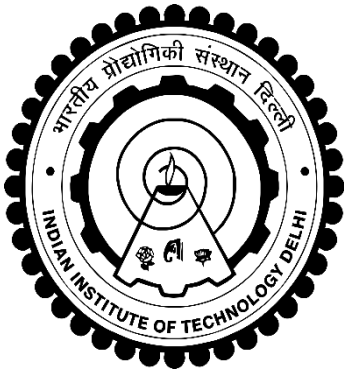
Summer Project Report (2017)

Project Title:

Develop an online System to host capture the flag contest and hackathons

Under

Center of Excellence in Cyber Systems and Information Assurance (CoE-CSIA), IIT Delhi



Duration: 15th May, 2017 to 14th July, 2017

Team Members:

Name	Entry Number
Yash Malviya	2016CS50403
Sumit Kumar Ghosh	2016CS50400
Ayush Patel	2016CS50396
Rudraksh Agarwal	2016PH10549

Supervisor: Prof. Ranjan Bose

Detailed description of project

Capture the Flag (CTF) is a special kind of information security competitions. There are two common types of CTFs: Jeopardy and Attack-Defence.

This website will be an always online CTF platform. We will primarily focus on developing a jeopardy style CTF hosting site and then extend it to other format CTFs (attack-defense) as well.

Jeopardy-style CTFs will have a couple of questions (tasks) in range of categories. For example, Webapp hacking, Forensic, Cryptography, Binary etc. This type of CTF can be either individual or team-based competition. Participants will gain some points for every solved task. More points will be awarded for more complicated tasks. The next task in chain can be opened only after some team or paricipant solve previous task. When the game time is over a score bot will show sum of points to declare a CTF winner.

Attack-defence can be another interesting kind of competition to be included. Here every team will have their own network(or only one host) with vulnerable services. One team has time for patching their services and developing exploits. So, then teams are connected and the war game starts! The teams should protect own services for defence points and hack opponents for attack points.

Our Website will have a leader's board showcasing overall ranking of users and teams which will be sum of points they gained during contests. Users too can submit their own contests.

We will use html with bootstrap for frontend. AngularJs for client side and nosql(MongoDB) for database

Weekly Deadlines for project

Week1: Learning web technology

- Revising HTML, bootstrap(css), Nodejs
- Practicing framework: Emberjs(Frontend)

Week2: Learning web technology (3days) and Learning hacking

- Learning MongoDB
- practicing framework: Sailsjs(Backend)
- Hacking: insectechs courses and video tutorials

Week3: Improving CTF specific hacking skills

- Improving CTF specific hacking skills: reverse engineering, cryptography, web vulnerabilities, binary exercises, networking

Week4: Setting capture the flag problems

- First half of the week: taking in always online CTFs.
- Second half of the week: setting up CTF problems

Week5: Implementation of CTF web application

- implementing frontend by deriving inspiration from famous CTF websites
- programming scoring, hosting system
- configuration of database

Week6: deployment CTF web application and Improvizations

- deploying the website on IITD internal server
- organising a CTF contest

Week1: Outcomes of the project

Further discussion on structure of the project:

Basically we will be using Emberjs framework for frontend, our backend server will have Sailsjs framework integrated with MongoDB (Document based Database Management System).

Starting from basics we will be making a simple emberjs app which which will contain basic MVC elements i.e. models, views, controllers, templates and routers. It will server as the foundation block of our project. Then we will make migrations on the database using the above models.

We will have an admin panel, a host panel and a client panel. Admin panel will take care of development, bug, security etc. related issues of site. Host panel will be hosting challenges and hackathons on the site. Clients may accept these challenges and then work upon them and submit their responses.

Initially following the jeopardy style, the challenges will be stored in the database with their encrypted and hashed solutions(flags). These will be available in site with proper User Interface (UI), from where the clients can download the challenges(concept of jeopardy sytle). There will be a portal where the solutions(64bit strings proving the successful completion) of the challenges can be uploaded by the clients and can be evaluated.

The site will thereby be hosted on a paid web server or on the IIT Delhi Intranet Server whichever be more suitable. If the project is accepted by authorities then we will be having an administration unit for

maintaining the security of the site so that the site itself is not prone to being **“HACKED”.**

Week 1 Foreseen Objective:

- Revising HTML, bootstrap(css), Nodejs
- Practicing framework: Emberjs(Frontend)

Week 1 Objectives Achieved:

- Attained good enough knowledge of HTML and Bootstrap(js and css) from w3school.
- Learned and practiced javascript(from Codecademy and w3school).
- Started learning Nodejs from thenewboston and nodeschool.
- Gained vital knowledge in-built global modules like http and file system.
- Started learning Emberjs, moreover learned to implement custom HTML plugins and js connection libraries and data transfer libraries.

Week2: Outcomes of the project

Week 2 Targets

- Learning MongoDB
- practicing framework: Sailsjs(Backend)
- Learning Hacking: insectechs courses and video tutorials

Week 2 Achievements

Nodejs Practice

- Created a basic express app, which had a simple navigation bar with a few links to webpages of the same app (implemented through fs module). Also learned and used important Nodejs modules such as http, connect, fs.

Emberjs: learning and implementation (continued)

- Continued learning Emberjs framework from Treehouse Tutorials.

- Made an Ember Blog App which had dynamically updated posts and comments feature.

Learning and practicing Sails

- Started learning Sailsjs framework from Sailscasts.
- Created a Demo Sails (backend) app.

Learning MongoDB

- Learnt basics of noSQL and MongoDB from documentation and various other sources.
- Integrated Sails backend server with mongoDB’s database.

Getting more experience in Sails and implementing some necessary features

- Implemented features in a demo app that users will have on the final app. The examples of the features are as follows-
  1. Encrypted password (encryption using Bcrypt Node module) getting stored on user creation.
  2. Basic user sign-up.
  3. Ability to update user information after user creation.
  4. Making the app secure to Cross-Site Request Forgery.
  5. Client side information validation during sign-up with help of jQuery(addition module: jQuery Validate).