DAILY ONLINE ACTIVITIES SUMMARY

Date:	**23-05-2020**	**Name:**	**PRASANNA**
Sem & Sec	**8th,B**	**USN:**	**4AL16CS068**
Online Test Summary			
Subject	---		
Max. Marks	----	**Score**	-----
Certification Course Summary			
Course	**Introduction to ethical hacking**		
Certificate Provider	**Great learner academy**	**Duration**	**6 Hrs**
Coding Challenges			
Problem Statement: prob1- ***To calculate number of uppercase and lowercase letter in given string***			
Status: Solved			
Uploaded the report in Github	**Yes**		
If yes Repository name	**prasanna_p**		
Uploaded the report in slack	**Yes**		

Online Test Details: (Attach the snapshot and briefly write the report for the same)

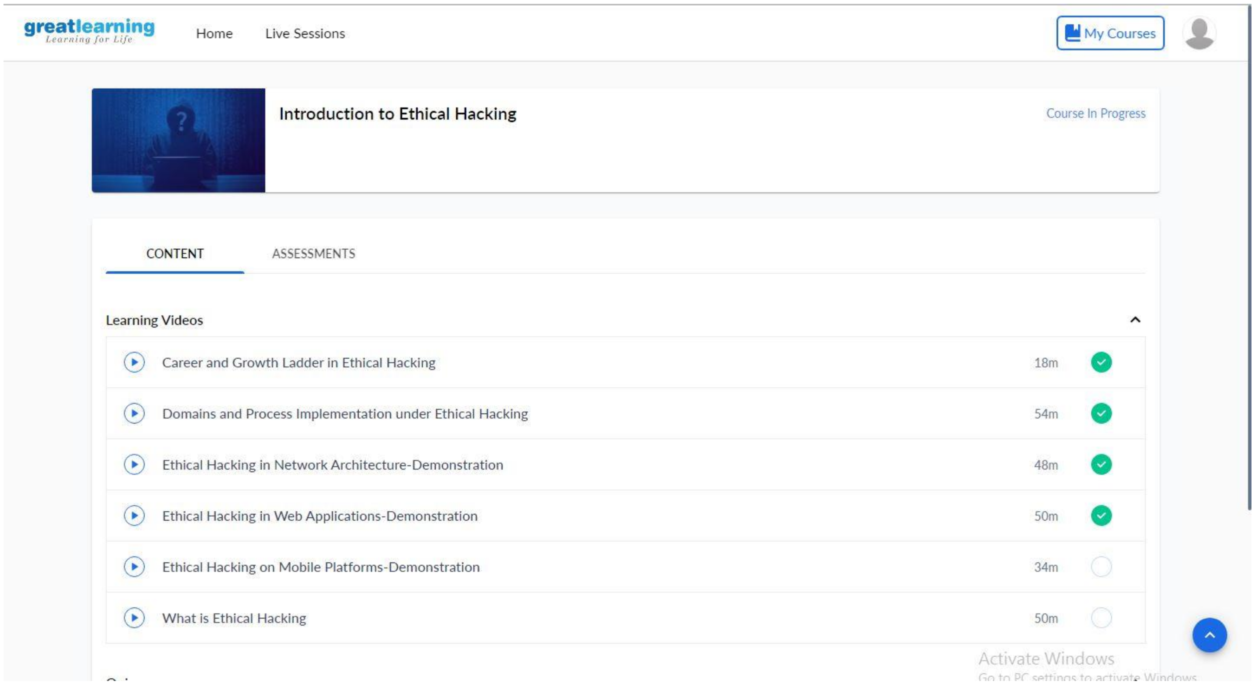
Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same

1. Certification Course Details:

Mobile App pen-testing:

The Mobile Application Penetration Testing Methodology (MAPTM), as described by author Vijay Kumar Velu in his ebook, is the procedure that should be followed while conducting mobile application penetration testing. It is based on application security methodology and shifts the focus of traditional application security, which considers the primary threat as originating from the Internet.



Introduction to Ethical Hacking

Course In Progress

Learning Videos

Career and Growth Ladder in Ethical Hacking	18m	✓
Domains and Process Implementation under Ethical Hacking	54m	✓
Ethical Hacking in Network Architecture-Demonstration	48m	✓
Ethical Hacking in Web Applications-Demonstration	50m	✓
Ethical Hacking on Mobile Platforms-Demonstration	34m	○
What is Ethical Hacking	50m	○

The mobile application penetration testing methodology focuses on client-side security, file system, hardware, and network security. It is has been long considered that the end user is in control of the device.

Stages of the Mobile Application Penetration Testing Methodology:

Discovery requires the pentester to collect information that is essential in understanding events that lead to the successful exploitation of mobile applications.

Assessment or analysis involves the penetration tester going through the mobile application source code and identifying potential entry points and weaknesses that can be exploited.

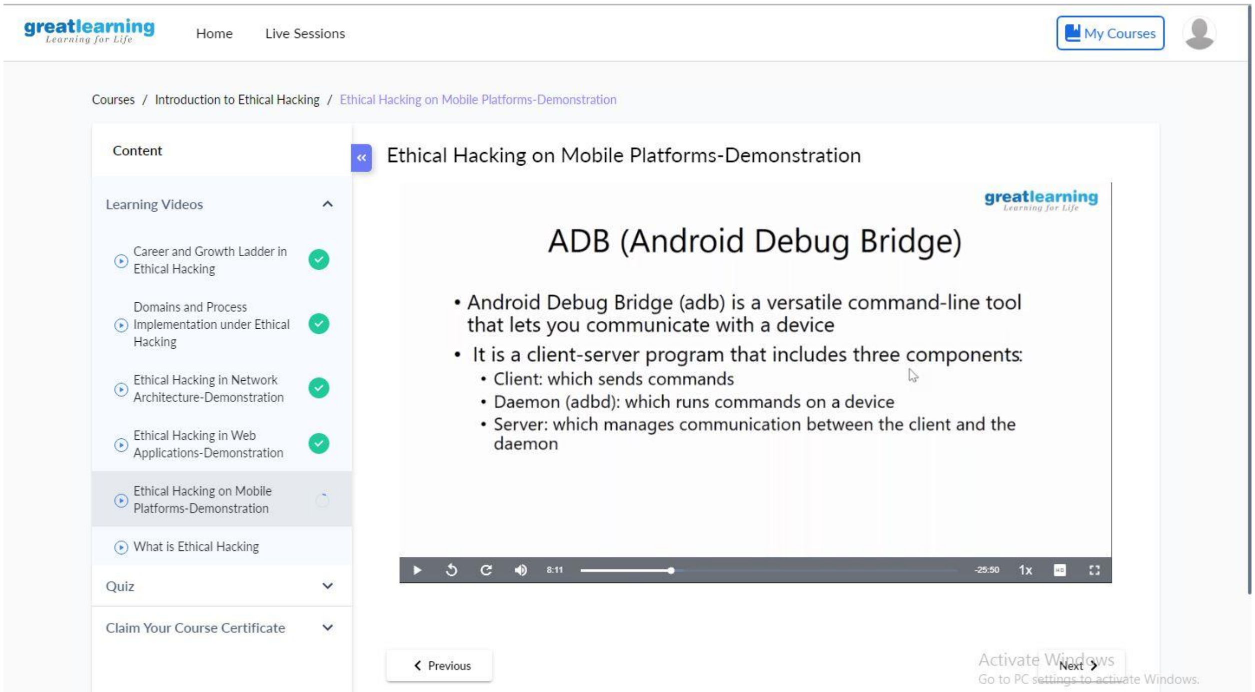
Exploitation involves the penetration tester leveraging the discovered vulnerabilities to take advantage of the mobile application in a manner not intended by the programmer initially did not intend.

Reporting is the final stage of the methodology and it involves recording and presenting the discovered issues in a manner that makes sense to management. This is also the stage that differentiates a penetration test from an attack.

- **Open Source Intelligence (OSINT)—**The pentester searches the Internet for information about the application. This might be found on search engines and social networking sites, leaked source code through source code repositories, developer forums, or even on the dark web.
- **Understanding the Platform—**It is important for the penetration tester to understand the mobile application platform, even from an external point of view, to aid in developing a threat model for the application. The pentester takes into account the company behind the app, their business case, and related stakeholders. The internal structures and processes are also taken to account.
- **Client-Side vs Server-Side Scenarios—**The penetration tester needs to be able to understand the type of application (native, hybrid, or web) and to work on the test cases. The application’s network interfaces, user data, communication with other

resources, session management, jailbreaking/rooting behavior are all taken into account here.

- **Dynamic Analysis—**The pentester reviews the mobile application as it runs on the device. Reviews done include forensic analysis of the file system, assessment of the network traffic between the application and server and an assessment of the application's inter-process communication (IPC).
- **Reverse Engineering—**This involves converting the compiled applications into human-readable source code. The penetration tester reviews the readable code in order to understand the internal application functionality and search for vulnerabilities. Android application source code may be modified once reversed and recompiled. The following tools can be used while conducting reverse engineering:


greatlearning
Learning for Life
Home
Live Sessions
My Courses
Courses / Introduction to Ethical Hacking / Ethical Hacking on Mobile Platforms-Demonstration
Content
Learning Videos
Career and Growth Ladder in Ethical Hacking
Domains and Process Implementation under Ethical Hacking
Ethical Hacking in Network Architecture-Demonstration
Ethical Hacking in Web Applications-Demonstration
Ethical Hacking on Mobile Platforms-Demonstration
What is Ethical Hacking
Quiz
Claim Your Course Certificate
Ethical Hacking on Mobile Platforms-Demonstration
greatlearning
Learning for Life
8:11
-25:50
1x
Previous
Next
Activate Windows
Go to PC settings to activate Windows.

ADB (Android Debug Bridge)

- Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device
- It is a client-server program that includes three components:
  - Client: which sends commands
  - Daemon (adbd): which runs commands on a device
  - Server: which manages communication between the client and the daemon

Android Debug Bridge:

- ADB is a versatile command line tool that lets you communicate with a device.

- It is a client-server program that includes three components:

- Client which sends commands
- Daemon which runs commands on a device
- Server which manages communication between client and daemon

2. Coding challenge:

1. Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.
   Suppose the following input is supplied to the program:
   Hello world!
   Then, the output should be:
   UPPER CASE 1
   LOWER CASE 9

Program:

```
s = input()
d={"UPPER CASE":0, "LOWER CASE":0}
for c in s:
     if c.isupper():
        d["UPPER CASE"]+=1
    elif c.islower():
        d["LOWER CASE"]+=1
    else:
         pass
```

```
print "UPPER CASE", d["UPPER CASE"]
print "LOWER CASE", d["LOWER CASE"]
```