App Dev - 1

by Gagneet Kaur

Date: Dec 06, 2022

MAD-1 WEEK 9

ACCESS CONTROL

- Access control refers to the general way of controlling access to read / write / modify information in any web application.
- Web application need access control to allow users with different backgrounds to use the application without any security issues & risks.
- To implement CRUD operations, read-write access is required
- Example of Access Control Apps : Linux file system, Email, Ecommerce, etc.

Discretionary Access Control (DACs)

→ The owner can transfer authenticated objects or information access to other users.
→ The owner can determine the access type of other users.

Mandatory Access Control (MAC)

→ In MAC, the users cannot override the security policies accidentally or intentionally.

- In a flask app, controller is where you apply access control directives.

• Log files tend to be mostly write operations & few read operations.

PAGE NO.
DATE:

SECURITY MECHANISM

• ACCESS CONTROL is the way of limiting access to a system/ resource and granting permissions to only those users who have valid credentials.

• Role Based Access Control (RBAC) is the method of restricting system & access based on the role of authorized individual users within an enterprise or organization. In RBAC, users can switch between roles.

Benefits of "principle of Least Privelege" : → It recommends that users access to system resources should be restricted to the minimum they need to do their work.
- Better Security
- Better Stability
- Ease of Deployment

• TOKEN is the most secure way of authentication method to control access given to a web application.

• Token-based authentication is a protocol that allows users to verify their identity for one-time & they will receive a unique machine generated encrypted string in exchange.

• Access Tokens would be preferred method for machine-to-machine authentication without password.

• HTTPS certificates are associated with A domain name.

Cookies are used to
- store session information
- store values that the server needs to know about the client connection.

SESSIONS
↳ are primarily used to give the server info about what the client was doing previously

- 401 status code represents "unauthorized" user.
- Client-side sessions are completely stored in the cookie
- Server-side sessions are stored on server & are looked up from cookie.
- SERVER SIDE INFORMATION :
  - → It requires persistent storage at the server
  - → It may require multiple backends & storage types.
- SESSION MANAGEMENT :
  - → Session management manages session between the web application & the users.
  - → In the client side session, the session information is completely stored in a cookie.
  - → In the server side session, the session information is stored on the server.

Secret keys are used by the server while generating cookies to ensure that the user cannot modify the content of the cookie.

HTTPS
↳ HTTPS

- Response flow for HTTP authentication framework

1. The server responds to a client with a 401 (unauthorized) response status.
2. Server provides information on how to authorize with a WWW-Authenticate response header containing at least one challenge.

- Server Authentication determines who is the correct & authorised server in order to establish security.

PAGE NO.
DATE:

3. The client (typically a browser) presents a password prompt to the user.

4. A client that wants to authenticate itself with the server, requests again, including the Authorization request header with the appropriate credentials.

- Client Certificate Authentication

Client certificate is more often found in corporate intranets than in public internet.

→ A client certificate authentication is basically a process by which client is securely permitted to access a server by exchanging a digital certificate called CLIENT AUTHENTICATION CERTIFICATE.

→ A client certificate ensures the server that it is communicating with a legitimate user.

→ In case the client is proved to be legitimate, the client does the handshake with the server in order to exchange information.

- HTTP & HTTPS → provides security for the connection.

→ HTTP sets up an open connection between the client & the server on a fixed port.
→ In case of HTTP, the data can be tapped or altered.
→ HTTPS sets up an 'encrypted' channel between the client & the server.
→ In context of HTTPS, an encrypted channel between the client & the server is established using SSL. (secure sockets layer)

→ HTTPS reduces the likelihood of someone being able to intercept what is on the link itself.

→ When you connect to GMail & enter password, the password will be sent over the HTTPS link as plain text & visible to the server.

LOGGING → is typically used to detect unusual activity on the server.

→ In the context of HTTPS, a client want to submit username & password to the server using ~~or~~ GET method which will establish an insecure link between client & the server.

- Incognito mode is a feature that allows browsing the web without storing the browsing data.
- The web server uses cookies to track users as they navigate the website
- INDICATION OF POSSIBLE SECURITY ATTACK :
  - large no. of requests in short duration
  - requests with "malformed" URLs.
  - repeated requests to unusual endpoints.
- EXAMPLE OF TIME SERIES PROBLEM :
  - no. of requests made per unit time into particular site.
  - time of a user logging in into a social media site.
- Time Series Databases are optimized for queries such as :
  - no. of visitors to a website in the last month.
  - time of day when traffic on a site is maximum.

HTTP : → In an HTTP GET request, all parameters of the request can be logged as part of the URL.

→ There are many security issues with HTTP basic & there are better auth mechanisms.

Date : Dec 06, 2022

PAGE NO.
DATE:

APPLICATION DEVELOPMENT
WEEK 10

Application Testing

→ Static & Dynamic Testing (It improves the quality of software application by finding error in early stages of software dev. process.)
- ↳ Code review
- ↳ functional tests ; apply suitable inputs

→ **WHITE BOX TESTING** : tests can be created based on knowledge of internal structure (can examine internal variables)

→ **BLACK BOX TESTING** : tests based on how it would look from outside (actual code can't be seen) (only interfaces are available)

→ **GREY BOX TESTING** : hybrid approach b/w white & black box
- enforce interface as far as possible

→ **Regressions** : loss of functionality introduced by some change in the code
- Future modifications should not break existing code
- But sometimes it is necessary, ie, in case of updating API versions.

• Code Coverage

(1) FUNCTION COVERAGE (test invokes that every fn get called once)
(2) STATEMENT COVERAGE (all statements in code will be executed)
(3) BRANCH COVERAGE (atleast 2 tests needed ; branch taken ; branch not taken)
(4) CONDITION COVERAGE (atleast 2 tests needed ; condition fails or succeeds)

Test fixtures are used to set up the environment & data needed for a test.

PAGE NO.
DATE:

LEVELS OF TESTING

• Sequence of levels of testing :
Unit Testing → Integration Testing → System Testing → User Acceptance Testing

• Initial Requirements gathering
- Stakeholders
- Functionality - Non-functional req.
- extensive discussions with end users required

• UNITS OF IMPLEMENTATION
- break functional requirements down to small, implementable units
- Each one may become a single controller

• UNIT TESTING
- test each individual unit of implementation

• INTEGRATION TESTING (used to bring out additional sources of error that may come from interactions between components.)
- Integration : when an app consists of multiple modules
- problems : individual units works - combined sys does not
- continuous integration : each commit to main branch triggers a re-evaluation of integration tests.

• SYSTEM LEVEL TESTING (requires fairly detailed knowledge of the computing environment in which the application will finally run.)
- Includes server, environment
- mainly Black box : should validate final usage.

• SYSTEM TESTING AUTOMATION
- has to simulate actual user interaction (includes DB ; persistent connections)

• USER ACCEPTANCE TESTING
- deploy final system
- Beta Testing

Automation Testing Framework:
1. Selenium
2. Katalon
3. Cucumber

PAGE NO.
DATE:

TEST GENERATION

- API - based testing is a type of software testing that analyzes & validates an API & verify its expected functionality, security & performance.

- MBT → - model based testing - software under test is checked against predictions made by model
  - It is a software testing technique in which the test cases are generated automatically from a model of the system.
- MBT activities in a test process should follow a sequential order.
- A model in MBT is a description of a system's behaviour, e.g., input sequences, actions, conditions, output & flow of data from input to output.

- GUI Testing:

- save all gui elements for size, position, width, length & acceptance of character or numbers present on the page.
- are the font used in an application readable.
- are the warning messages / error messages displayed correctly
- checking alignment of text is proper.
- checking the positioning of GUI elements for different screen resolution.

- Security Testing:

- It points out the ~~vult~~ vulnerabilities, threats, risks in a software application.

- Security testing tool : Owasp

- Checking cookies & session time for application can be a potential test case for security testing.
- Fuzz testing or Fuzzing is a type of security testing.

PYTEST

- PyTest & PyUnit (UnitTest) in python allows users to write test codes using Python.

White Box Vs Black Box Testing

White Box Testing	Black Box Testing
White Box Testing is focused on the code conditions, structure of the code, designing test cases.	Black Box testing focus on the functionality & behaviour of the application, system testing, acceptance testing, security testing.
It is performed by a group of testers having full knowledge of internal structure of the application.	It is done without the knowledge of the internal structure of the software application.

- White Box Testing involves the testings of the software code for
  - the flow of specific inputs through the code
  - the functionality of conditional loops.

Shriansh
PAGE NO.
DATE:

- Methods for testing functional correctness of an app :
  - Code Review
  - Automated Theorem provers
- White Box Testing reveal the most bugs in a program.
- 100% Code Coverage during testing indicates that
  - all lines of code have been covered by at least one test.
  - the application may still have bugs
- Black Box Testing may be preferable over white box testing because it allows cleaner separation between the teams doing implementation & testing.
- PROs of Black Box Test :
  - testers can be non-technical & may be able to test without having detailed functional knowledge of system
  - encourages clean abstraction of interface.
- CODE COVERAGE :

1. Condition coverage → cover all possible outcomes of each condition
2. Statement → all the statements in the code are executed
3. Branch → covers all exit routes of a branch
4. Function → functions in the code are invoked at least once

- Integration Testing : It checks the data communication amongst the smaller software modules. In this, units or individuals components of the software are tested in a group.

Date : Dec 07, 2022

MAD-1 WEEK 11

XML ; HTML (A markup language is used to control the presentation of data)
- XML has no predefined tags
- → HTML-5 is a living standard. - no major changes, but continuous adaptation.
- → SGML : meta lang → for defining markup languages
- → HTML & XML are based on SGML
- → HTML-5 is not based on SGML parsing rules.
  - → RSS feeds are primarily used to get information about the latest updates on a given page

JAVASCRIPT

- → Javascript provides the adaptation layers.
- → JS can be used directly to draw with the canvas API.
- → JS : object-oriented language & HLL.

> it can be used for functional programming & event driven programming !!

- → Javascripts is needed for :
  - adding interactive behaviours to web pages we make.
  - loading content into the document whenever needed, without reloading the entire page
  - It allows us to add dynamic behaviour & special effects to the webpage.

- → To embed JS into HTML document, <script> tag is used.
- → To define a function in JS, function keyword is used.

- → CORE JS allows developer to :
  - provide functionality such as dynamically creating HTML & setting CSS styles.
  - collecting & manipulating a video stream from a user's webcam
  - integrate with 3rd party frameworks & libraries that you can appl

HTML to accelerate the work of building sites & applications.
- JS can be used to write games.
- JS uses events to provide user interaction.

Framework

→ Purpose : 1. Basic functionality already available
→ Problem : lots of code repetition
→ Solution : design patterns

Single Page Application
→ effectively becomes an app itself which does not like multiple different web pages that you need to navigate through

→ 'React' is used to create JS ~~components~~ components.

→ syntax to change the contents of paragraph element given : `<p id = "MyId"> Hey There </p>`

```
document.getElementById("MyId").innerHTML = "My name";
```

→ Different Frameworks : 1. jQuery 3. Vue
2. .Net

Backend Framework : Express.js

→ Frameworks are structures with a particular context & help you create web applications within that context.

→ Working with JS, an app developer should have a good command on JS frameworks.

webcam
apply to

The primary goal of introducing web components is to promote the idea of code reusability.

Custom Elements

Web Components : 1. Shadow DOM
2. Custom Elements
3. HTML Templates

It is an API used to keep styling of component separate from the rest of the page

→ HTML 5 elements used to create a markup template : <slot> and <template>

→ The objects in the HTML DOM are organised according to HIERARCHY.

→ JS can manipulate cookies using the cookie property of the Document object. It can read, create, modify & delete the cookie or cookies that apply to the current web page.

- ~~Mak~~ Self closing tags :
, <rect>, <input>
- There are compilers to convert several languages to forms of JS that can run on existing browsers.
- Flask is a frameworks for building PYTHON based web apps.
- HTML templates & slots are used to create web components.
- Implementation of tag functionality can't be done purely in XHTML alone, something like JS is required.

Date : Dec 07, 2022

PAGE NO.
DATE:

MAD-1 WEEK 12 (Deployment)

Components of An App

- Requirements for Permanently Deploying An App
  - → dedicated servers
  - → always on internet connection
  - → uninterrupted power
- Application Scalability
  - → application scalability defines the potential for application to grow over time
  - → scaling depends on the entire system architecture & not just the software
- Deployment
  - → software deployment is the process to deploy the software to the production environment so users can start using the product.
  - → Deployment is always performed after the development of any application.

COMPONENT OF LOCAL DEVELOPMENT ENVIRONMENT :

- File System
- Desktop
- Text Editors or IDE

- What is Data Center?
  - → Data center consists of a set of computers located in one location that takes care of the components such as routers, switches, storage systems, cooling systems, servers & network connectivity that enable the delivery of shared apps & data among organizations.

SaaS - software as a service • takes responsibility for installing software & keeping software up to date with security patches
IaaS - infrastructure as a service
PaaS - platform as a service

- **Load Balancer**
  - It is used when there is multiple frontend present
  - It can be used to add additional layers of security to your application.
  - A load balancer by itself may not be very powerful since it only needs to forward the requests.

SERVICE APPROACH

- Examples of Online office Platforms : Google Docs ; Dropbox ; Cisco Webex

(1) **SaaS** : software as a Service
- A SaaS application is accessible via a web browser or lightweight client application.
- By the use of SaaS, we can host applications on a remote server. - Example : Gmail ; online office platforms

(2) **IaaS** : Infrastructure as a Service
- Its a form of cloud computing service that provides on-demand access to compute, storage & networking resources.
- Migrating your organizations infrastructure to an IaaS allows you to reduce on-premises data centre maintenance & save money on hardware.
- It gives maximum flexibity to a client to modify the application deployment.
- It delivers infrastructure & raw machines including servers, networking, power etc.
- In IaaS, resources can be purchased as needed.
- Examples : Cloud Compute Systems (AWS, Google compute Engine, Azure, Digital Ocean)

(3) PaaS : Platform as a Service

- Provides takes care of : → installing security patches on OS & frameworks.
  - power, network, machine management
  - OS installation, security patches
  - base application platform
  - multiple databases & connectivity options
- Developer needs to :
  - manage app code
  - specify req. on server sizing, DB, connectivity
- Examples : Replit, Glitch, Heroku etc.
  Google App Engine.
- CloudShell : It is a popular PaaS service, which is used to develop, build, debug and deploy our cloudbased apps & manage your google cloud resources with the flexibility of a Linux Shell.

DEPLOYMENT

- Version Control :

1. Centralized → ⇒ Client-Server Model
   ⇒ only one developer is allowed to work on a particular part of code at one time.
2. Distributed → ⇒ Peer to Peer Model
   ⇒ It is possible for two developers to be working on the same source file at same time.

- **GIT** : 1. It is free & open source
  2. It is distributed version control system.
  3. Git is designed to handle development of small & large projects.
  4. It was created by Linus Torvalds, the person who started the Linux kernel.

- **Benefits Of Version Control Systems** :
  1. It traces the changes made to the code.
  2. It simplifies the code review process and ultimately saves time.
  3. It stores modifications to code efficiently.

- **Best Practices** :

  Test driven ~~deploy~~ development → Code Review → Integration Pipeline optimization

- **Continuous Integration** → CI requires powerful servers that can run all tests on a repository every time a change is pushed to server.

- **Continuous Delivery** : Once CI testing passed, package files for release
  It can be used to automate the process of generating nightly builds of software that can be tested by users.
  It is an extension of CI.
  - automated delivery of "release package" on each successful testing.

- **Continuous Deployment** : Deploy to production
  - passed tests → deploy to users.

  ↳ problems :
  - presence of bugs that were not caught by tests
  - users may be forced to change to a new interface when they don't want to.

• Contin... after sent
• It ca...

CO...

- Continuous Deployment refers to the procedures that must be followed after code has been integrated in order for app updates to be sent to users
- It can be helpful to create a feedback loop with the end user with the app.

Shrinath
PAGE NO.
DATE:

CONTAINERS

- **APP ENGINES** : They are fully managed platforms for developing & hosting applications.
  It provides end to end management of an application.
- **COMPUTE ENGINE** : It allows you to create Virtual Machines
  Compute engines are more flexible than app engines.
- **CONTAINERS** :
  - Containers are self contained environment with OS and is packaged along with its libraries & dependencies in order to run the process.
  - They are primarily used with the linux kernel namespaces.
  - They are also used in the context of sandboxing.

  EXAMPLES : Docker, Kubernetes etc.
- **ORCHESTRATION** :
  - Application or service orchestration is the process of integrating 2 or more apps and services together to automate a process, or sync data in real time.
  - Kubernetes is a tool that allows us to orchestrate processes of an app.
  - All the processes are to be synced together are independent of each other, in general.

- Multiple machines can simultaneously perform the front-end role for a single application.
- By using multiple IP addresses the requests can be sent to different servers.

Date: Dec

PAGE NO.
DATE:

CDN (Content Delivery Networks)

- a good CDN → multiple servers in different geographical locations.

→ These are used to reduce load on servers by hosting static assets like images & scripts.

#

What Is A Platform ?

- A platform is a combination of hardware & software used for app dev.
- It is a collection of guidelines that allow programmers to create software applications using the appropriate technological stack.
- A particular version of OS is a part of software platform.

- It is possible to run multiple front end servers on a single physical server using concepts of parallelism.
- When multiple backend servers are used, database replication techniques must be used to make sure that data is same in all cases.