Experiment No. 7

Aim: To study and Implement Security as a Service on AWS/Azure

Objective: To understand the Security practices available in public cloud platforms and to demonstrate various Threat detection, Data protection and Infrastructure protection services in AWS and Azure

Theory:

As the use of file sharing increases across the industry, more attention is being paid to the inherent security of these solutions and the need for corporations to provide enterprise file sync and share (EFSS) solutions that meet IT's security parameters.

Security Features of ownCloud

ownCloud's drivers for continued security improvement is to not only fix individual symptoms (e.g. the single bugs), but to also focus on identifying and resolving the root cause to prevent whole categories of vulnerabilities. ownCloud internal security processes and secure software development lifecycle aligns with industry standards such as ISOs 29147, 30111 and 27304.

- **Strict Content Security Policy :** Content Security Policy (CSP) is one of the most useful and powerful web security features introduced in recent years. With CSP, applications can instruct the browser to follow a specified security model, including instructions to not execute any inline scripts or load remote resources.

- **Data in Session is Stored Encrypted:** PHP stores session related data within sessions. These are usually small files on the server containing data such as the login state or the username. We have hardened the PHP session storage in such a way that the ownCloud server can only read session data at the same time the user is using ownCloud. This is done by encrypting the stored session data with an encryption key stored in another cookie. If the userrequests a page on ownCloud the encryption cookie will be sent by the sync clients or the web browser. Only with this cookie (which is not stored on the disk of the application server) can the session content be decrypted.

- **Secure by Default Model:** New ownCloud code uses the so called "ownCloud App Framework", a modern MVC-like framework to develop code for ownCloud. Code relying on this framework uses a lot of secure defaults such as requiring CSRF (another specific kind of web vulnerability caused by the original design of the web) and authentication checks being opt-out rather than the more common (and less safe) opt-in. The default mode for every critical security feature in ownCloud is "on", and requires the developer to deliberately "opt-out" of these security checks. These secure defaults are part of ownCloud's secure software development lifecycle. Secure defaults make it