BEST PRACTICES IN DESIGNING AND IMPLEMENTING CLOUD AUTHENTICATION SCHEMES

Zhihao Zheng, Yao Zhang, Vinay Gurram,
Jose Salazar Useche, Isabella Roth, Yi Hu

Department of Computer Science, Northern Kentucky University,
Highland Heights, Kentucky USA 41099

ABSTRACT

At present, the development and innovation in any business/engineering field are inseparable from the computer and network infrastructure that supports the core business. The world has been turning into an era of rapid development of information technology. Every year, there are more individuals and companies that start using cloud storages and other cloud services for computing and information storage. Therefore, the security of sensitive information in cloud becomes a very important challenge that needs to be addressed. The cloud authentication is a special form of authentication for today's enterprise IT infrastructure. Cloud applications communicate with the LDAP server which could be an on-premises directory server or an identity management service running on cloud. Due to the complex nature of cloud authentication, an effective and fast authentication scheme is required for successful cloud applications. In this study, we designed several cloud authorization schemes to integrate an on-premises or cloud-based directory service with a cloud application. We also discussed the pros and cons of different approaches to illustrate the best practices on this topic.

KEYWORDS

Cloud Application Authentication, Identity Management in Cloud, IAM.

1. INTRODUCTION

With the development of science and technology, more and more people are now using cloud services, such as cloud storage, cloud database, etc. The concept of cloud computing has been applied in a lot of fields, such as finance, medicine, education, and manufacture [3]. Cloud computing - at least as an extension of virtualization, has become increasingly influential. The advantages in cloud technology are that large amount of data can be stored and computed at a very low cost. Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

Identity management in the cloud is particularly difficult since the identity itself has cross-border features. At the same time, the identity management could have a critical impact from both architectural and organizational point of view. Many businesses are afraid of using the cloud because it would expose themselves to possible attacks and data corruption [2]. In addition, many companies do not have sufficient resources to manage identity authentication in the cloud because they lack flexible identity management to cover both on-premises and cloud native

David C. Wyld et al. (Eds): ACSTY, AIBD, MLSC, CCCIOT, NATP - 2021
pp. 67-76, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110307

applications. Due to the complex nature of cloud authentication, an effective and fast authentication scheme is required for a successful cloud application [5]. When implementing a cloud authentication scheme to integrate an on-premises or cloud-based directory service with a cloud application, we aimed to discuss the pros and cons of different approaches to illustrate the best practices on this topic.

2. BACKGROUND AND MOTIVATIONS

2.1. Standards to Improve Scalability

Cloud computing allows people to access a configurable pool of computing resources—networks, servers, storage, applications, and services over the network at any time and on a convenient, ready-to-use basis. These can be quickly prepared and published with minimal administrative effort, or interaction with service providers [2]. It allows organizations to instantly add processing power or functionality while not investing in new infrastructure, training new employees, or purchasing new software licenses [4].

Cloud computing covers any subscription-based service that extends existing IT capabilities on the Internet in real time. Public cloud usually refers to software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS). A private cloud is an application or platform that is specific to a particular organization and deployed on its own site, often hidden behind a firewall [3].

2.2. Performance Scalability

When we think about scalability, most people immediately think of how a system handles large-scale transactions, floating-point operations per second, and so on. A key feature of the cloud is its ability to meet changing needs by adding or subtracting computing power, providing elastic scalability [7]. For example, when it is urgent to require large-scale computing power, an application can be upgraded quickly on the Amazon Elastic Compute Cloud (EC2) to use a virtual machine with larger computing and storage capabilities. To extend this concept a little further, the new cloud application is designed to support the linear expansion of the architecture of N+1, which can support nearly infinite scale of computing operations.

2.3. Integrate and Manage Scalability

A not-so-comprehensive and scale-related challenge is about how quickly organizations can deploy, integrate, and manage a system over time [6]. When the system causes friction - such as in administrative tasks, this can cause system scalability barriers, especially for identity management [4]. If the infrastructure is defined as common hardware, software, and network services that generate IT capabilities within the enterprise, the identity management infrastructure includes directory services, identity and access management services, network proxies, and verification systems used throughout the enterprise. Many companies today are trying hard to build an identity infrastructure that can work under a cloud architecture and can gracefully upgrade in a cloud fashion.

To be able to upgrade to meet cloud architecture and growth needs, system architects must focus on optimal management and integration of identities [2]. Identity management is a key bottleneck for adopting the cloud for many businesses. Architects understand that their vision must go beyond the basic performance level of cloud scalability, and also design a strategy that allows the management and integration of identities to be scalable.

2.4. A Cloud-level Identity Structure

Through different technologies, standards, and use cases, the identity verification can be obtained across the previously separately managed security domains [8]. So that users in one domain can securely and seamlessly access data and systems in another domain without the need for redundant user management. With this federated identity, many elements and fields are intertwined, just like weaving cloths.

In the past, organizations stored network identities in various directories and identity databases. With the growth of the Internet and the emergence of cloud applications, they have discovered that they need to manage identities outside the traditional network [6]. Today's network administrators must manage multiple accounts for corporate and cloud applications. This duplication of labor increases the workload and leads to security risks due to administrators having to manage multiple user identities and passwords. Cooperation with outside partners and contractors also requires the company to open up network boundaries to outsiders.

To ensure the security of so many information assets and data, companies need to seamlessly use identity management to connect to the cloud. To achieve successful cloud identity management, the industry must ensure that the identity meets the unique architectural needs of the cloud, and identities are regarded as a structure that will be integrated, abstracted, and extended. The identity is delivered as SaaS, just like the cloud platform itself supports [5].

3. OUR MODEL AND APPROACHES

3.1. An Abstract Concept

To implement a cloud-level identity structure, it requires the abstraction of identities into identity services [9]. Application developers historically plug identities into the application itself and maintain a local user base to perform authentication. This leads to redundant and often stale data, passwords, and greater help center overhead.

In the past decade, applications began to externalize identity management, starting with an external directory that intensively authenticates users based on Lightweight Directory Access Protocol (LDAP). This is an important step for the scalability of identity management, but we need to do more - LDAP password authentication is not enough. Businesses must be able to use more than one type of certification, depending on the level of threats to be applied [7].

3.2. Problem Description

With the increasing of information processing and storage needs, enterprise users have more and more demands for the efficient information synchronization and service collaborations. Nowadays, cloud storage and cloud computing has become a popular resource. The cloud has the characteristics of convenient resource sharing, low maintenance and management cost, and large scale. Enterprises are more likely starting to build their cloud data center. However, most of the enterprise data are still stored locally and cannot be perfectly connected with public or private clouds [10]. How to integrate local and cloud storage easily and improve the utilization rate of cloud resources is a problem that many enterprises are facing. Most companies do not store all data on the cloud, since some of company’s data are highly classified like bank’s credit card information. So organizations will keep some important data on the local server, and other data will be saved on cloud storage.

In our experience, we are going to set up three different solutions to test which solution could provide us high performance, high security and low cost. The tools we are going to use are Windows sever 2016 and Amazon Web Services (AWS). Windows sever 2016 is a server operating system, and Active Directory Federation Services is possible to configure AD FS to authenticate users stored in non-AD directories. AWS (Amazon Web Services) provide services to individuals, companies and governments [8].

Storing data locally means the enterprise has its own dedicated data center. Traditionally, this is how most organizations design and maintain networks. Regardless of the other aspects, this requires physical hardware, the space required for the hardware, and backup and disaster recovery services [12]. On the another hand, some enterprises choose to store data in the cloud. Cloud is actually a server network that serves different functions from each server. Some servers store data and some run applications. We could easily notice that we are tending to not buy boxed software from stores, but we pay the monthly fee on the Internet access platform. This is a real running cloud.

In this study, we designed cloud authorization schemes to integrate an on-premises or cloud-based directory service with a cloud application. We also discussed the pros and cons of different approaches to illustrate the best practices on this topic.

The Amazon Web Services (AWS) is a professional cloud computing service that offered by Amazon [11]. It was launched in 2006 to provide IT infrastructure Services to businesses in the form of Web Services. The services that AWS provides including elastic compute cloud (Amazon EC2), Simple Storage Service (Amazon S3), Simple database (Amazon backs), Simple Queue Service and the Amazon CloudFront… etc. As the largest services provider, Amazon AWS provides infrastructure and services that build reliable, fault-tolerant, high-availability systems in the cloud [13].

The cloud application communicates with the LDAP server which could be an on-premises directory server or an identity management service running on cloud. Due to the complex nature of cloud authentication, an effective and fast authentication scheme is required for a successful cloud application.

3.3. Models and Procedures

3.3.1. System Architecture

The foundation of the system involves setting up the Windows Server 2016 and building sample Rails login web application by using AWS for Rails Developers. This step is basically for us to get used with everything we were going to use such as Windows Server 2016, Amazon web service (AWS), and Ruby.

We used the SDK with Ruby on Rails. The AWS SDK for Ruby helps take the complexity out of coding by providing Ruby classes for almost all AWS services, including Amazon Simple Storage Service, Amazon Elastic Compute Cloud, and Amazon DynamoDB [9]. Before we use it, we need to install the AWS SDK for Ruby. Then we tried a couple of commands to test whether it works correctly. Such as creating a bucket, adding files to the bucket, listing the contents of that bucket, etc.

Then we configured the AWS SDK for Ruby by using the AWS access keys. We also set up the AWS credentials. After that we set up the region and nonstandard endpoint.

We integrated the AWS SDK for Ruby with Rails, then we used the Amazon SES that support for ActionMailer. After that we could log in by entering the target webpage on the browser. The screenshot illustrated in Figure 1 shows the login page to be adapted with different authentication schemes.

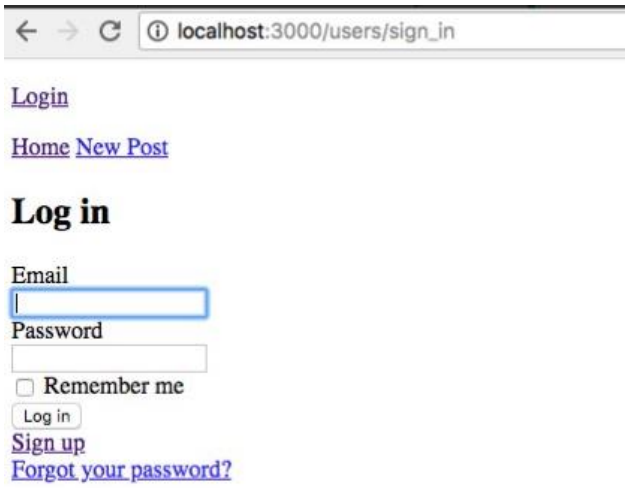

Figure 1. Login Page to be Adapted with Different Authentication Schemes

3.3.2. Model 1: On-Premises Authentication Service with a Cloud Application

Then, we set up the first solution: on-premises authentication service with a cloud application: we used ADFS connected with AWS, enabling federation to AWS using Windows Active Directory, ADFS, and SAML 2.0.

For this step, we used the active directory federation service (ADFS) to make connection with the cloud application. We used the Amazon Elastic Beanstalk that allows users to quickly deploy and manage their applications without configuring the infrastructure that runs those applications [11]. The connection between authentication service and the application is shown in Figure 2.

Firstly, we installed the AD Federation Service on Windows Server 2016. Then we enabled our users to access Office 365 with AWS managed Microsoft AD. We added two containers by ADFS to the AWS Microsoft AD. Then we installed the ADFS, we integrated ADFS with AD [8].

After that, we deployed a Ruby on Rails application to Elastic Beanstalk [9]. These are the steps we did:

1. Create a Rails App to Deploy
2. Create an Application on Elastic Beanstalk
3. Install AWS CLI and EB CLI
4. Create an Environment on Elastic Beanstalk
5. Set Up an RDS Database
6. Observe Our Working App

Although there are too many choices to pick from for this step, it’s very important to use the Rails application directory when running the AWS and eb commands.

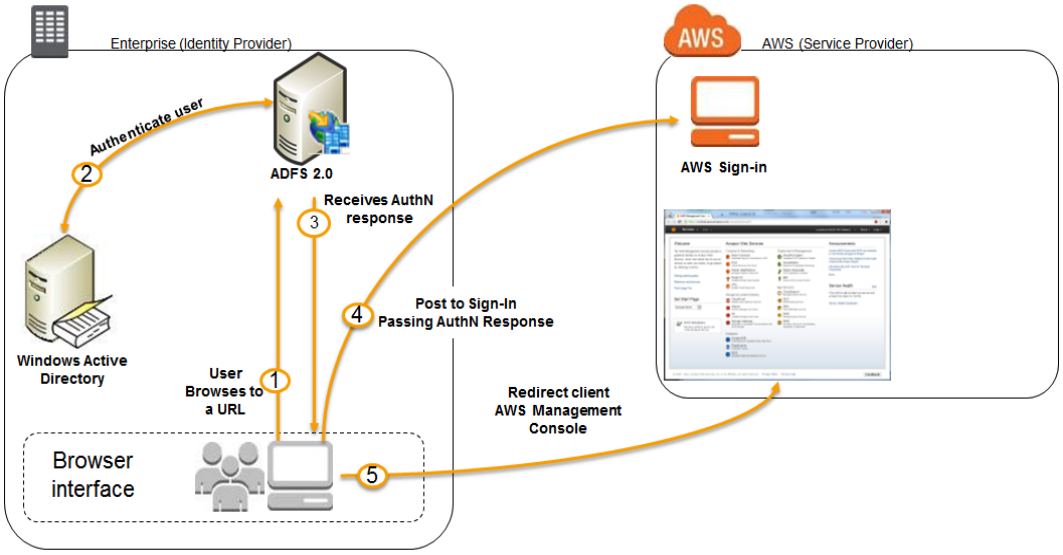

Figure 2. On-Premises Authentication Service with a Cloud Application [14]

3.3.3. Model 2: Third-Party Authentication Service with a Cloud Application

Next, we set up the second solution: third-party authentication service with a cloud application: build and deploy a federated internet identity application with AWS Elastic Beanstalk and then log into it with Amazon.

For this step, the third-party service we used is the Amazon account. We were trying to let our users use their Amazon account to log in when they were on our website. Since Amazon is very popular and people almost using Amazon every single day, it is easy to connect with Amazon instead of other account such as Google or Facebook.

To build and deploy a federated web identity application with AWS Elastic Beanstalk and Login with Amazon [12], we did the following steps as shown in Figure 3. This is the identity authentication scheme in the authentication process, the following picture illustrates the entire process.

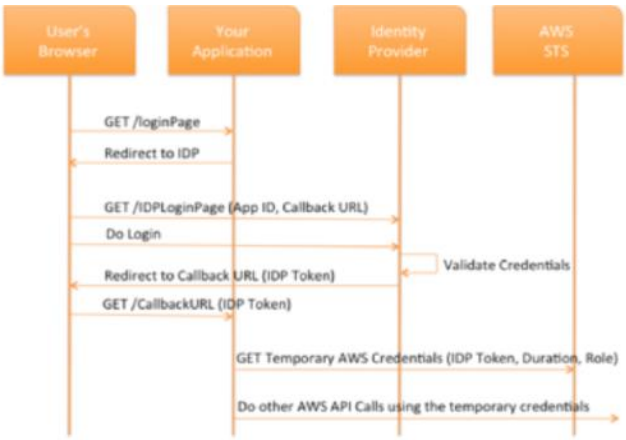

Figure 3. Identity Federation [15]

To develop a web-based application using Federated Web Identity, we firstly deploy the application to the Amazon Elastic Beanstalk. Secondly, we register our application with the Amazon Identity Provider. Then for our application, we need to define the permission in AWS. After that, for the load balancer, we configured SSL certificate. Then we configured our application on AWS. In the end, we tested our application to get the login page adapted with the third-party authentication service as shown in Figure 4.

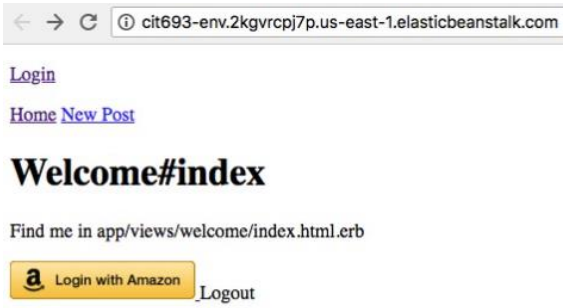

Figure 4. Login Page Adapted with Third-Party Authentication Service

3.3.4. Model 3: Cloud Authentication Service with a Cloud Application

In this step, we set up the third solution: cloud authentication service with a cloud application: log on to AWS services by using on-premises active directory: For this step, we used the AWS Directory Service for Microsoft Active Directory, it also known as AWS Microsoft AD. It enables our directory-aware workloads and AWS resources to use managed Active Directory in the AWS Cloud [13]. AWS Microsoft AD is built on actual Microsoft Active Directory and does not require us to synchronize or replicate data from your existing Active Directory to the cloud [11]. Figure 5 and 6 shows the AWS directory service for Active Directory and cloud directory details.

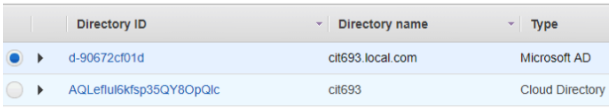

Figure 5. Directory Names and IDs for AWS Directory Service for Active Directory

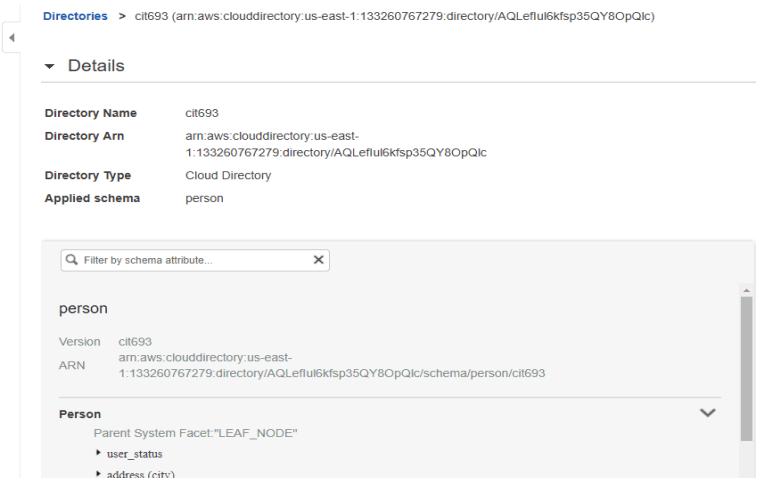

Figure 6. AWS Cloud Directory Example

We also created an Angular2 web application and implemented a third party authentication method. With the purpose of implementing a third party application to achieve authentication, we created a simple Angular2 Single-Paged web application.

Taking advantage of the fact that Angular2 was conceptualized to work as single components for better performance and simplicity in writing code, we created three components. The first component, containing the "Home" view, which only renders a simple line of text.

The second component, called the "Navbar," renders a navigation bar where some buttons will be placed that when pressed will call a service which will display a series of options for the user to decide which profile and/or social network will be used to later perform authentication. The third component, called "protected," will render information retrieved from the profile used to perform authentication once the user is properly authenticated.

In order to successfully perform authentication, a third party application called "Auth0" was implemented. Auth0 offers a free subscription as long as there are less than 7000 active users. Auth0 is not only very easy to be integrated as an authentication service, but also serves as an universal authentication layer for both on-premises and cloud native applications.

4. RESULTS AND CASE ANALYSIS

The use of these three cloud authentication solutions is highly dependent on the size of the institution/organization. Table 1 illustrates the comparison result based on cost, security, ease of maintenance, and client data management.

Table 1. Cloud Authentication Comparison

	Cost	Security	Maintenance	Client Data Management
ADFS	High	Highly Secure	Hard	Hard to management, but easy to obtain data
Third-Party	Median	Relatively not Secure	Easy	Easy to management, but hard to obtain data
AWS Cloud Directory	Low	Secure	Relatively Easy or Median	Easy to management, but hard to obtain data

We have some cloud authentication suggestions for these three kinds of clients below:

1. For start-up companies: we should mostly think about the cost and the number of customers. So, using the third-party and AWS directory solution not only can reduce the cost, but also give you a chance to attract customers from some third-party platforms. In addition, the maintenance will be relatively easy since they don't have to hire too many employees to manage these account and data. For the maintenance of server, they don't have to hire people as well since the third-party and AWS directory service have already cover the maintenance as well.

2. For median-size companies: for this kind of companies/organizations, they may already have the local active directory. So that they can easily use the ADFS to authenticate the

web applications. For example, some organizations like university will highly likely to use this solution. On another hand, for those companies that are doing business/entertainment, they may want to attract clients from third-party web applications. they can easily add the third-party connection on their web apps.

3. For large organizations: in this case, they usually have a large group of cloud service professionals, great budget, as well as multi-cloud services. They are very likely to use federated authentication schemes. We recommend them to add the third-party solution to make it more convenience for their clients to login.

5. CONCLUSIONS

In this study, we set up three different cloud authentication solutions to test which solution could provide us high performance, high security and low cost. We designed several cloud authorization schemes to integrate an on-premises or cloud-based directory service with a cloud application. We also discussed the pros and cons of different approaches to illustrate the best practices on this topic. Our goal is to illustrate the best practice on cloud authentication based on usage scenarios.

ACKNOWLEDGEMENTS

We greatly appreciate Dr. Traian Marius Truta, for helping review the draft version of the paper and providing comments.

REFERENCES

[1] D. Chang, M. Benantar, J. Chang, V. Venkataramappa, “Authentication and authorization methods for cloud computing security”, USPTO Patent Grants, July 2014.
[2] D. Chadwick, K. Fatema, “A privacy preserving authorisation system for the cloud”, Journal Of Computer And System Sciences, Canterbury, United Kingdom, 2013.
[3] W. Smari, A. Navarro, W. McQuay, B. Tang, R. Sandhu, Q. Li, “Multi-tenancy authorization models for collaborative cloud services”, Concurrency and Computation, November 2014.
[4] E. Bertino, F. Paci, R. Ferrini, and N. Shang, “Privacy-Preserving Digital Identity Management for Cloud Computing”, In IEEE Data Engineering, pages 21–27, 2009.
[5] M. Dragos Marian, “Cloud Identity and Access Management – A Model Proposal.”, Journal Of Accounting And Management Information Systems, Romania, 2012
[6] T. Piepers, “Cloud Identity & Access Management Model: success factors for Identity & Access Management in cloud computing.”, Masters Thesis, Netherlands, 2013.
[7] S. Smita and M. Deep, "Identity Management issues in Cloud Computing", International Journal of Computer Trends and Technology (IJCTT), vol. 9, no. 8, 2014.
[8] L. Song, C. Jie, Z. Hong, L. Lu, “Public Auditing with Privacy Protection in a Multi-User Model of Cloud-Assisted Body Sensor Networks”, Sensors, Vol. 17 Issue 5, p1-19, Ipswich, MA, May 2017.
[9] F. Nzanywayingoma, Y. Yang, “Efficient Resource Management techniques in Cloud Computing Environment: Review and discussion.”, Telkomnika, Vol. 15 No. 4, pp. 1917-1933, Beijing, China, December 2017.
[10] S. Rizwana, M. Nerul Navi, M. S, M. Kharghar Navi, “Identity Management in Cloud Computing”, International Journal of Computer Applications, Vol. 63, No. 11, 2013.
[11] U. Habiba, R. Masood, M. Shibli, M. Niazi, “Cloud identity management security issues & solutions: a taxonomy”, Complex Adapt Syst Model, Vol 2, No. 5, 2014.
[12] M. Darwish, A. Ouda, L. Capretz, “Cloud-Based Secure Authentication (CSA) Protocol Suite for Defense against DoS Attacks.”, Journal of Information Security and Applications, 2015.
[13] P. Cigoj, B. Blažič, “An Authentication and Authorization Solution for a Multiplatform Cloud Environment. Information Security Journal: A Global Perspective”, vol. 24, no. 4-6, pp. 146-156, Ljubljana, Slovenia, August 2015.

[14] AWS, Enabling Federation to AWS Using Windows Active Directory, ADFS, and SAML 2.0. https://aws.amazon.com/blogs/security/enabling-federation-to-aws-using-windows-active-directory-adfs-and-saml-2-0/
[15] AWS, Identity federation. https://aws.amazon.com/identity/federation/

AUTHORS

Zhihao Zheng was a graduate student at Northern Kentucky University majoring in computer information technology. He is interested in Database Systems, Data Security, Data Mining. He has worked on several projects such as AWS solution Architecture, Data Mining in Python and Weka, Effect of Virtualization on System Performance, etc.

Yao Zhang was a graduate student at Northern Kentucky University majoring in computer information technology. She is interested in Database Systems, Data Security, Data Mining. She has worked on several projects such as AWS solution Architecture, Design and Analysis of Experiments, Applied Mathematical Models, etc.

Jose Salazar Useche is an undergraduate student at Northern Kentucky University majoring in Computer Science. He is interested in Machine Learning, Computer Vision and Quantum Computing. He has worked on several projects such as a Yoga postures classifier using Google's AIY Vision kit, a Spotify-like music catalog and a task manager app.

Dr. Yi Hu is a Professor of Computer Science at Northern Kentucky University. He is also a CISSP and CEH. He has published more than 30 papers on security and trust management. In addition, he is the Director of Center for Information Security at NKU and Director of Kentucky Collegiate Cyber Defense Competition with extensive experience on promoting security education and awareness.

Vinay Gurram was a graduate student at Northern Kentucky University. His research interests are cloud computing and cybersecurity.

Isabella Roth is an undergraduate student at Northern Kentucky University. Her research interests are cloud authentication and cloud data security.