into two types direct and recommender trust. In Abdual Rahmans model the *direct trust* has four agent-specified values (“very trustworthy”, “trustworthy”, ”untrustworthy”,” very untrustworthy”) and the recommender trust can be formed from word of mouth recommendation or previous experiences where it’s considered “reputation”.

Kamvr et al [Kamv2003] proposed The Eigen Trust reputation algorithm in P2P environment, where the system using the algorithm works on decreasing the number of inauthentic downloads is peer-to-peer file sharing environment. The algorithm donates a
global trust value for each user based on the user’s download history. The global trust value helps peers to choose from which other peer to download from based on the value of the global trust.

In the area of ranking whether a web page is a spam, [Gyon2004] present a technique where they select a subset of seed pages to be evaluated as reputable pages by an experts. When it’s evaluated they use the in common structure to differentiate the good pages from the spam.

Khalid Alkhattabi has designed and implemented a Community-Based Authentication and Authorization (CAA) for its users to pre-designate community group of members for authenticating each other. As soon as a server receives an access request from the user, it sends an SMS or utilizes Apple Push Notification service (APNs) to a pre-designated community group of members, requests them to authenticate that user and/or to approve critical access of valuable shared resources. The preliminary evaluation of the prototype shows that with the 60% threshold of 10 willing community members, the approval can reached within one minute [Alka2015]. In his current implementation, the community authentication rules are very simple, either all community member agrees before allow or any one member agree the applicant is allowed. It does not consider the trustworthiness-ranking. In this thesis we will focus on the specification of rules and their enforcement mechanism in CAA system and provide software tools for authorizing community resources, e.g., in disaster relief situations.
1.4 Proposed "ccAuth"

Trust ranking system is an evaluation system that based on ranks which is are giving by group of recommenders to a new community member, the rank number shows the level of trustworthiness and resources access for that new member. The rank is based on the nature of social aspects related to the new community member and recommenders.

A recommender is an existing community member that has a high level of trust rank in the system. The recommender mission is to evaluate and trust rank the trustworthiness of the new member based on social relationship or registered information in governmental recorders.

The process of ranking new community cloud member goes through stages:

1. The new member will register in the portal and add a recommender ID then send a request to the chosen recommender for joining the community.

2. The recommender, who receives the request, calls a subgroup of recommenders for voting.

3. The recommenders vote to rank the new member based on provided information. The vote will take into consideration of the trust rank of each recommender.

4. Recommendes will give the new member a trust access, which is a limited access to resources as a precaution.

5. The new member cannot make changes without permission for at least one of the high level recommenders.
6. For stronger security measure, a new member can be put under probation period when the activity of the member is recorded. Based on the results, recommenders can lift the probation condition and upgrade the members’ rank.

Figure 1-2 Workflow for the rank system
Community resource access management process will be enforced as follows:

1. The member registers for the first time then the system will setup the access management for the new member based on any updates on the rank status.
2. The member requests a specific resource.
3. The community portal will retrieve the member’s profile (trust ranks and history of access) and consult with the current access limitation and rules and decide whether to grant the access.
The results of the trust ranking system can be used to grant or deny accesses or resources and to select candidates for a specific community task, such as committee members for a local governing group. Where a community member that has a higher Trust rank will have more access to resources and can make different changes based on the trust rank, also the higher trust rank the more weighted in calculation for future member joining process.
CHAPTER 2

ccAuth: CONCEPTS & DEFINITIONS

2.1 Role-Based Access Control (RBAC) Model

The concept of roles has been around in software applications security area for over 30 Years. In recent years, role-based access control has emerged as a full-fledged model as mature as Mandatory Access Control (MAC) and Discretionary Access Control (DAC) [Samp 2002] [Sand1993]. RBAC model has matured to the point where it is prescribed as a generalized approach to access control [Ferr2001]. For example, RBAC was recognized to be "the most attracted solution for providing security features in multi-domain digital government infrastructure" [Josh2001].

One important feature of the RBAC model is that it simplifies the role management and security administration. As an example, if a user changes his/her position within the organization, then we can simply assign this user with the new role and the privileges of that user will change without use manually revoking the privileges, its time and effort saver.

In our approach we used RANK instead of ROLE. Each user will be assigned a rank, which is fixed number, and each rank is associated with certain privileges.
2.2 eXtensible Access Control Markup Language (XACML)

XACML is an XML-based OASIS standard that describes

- Policy language: describe the access control requirement.

- Request/response language:
  1. Create a request to ask if an action should be allowed or not.
  2. Translate the result as a response containing the decision as one of the four possible values: Permit, Deny, Indeterminate or Not Applicable.

2.2.1 XACML Architecture

![Figure 2-1 General XACML Architecture](image0.png)

Figure 2-1 General XACML Architecture
The general XACML-based architecture goes through the following steps:

1. Request is sent to the Policy Enforcement Point (PEP).

2. PEP creates an XACML request, which contains the Subject, Resource, and Action attributes then sends it to the Policy Decision Point (PDP).

3. PDP asks an attribute source (also called Policy Information Point (PIP)) to gather any additional attributes if necessary.

4. PDP receives the request, and then searches to find a policy that applies to the request.

5. PDP then creates an XACML response, sends it back to the PEP with the Decision.

6. PEP allows or denies access to the requested resource based on the response.

[OAS2004]
CHAPTER 3

ccAuth DESIGN

The basic functions of ccAuth application are:

- The member can request to be authenticate by a sub group of recommenders
- The system notifies the members when there is pending request.
- The system controls the member's access to different resource by enforcing a set of rules.
- The system has a ranking system to control the resource access, to increase the security level.
- The systems collects the member data and history to analyzed in the future in case of upgrade or enforce a new limitation on the member access.

3.1 System Architecture

Community Cloud Auth system consists of Community cloud portal that enable members to access their Control Panel. It consists of different tabs for different tasks to ease the access to different resources and tasks. The system also has a phpmyAdmin database that design specifically to the community.
3.1.1 ccAuth web application Design

Figure 3-1ccAuth Home page

Home Page:

A web page designed to the user log in. the web page will connect with MySQL database to mb_info to retrieve the user rank and mb_preivilge to return the user privilege.
Main Page:

A web page that allow the user to execute action on the data base such as insert or update. The web page is connected to the MySQL database mb_info to insert new member or update current one. Note this page has limited access only privilege users can execute the action.

3.1.2 Encoding Rules In XACML Format

To interrupter the ccAuth rules in to XACML I used two software. Eclipse IDE in conjunction with ALFA plugin, ALFA is a plugin software that translates pseudo code into an XACML language. I wrote my policy sets using Eclipse IDE as:

namespace ccAuthaccess{
    import Attributes.*

    policy ccauthSignIn{

apply deny Overrides

rule{

    target clause resourceType == "ccAuthWebApp"

deny

condition subjectRank==0 && subjectId=="null"

}

}
policy recoredAcess{

    // compare the user rank if its larger than the record classification allow it

    apply firstApplicable

    rule allowAccessIfClearanceSufficient{

        target clause resourceType == "data-record"

        condition subjectRank>resourceClassification

        permit

    }

}

}
policyset topLevel{

apply permit Overrides

acessrecordepolicy
databasacesspolicy

}
policy acessrecordepolicy {

}
target clause resourceType == "data-record"

apply denyOverrides

rule
{
permit

target clause subjectRank == 5 and actionId == "view" and actionId == "edit"
or subjectRank == 4 and actionId == "view" and actionId == "edit"
}

policy databasaccesspolicy {

target clause resourceType == "database-access"

apply permitOverrides

rule
{

    // users access limits on rank

    permit

    target

    clause subjectRank == 5 and actionId == "view" and actionId == "edit"

    or subjectRank == 4 and actionId == "view" and actionId == "edit"

    or subjectRank == 3 and actionId == "view"

    or subjectRank == 2 and actionId == "view"

}
or subjectRank == 1 and actionId=="view"
}
}
}
}

// users access limits on rank
permit
target
clause subjectRank == 5 and actionId=="view" and actionId=="edit"
or subjectRank == 4 and actionId=="view" and actionId=="edit"
or subjectRank == 3 and actionId=="view"
or subjectRank == 2 and actionId=="view"
or subjectRank == 1 and actionId=="view"
}
The above code describes the access limitation of ccAuth system. The user rank is the key of access for example above a user with a rank of “5” is conceded high level of trust worthiness, this rank allows the user to access to records, view it and modify it unlike user with “2” rank that is only allowed to view records.

3.2.2 Algorithm calculate the Rank from recommenders

Here is the algorithm for calculating the rank of a new member based on the recommendation of the selected community members. The recommended rank for the new member by the recommenders will be adjusted by the rank of the recommender. The average of the all recommending ranks will be used to derive the rank value for the new member. They can be adjusted later with the contribution effort and credibility based on the history record.

Here is an example how the average recommended value translated into the ran value.

If RecommendRank=85 less and equal 100
Then Rank=5
else If RecommendRank=75 less than 85
The Rank=4
else If RecommendRank=56 less than 75
Then Rank=3
Else If RecommendRank=35 less than 56
Then Rank=2
Else If RecommendRank=1 less than 35
Rank=1
3.2.3 Rank Adjustment Based on Contributions and Credibility Over Time

Here we propose a method for calculating the rank adjustment based on contribution to the community effort such as volunteer work, serving community committees. We envision the contribution effort for a period be translated into a fix value between 0 and some maximum value, say 5. They are recorded in the database with timestamp and userid. At the end of a period, the effort is then tallied for the aggregated value. The aggregated value is then normalized to a value between 0 and 1. The adjustment is then added to the current rank of the member.

The following shows a tracker_table which includes mb_id, timestamp, contribution value for each user and new rank that associated with value.

![Tracker Table](image)

Formula use to compute the rank adjustment:

\[
\text{rankAdjustment} = \frac{\sum_{d=1}^{n} \text{Contribution}_d}{\text{MaxContribution}} \quad (1)
\]
where Contribution\textsubscript{d} is the daily contribution by a member, n is the number of days in the period. MaxContribution is the max value a community member can contribute within the period. The rankAdjustment will then be added to the current rank. The value is capped at a maximum rank value say 10.

The value of rankAdjust will determine upgrading rank, if the user contribute 50% or more of the maximum contribution then we will award him/her with one point. It will read the old rank first then upgrade it to the next rank.
CHAPTER 4

PERFORMANCE EVALUATION

We collect the performance data of our ccAuth server on the three commonly used operations in three configurations. The baseline is the performance of the operations with database without the user ranking computation. The second one is the performance of the insert, delete, update operations with the user ranking checking. The third one is the performance with AWS cloud network overhead included. We use AWS region at Oregon to set up a server for the test.

The local server is setup with the following system configurations:

<table>
<thead>
<tr>
<th>Hardware Configuration</th>
<th>Sony Laptop with Intel i3-2350M CPU, 2.3GHz clock rate, 4GB RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows 10</td>
</tr>
<tr>
<td>XAMPP Server</td>
<td>XAMPP for Windows 5.6.21</td>
</tr>
<tr>
<td>Database Admin Package</td>
<td>phpMyAdmin 5.0.12</td>
</tr>
<tr>
<td>Server Side Scripting</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>App Development Env.</td>
<td>Microsoft Visual Studio 2015</td>
</tr>
</tbody>
</table>

Table 4-1 System Configurations

The AWS server is Windows 2012 R2 Server with t2.large size located in Oregon.

In our evaluation we considered time, to compute this factor we identified the current time and subtracted time of receiving request. Here is how time calculation is done in ASP.net.
This line computes the current time and pass it to a variable (now)

    DateTime now = DateTime.Now;

Here calculate the end time after sending the request and pass it to a variable (end)

    DateTime end = DateTime.Now;

We then calculate the differences between the two time variables and save it to TimeSpan ts variable.

    TimeSpan ts = end - now;

The print the result of the computation on the screen

    Response.Write("time span =" + ts.TotalMilliseconds + " millisec");

Table 4.2 shows the performance results of six requests for each of Inert, Delete and Update command in our ccAuth server with user rank checking.

Time unit is in milliseconds. Table 4.3 shows the baseline of database operation which does not include the user rank checking.

<table>
<thead>
<tr>
<th>Insert Op</th>
<th>Delete Op</th>
<th>Update Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>158.5864</td>
<td>195.9884</td>
<td>112.9836</td>
</tr>
<tr>
<td>253.0337</td>
<td>124.2062</td>
<td>84.803</td>
</tr>
<tr>
<td>196.014</td>
<td>101.0832</td>
<td>149.5409</td>
</tr>
<tr>
<td>125.9447</td>
<td>124.9109</td>
<td>107.9354</td>
</tr>
<tr>
<td>159.155</td>
<td>170.8617</td>
<td>111.2344</td>
</tr>
</tbody>
</table>

Table 4-2 Web application commands performance results

<table>
<thead>
<tr>
<th>Insert No-rank</th>
<th>Delete No-rank</th>
<th>Update No-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>211.117</td>
<td>187.9717</td>
<td>200.0589</td>
</tr>
<tr>
<td>147.4643</td>
<td>154.2903</td>
<td>207.2401</td>
</tr>
<tr>
<td>80.929</td>
<td>154.2903</td>
<td>212.9688</td>
</tr>
<tr>
<td>144.2064</td>
<td>103.0797</td>
<td>193.1231</td>
</tr>
<tr>
<td>131.0864</td>
<td>124.9463</td>
<td>212.6738</td>
</tr>
</tbody>
</table>

Table 4-3 Web application performance results without checking rank
Figure 4.1 shows the boxplots of the performance of ccAuth web application on Insert and Delete operations under the three different configurations.

![Boxplot Diagram]

**Figure 4-1 Performance of ccAuth system on Insert and Delete operations under three different configurations.**

<table>
<thead>
<tr>
<th>InsertAWS</th>
<th>DeleteAWS</th>
<th>UpdateAWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>270.7894</td>
<td>308.1914</td>
<td>225.1866</td>
</tr>
<tr>
<td>365.2277</td>
<td>236.4002</td>
<td>196.997</td>
</tr>
<tr>
<td>217.8483</td>
<td>273.3916</td>
<td>266.0187</td>
</tr>
<tr>
<td>308.7217</td>
<td>213.7912</td>
<td>262.2489</td>
</tr>
<tr>
<td>237.9987</td>
<td>236.9649</td>
<td>219.9894</td>
</tr>
<tr>
<td>271.2749</td>
<td>282.9817</td>
<td>223.3544</td>
</tr>
</tbody>
</table>

*Table 4-4 Web application command performance results with AWS network delay included*
CHAPTER 5

PROBLEMS AND LESSONS LEARNED

We faced some issues during the implementation of OpenAM. OpenAm is open source software that gives you a platform for testing and applying XCAML policies to different resources. Due to the fact that OpenAM is an open-source implies that there will be some bugs that may differ in different versions of the software.

1. When trying to name the policy in OpenAM (On the New Policy page) the name cannot have leading space. The program will not inform you in a clear message where the problem is and it is the nature of it.

2. As mentioned before different versions of Web Agent can have different command to apply. In the installation of Web Agent we faced an issue of using this line

   $ ./agentadmin --install --acceptLicense

   in Web Agent 4 the command above will not run and you will face an error message, instead you may replace it with

   $ ./agentadmin --i --acceptLicense
we did not face this issue in Web Agent 3. So you should take in consideration which version you will use.

3. we faced an issue of security limitation as Forbidden 403 web page

```
# amagent_auth_handler(): failed to get agent configuration instance, error: file parser error
```

we had to change the ownership to ease the access using this line

```
/opt/apache24_agent/bin/agentadmin --s
/usr/local/apache2/conf/httpd.conf $AM_SERVER_URL
$AGENT_URL / WebAgent /opt/pwd --acceptLicence --changeOwner
```

This step you may consider it before in hand, due to securities matters.

4. We tried to run apache on port 8000 was blocked. Here is the message in “journalctl –xe” log.
We solved this issue by executing the commands:

```
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
```

We faced issues with ASP.NET in connecting to the database through net connector. ASP.NET Visual basic environment did not recognize MySQL database so we tried using net connector to connect to our MySQL and faced problems setting the
software up. Even though we tried to remove net connector and re-setup the operating system did not run it, we believe the reason is the version of the operating system. Windows 10 home is fairly new and can make conflicts with other software. The solutions for this are either by having the connection string saved in a web.config or to add it in each page load. We used the second approach.

5. We faced a problem with importing the policies into OpenAM, after exporting an existed policy we found out that the policy from OpenAM has a different attributes and some differences in the policy editing. It was not wise to continue using OpenAM due to the consumed time in editing the policies.

6. OpenAM was not mature enough to connect to MySQL database for that we were not able to apply the policies.
CHAPTER 6

IMPLEMENTATION

To implement Community Cloud Auth system, I used:

• Windows 10 Operating System.
• The machine has CPU of Intel® Core ™ i3-2350M with a speed of 2.30 GHz, memory RAM 4 GB.
• phpMyAdmin 5.0.12.
• ASP.NET
• Microsoft Visual Studio
• Apache 2.4 with Open SSL
• Cento server
In Microsoft Visual Studio I used web application folder called it webappdemo which contains multiple folders and the web application forms:

- Homepage.aspx that includes
  - Homepage.aspx.cs where the code of the page saved

In Homepage we handle the log in of users into the portal also the page will save the user email and rank into a session variables.
protected void Button1_Click1(object sender, EventArgs e)
{
    string connectionstring = @"Data source=localhost; Database=ccauth1; User ID=root; Password='Com0Clo$";
    using (MySqlConnection cn = new MySqlConnection(connectionstring))
    {
        cn.Open();
        MySqlCommand cmd = cn.CreateCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select * from memebr_info where mb_email = " + t1.Text + " and mb_pass= " + t2.Text + ";//and mb_status='active';"
        cmd.ExecuteNonQuery();
        DataTable dt = new DataTable();
        MySqlDataAdapter da = new MySqlDataAdapter(cmd);
        da.Fill(dt);
        foreach (DataRow dr in dt.Rows)
        {
            Session["mb_email"] = dr["mb_email"].ToString();
        }
    }
}
Session["mb_Rank"] = dr["mb_Rank"].ToString();
cn.Close();
Response.Redirect("general.aspx");
//break;
}
}

The user will Enter Email and password into a text boxes. The log in button the will retrieve the user credential including the rank and save them to session procedure.

Session["mb_email"] = dr["mb_email"].ToString();

Session["mb_Rank"] = dr["mb_Rank"].ToString();

This will carry on into the next pages to confirm the identity of the user.

- In mainpage.aspx the user can connect and modify according to access limit associated with the user rank, user can add, delete and update.
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["mb_email"] == null)
    {
        Response.Redirect("Homepage.aspx");
    }
}

```
SELECT * FROM `membr_privilege`
```

<table>
<thead>
<tr>
<th>Rank</th>
<th>mb_privilege</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Insert</td>
</tr>
<tr>
<td>5</td>
<td>delete</td>
</tr>
<tr>
<td>5</td>
<td>view</td>
</tr>
<tr>
<td>2</td>
<td>view</td>
</tr>
</tbody>
</table>

CheckPrevilge() is procedure that takes to inputs from the membr_privilege table shown above. CheckPrevilge() is a Boolean method that checks logged in user access limits, by associate each privilege with a rank. The output of this procedure is allow the action done by the user or deny it.

private Boolean CheckPrevilge(int Rank, string Prevlige)
{
    Boolean Result = false;
    // Open prevlige table
    // search for the prevlige
// if exist return true , otherwise false

string connectionstring = @"Data source=localhost; Database=ccauth1; User ID=root; Password=";

using (MySqlConnection cn = new MySqlConnection(connectionstring))
{
    cn.Open();

    string sqls = "select * from membr_privilege where rank="+Rank.ToString()+" and mb_privilege="+Prevlige+"";

    MySqlCommand mycmd = new MySqlCommand(sqls, cn);
    MySqlDataReader myreader = mycmd.ExecuteReader();

    if (myreader.HasRows)
    {
        Result = true;
    }
}

    cn.Close();
}
return Result;
}

First the method calls CheckPrevilge() to check the user rank privileges if the user has a rank associate with this privilege "insert" the method will execute the insertion request, otherwise it will print a deny message.

protected void Button1_Click1(object sender, EventArgs e)
{

    int Rank = Convert.ToInt16(Session["mb_Rank"]);

    if (CheckPrevilge(Rank, "insert"))
    {
        string connectionstring = @"Data source=localhost; Database=ccauth1; User ID=root; Password="";

        using (MySqlConnection cn = new MySqlConnection(connectionstring))
        {
            cn.Open();

            string mb_surnm = t1.Text;
            string mb_lastnm = t2.Text;
        }
    }
}
string sqlstatment = "insert into memebr_info (mb_surnm,mb_lastnm,mb_email)
values(" + t1.Text + "," + t2.Text + "," + t3.Text + ");

MySqlCommand mycmd = new MySqlCommand(sqlstatment, cn);

mycmd.ExecuteNonQuery();
 cn.Close();

Response.Write("a member was created successfully");

As the previous action the method calls CheckPrevilge() to check the user rank privileges
if the user has a rank associate with this privilege "Delete" the method will execute the
delete request, otherwise it will print a deny message.
protected void Button2_Click(object sender, EventArgs e)
{

    int Rank = Convert.ToInt16(Session["mb_Rank"]);

    if (CheckPreivilge(Rank, "delete"))
    {

        string connectionstring = @"Data source=localhost; Database=ccauth1;
User ID=root; Password="";

        using (MySqlConnection cn = new MySqlConnection(connectionstring))
        {

            cn.Open();

            string mb_surnm = t1.Text;
            string mb_lastnm = t2.Text;
            string mb_email = t3.Text;


            string sqlstatment = "delete from memebr_info where mb_email ="" +
                                mb_email +"";

            MySqlCommand mycmd = new MySqlCommand(sqlstatment, cn);

            mycmd.ExecuteNonQuery();
        }
    }
}
Here the method calls CheckPrevilge() to check the user rank privileges if the user has a
rank associate with this privilege "update" the method will execute the update
request ,otherwise it will print a deny message.

    protected void Button3_Click(object sender, EventArgs e)
    {
        int Rank = Convert.ToInt16(Session["mb_Rank"]);
        DateTime now = DateTime.Now;
        if (CheckPrevilge(Rank, "update"))
string connectionstring = @"Data source=localhost; Database=ccauth1; User ID=root; Password="";

using (MySqlConnection cn = new MySqlConnection(connectionstring))
{
    cn.Open();
    string mb_surnm = t1.Text;
    string mb_lastnm = t2.Text;
    string mb_email = t3.Text;

    string sqlstatment = "update memebr_info set mb_surnm = ""+mb_surnm +"", mb_lastnm=""+mb_lastnm+"" where mb_email ="" + mb_email +"""";

    MySqlCommand mycmd = new MySqlCommand(sqlstatment, cn);

    mycmd.ExecuteNonQuery();
    cn.Close();

    Response.Write("a member was updated successfully");
}
}
else
{
    Response.Write("Sorry, you do not have the preivilge");
}

string connectionstring = @"Data source=localhost; Database=ccauth1; User ID=root; Password="; using (MySqlConnection cn = new MySqlConnection(connectionstring))
{cn.Open(); cn.CLOSE();}

int Rank = Convert.ToInt16(Session["mb_Rank"]); if (CheckPrevilge(Rank, "update")

Connect to MySQL by calling the data source, the database name and user Id
Call CheckPrevilge procedure to Check with mb_privilge table, in database, to confirm the action associate with the rank , check if the user rank is allowed to update table
CHAPTER 7

FUTURE WORK

We suggest to be able to use Social Security number to retrieve the information of the new member from a government agency such as city hall to confirm the entered information for more secure authenticated process. Unfortunately it’s not possible for us now. The system may have a history tracker for each user, this report will be helpful in case of changing the Rank of any member. Trust has different definitions. The difficulties are how to calculate trust and different kind of relationships and interpreter in numbers. People trust each other based on different reasons and different experiences, what a P1 think a trustworthy relationship as siblings may not be agreed by P2. In the future we may have more insight in this area, it was a time limitation. One suggestion is to collect the relationship in the system and compare the relationships and add higher rank to more repeated relationship.
CHAPTER 8

CONCLUSION

We have investigated the use of XACML rules and related tools for specifying the community authentication/authorization rule set to test the created rules. We also have installed OpenAM with different versions of ccAuth and found its deficiency in supporting the XACML policy created by other tools. This made us take different turn, for that reason we developed a ccAuth web app to allow secure resources access using the ranking variable. It is implemented in Asp.net with MySQL database support.

The ccAuth system can be integrated with mobile authentication system developed by Kalid Alkhattabi for disaster relief situations.
CHAPTER 9

REFERENCES


10.1 Configuration of ASP.NET visual studio

1. Download ISO images from visual studio.

2. Select Visual Studio 2015
   a. Visual Studio enterprise Edition 2015 and choose you operating system for download format.

3. After download go to the destination folder and open the files.

4. Open the vs_enterprise.exe
5. Choose your installation location, it’s detect automatically c:/ space from your hard disk. It has required at least 8-10Gb space.

6. Select Typical after click INSTALL.
7. The installation process will start.

8. The Setup completed message. And then click LAUNCH.
9. Visual studio installed
10.2 Configuration of ASP.NET core

1. Install .NET Core

Follow the steps based on which operating system you use

2. Create a new .NET Core project:

   mkdir aspnetcoreapp

   cd aspnetcoreapp

   dotnet new

3. Update the project.json file to add the Kestrel HTTP server package as a dependency:

   {
     "version": "1.0.0-*",
     "buildOptions": {
       "debugType": "portable",
       "emitEntryPoint": true
     },
     "emitEntryPoint": true
   },
"dependencies": {},

"frameworks": {

"netcoreapp1.0": {
"dependencies": {
"Microsoft.NETCore.App": {
"type": "platform",
"version": "1.0.0"
}
},

"Microsoft.AspNetCore.Server.Kestrel": "1.0.0"
},

"imports": "dnxcore50"

}}
4. Restore the packages:

    dotnet restore

5. Add a Startup.cs file that defines the request handling logic:

    using System;

    using Microsoft.AspNetCore.Builder;

    using Microsoft.AspNetCore.Hosting;

    using Microsoft.AspNetCore.Http;

    namespace aspnetcoreapp {

      public class Startup {

        public void Configure(IApplicationBuilder app) {

          app.Run(context => {

            return context.Response.WriteAsync("Hello from ASP.NET Core!");

          });

        }

      }

    }

6. Update the code in Program.cs to setup and start the Web host:
using System;

using Microsoft.AspNetCore.Hosting;

namespace aspnetcoreapp
{

public class Program
{

public static void Main(string[] args)
{

var host = new WebbHostBuilder()
.UseKestrel()
.UseStartup<Startup>()
.Build();

host.Run();
}
}

7. Run the app (the `dotnet run` command will build the app when it’s out of date):

dotnet run
8. Browse to http://localhost:5000

For more information you can go to https://docs.asp.net/en/latest/getting-started.html

10.3 Configuration of OpenAM

To configure OpenAM the link has the steps for it


10.4 The configuration of Web Agent:

[root@aalotaib bin]# ./agentadmin --install --acceptLicense

OpenAM Web Agent for Apache Server interactive installation.

Enter the complete path to the httpd.conf file which is used by Apache HTTP Server to store its configuration.

[q or 'ctrl+c' to exit]

Configuration file [/opt/apache/conf/httpd.conf]: /etc/httpd/conf/httpd.conf

Change ownership of created directories using User and Group settings in httpd.conf

[q or 'ctrl+c' to exit]
(yes/no): [no]: no

To set properties from an existing configuration enter path to file

[ q or 'ctrl+c' to exit, return to ignore ]

Existing OpenSSOAgentBootstrap.properties file:

Enter the URL where the OpenAM server is running. Please include deployment URI also as shown below:

(http://openam.example.com:58080/openam)

[ q or 'ctrl+c' to exit ]

OpenAM server URL: http://aalotaib.csnet.uccs.edu:8080/openam

Enter the Agent URL as shown below:

(http://agent.example.com:1234)

[ q or 'ctrl+c' to exit ]

Agent URL: http://aalotaib.csnet.uccs.edu:8000

Enter the Agent profile name

[ q or 'ctrl+c' to exit ]

Agent Profile name: WebAgent

Enter the Agent realm/organization

[ q or 'ctrl+c' to exit ]
Agent realm/organization name: [/]:

Enter the path to a file that contains the password to be used for identifying the Agent

[q or 'ctrl+c' to exit]

The path and name of the password file: /tmp/pwd.txt

Installation parameters:

OpenAM URL: http://aalotaib.csnet.uccs.edu:8080/openam

Agent URL: http://aalotaib.csnet.uccs.edu:8000

Agent Profile name: WebAgent

Agent realm/organization name: /

Agent Profile password source: /tmp/pwd.txt

Confirm configuration (yes/no): [no]: yes

Validating..

Validating..Success