SIT704

ADVANCED TOPICS IN DIGITAL SECURITY

SYAMALA PHANI DEEPIKA CHITTAJALLU

218157961

Summary Report

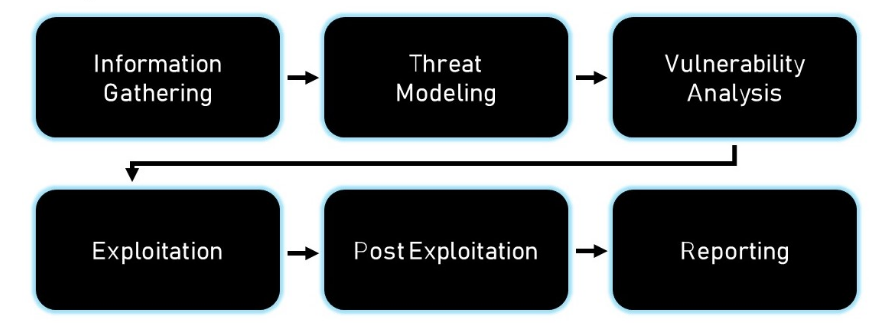
# A Security Testing Report

**Executive summary:**

This report target was to conduct a penetration test based on the OWASP three vulnerabilities listed. The three vulnerabilities under test are cross-site scripting, broken authentication, SQL injection. It was discovered that cross-site scripting enables the attacker to run harmful computer code on a target without the proper authentication. This is very bad in case of sensitive data involved. The broken authentication was found to allow user login to a service without the need to properly identifying themselves and authenticating the same. This poses a key business risk as users who are not in the system can access the data owned by the company. Finally, the SQL injection is a technique in which a code is injected and it is used for attacking applications which are data driven. The first section of the report details the methodology applied in carrying out the penetration test. The second section details the key vulnerabilities found and applying preventions to reduce the risks(Tomasin, 2017)

**Methodology:**

This section details the process the test team undertook to provide the various services of penetration testing of the OWASP technology. The general methodological process is as shown in the flow chart below. The general pen test will be as follows, the methodology, tools and testing procedure differs with the application vulnerabilities.



## **Information gathering**

At this initial phase, the test team gathers as much information as possible about the OWASP to make sure all the services are mapped into the report. This included the enumeration of all the services offered by the OWASP. Once all the services have been modeled, all the accessible systems that connect to the OWASP is listed to get the details of their correlation with the OWASP. The goal of this stage was to ensure the team is fully loaded with as much information as possible(Saadat Beheshti, Liatsis and Rajarajan, 2017).

## 

## **Threat Modeling**

Once the information gathering phase is done, the team conducts a threat modeling on the OWASP to identify key vulnerabilities within the system which is key aspect to the report. The initial steps in this phase involved conducting auto-scans on the network to discover shallow vulnerabilities based on the information gathered in the information gathering phase. Thorough scans were done using deep diving tools that even included writing manual test cases to ensure all the aspects of the system are discovered to ensure all the vulnerabilities are enlisted(Jacobs, 2014)

## **Vulnerability Analysis**

This phase of the pen test involves the documentation and analysis of the vulnerabilities identified in the threat modeling phase. The auto scans and manual test are analyzed to identify critical vulnerabilities. Once the critical vulnerabilities have been identified, they were weighted for further research and analysis. The output of this phase is the actual plan for the attack(Mason, 2012).

## 

## **Exploitation**

This phase involved the actual exploitation of the vulnerabilities identified and analyzed in the previous phases. Vulnerabilities such as the XSS, Broken Authentication, and SQL Injection. The goal of this phase are to be certain that the vulnerabilities identified above are actually exploitable(Marashdih and Zaaba, 2017).

## 

## **Post Exploitation**

This phase involves the analysis of the exploitation done in the preceding phase and details all the findings of the exploitation phase. This enables the team to have a better understanding of the key vulnerabilities discovered during the real exploitation of the vulnerabilities through the threat analysis. The outcomes of this analysis of the findings shall be the key input for the reported phase where the key vulnerabilities and potential remedies can be done to salvage the situation(Kirda et al., 2009).

## **Reporting**

This part of the processes is key as it will entail the actual communication results to the stakeholders. This is achieved by having a summary of the key findings with their priority levels given after a thorough analysis of the threat in the post exploitation review. The team then make recommendations to the various stakeholders on what can be done to reduce the impact of the risks associated with the vulnerabilities.

**Tools and Methodologies used to test our chosen vulnerabilities:**

A computer attack which is authorized and helps to evaluate the security levels of computer can be termed as penetration testing. Any flaw in a web application prompting security misuse can be termed as security vulnerability. These vulnerabilities target integrity and privacy issues of the web application. The chosen tools and operating systems which are used for handling the application vulnerabilities selected by the test team are listed below

**Kali Linux**: Kali Linux is designed for penetration testing in digital forensics with the help of Debian- derived linux OS consisting of 600 programs penetration testing priorly installed.

**Burp suite**: The security of web application can be tested using graphical tool called burp suite. In our project in order to test the tool we will be using the community edition which has several features like HTTP proxy will be provided which acts as a medium between destination server and the web server. We can identify pros and cons in this because destination address can be changed by the attacker.

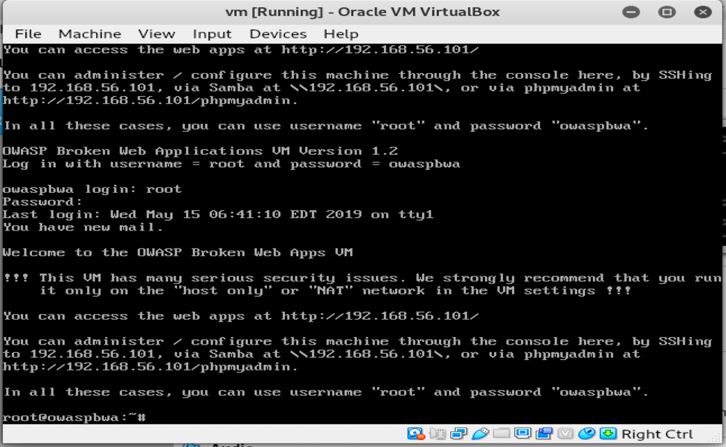
**OWASP Broken web application**: OWASP being an international organization incorporates tools, forums and records for providing open source of information which can be used by designers in creating web application or dealing with security issues related to web application. This application consists of vulnerabilities which are pre defined in context with various websites being exploited. To install this we require Kali linux os and has to be installed with a virtual box and run during the time of website exploitation.

**Detailed Testing Procedure:**

**Cross site scripting:**

XSS has been exploited in this report because it is second in top 10 most prevalent attacks. Here the test team is dealing with Stored XSS where application will store wrong input of the user which is viewed afterwards by admin. The risk factor of this is very high.

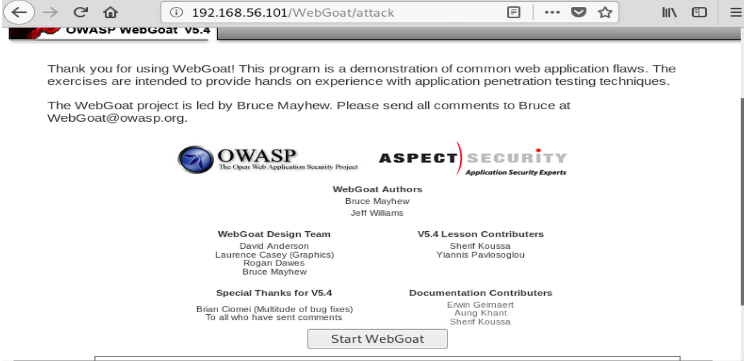
For this penetration testing first step is entering into virtual machine then start our application( OWASP Broken web application). Then we need to enter our login credentials as prompted. The screenshot below shows web application being accessed at 192.168.56.101



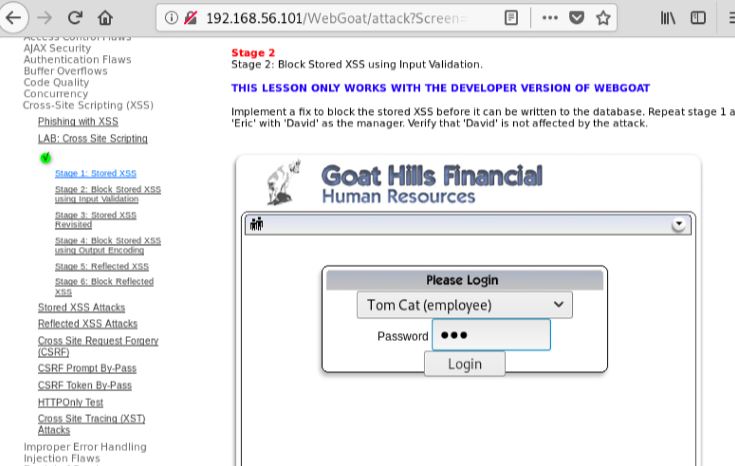
After this we should move to a particular address and check the web page of our application, then find various websites which are vulnerable. This website navigates us to a website named web goat consisting of different vulnerabilities.



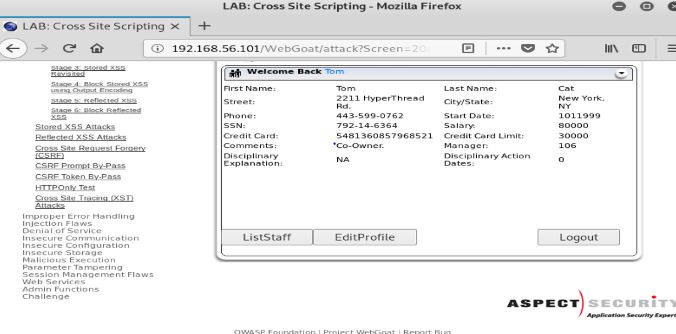
Now we need to start this web goat, on starting this we will get a prompting screen asking login credentials. After providing login credentials ( webgoat/webgoat ) and click the button ‘Start Webgoat’.

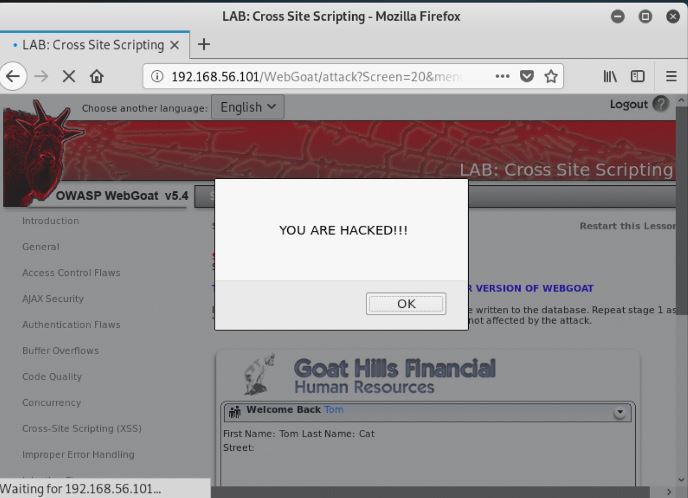


Once we enter into the website we can find different vulnerabilities listed to the left of our screen out of which we need to select one. Since we are working on cross site scripting we will be choosing the same thing here which opens up many more options from that we have selected Stored cross site scripting. We will be asked employee login credentials on the right side of the screen. For which we can get the required password with trail and error.



After entering into the employee page we will have permission of altering anything we like i.e. changing employee details which indicates vulnerabilities of websites. With the help of javascript code even details of employee can be altered and on pressing save we can get a message every time we try to access employee record



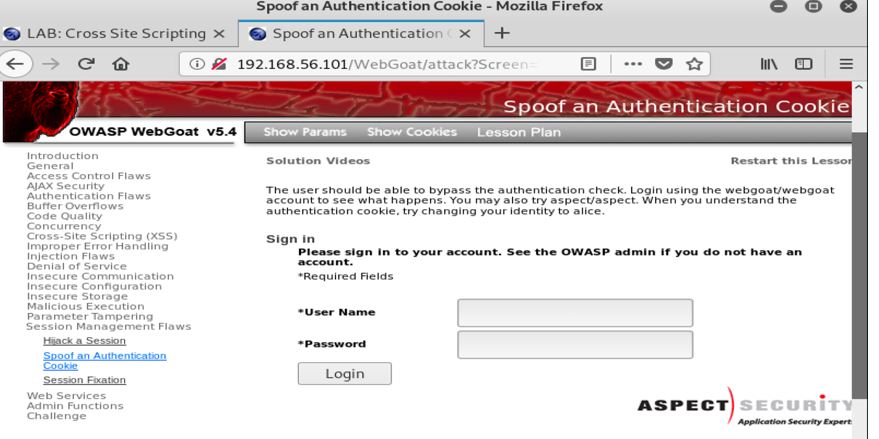


From this we can observe successful incorporation of XSS in a vulnerable website.

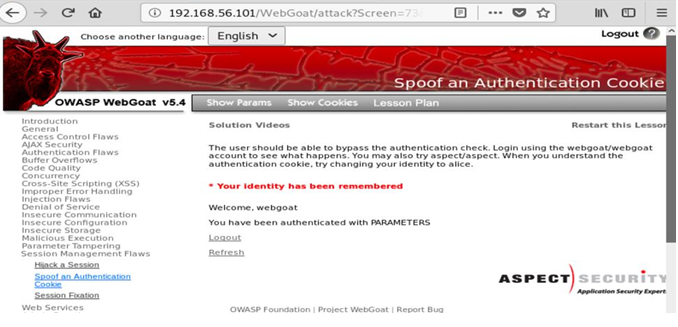
**Broken authentication:**

This is kind of vulnerability in which attacker can capture or pass authentication steps which website uses.

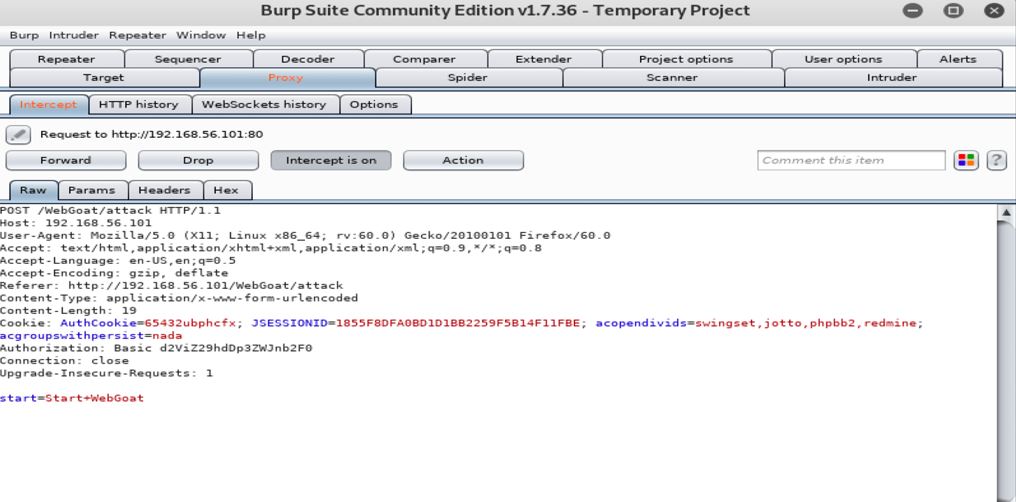
In order to perform penetration testing the test team have opened Webgoat and selected session management flaws in it and further selected spoof of authentication cookie.



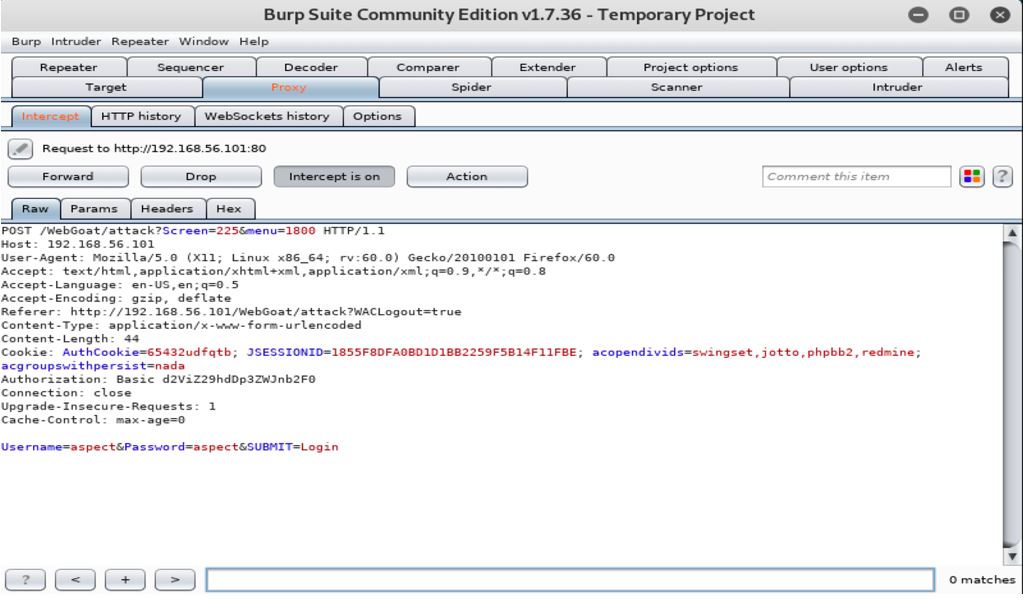
Now enter webgoat login credentials( webgoat/webgoat or aspect/aspect) which are mentioned in the screenshot and click on login button which gives us following message



For finding Authcookie for aspect and webgoat we need to move to burpsuite as shown below:



We can identify webgoat’s Authcookie as 65432ubphcfx and that of aspect as 65432udfqtb. In both these cookies prefixes are same and remaining part is different. By trial and error we have concluded that word is in reverse format and the alphabets are swapped with preceding alphabets it are written i.e. webgoat can be reversed as toagbew now after swapping with preceding alphabets we get ubphcfx. Which states that if we want to enter as a random user all we need to do is change the corresponding cookie value in similar pattern.



**SQL Injection:**

Kali Linux is the tool which is used for hacking the login credentials of SQL Injection i.e. OWASP, sqlmap, terminal and Mozilla Firefox in kali. Techniques which can be used for attacking with sqlmap are banner parsing fingerprint, HTML error message finger print and active finger print. Attacks which are threat for sql injection are those which allows attacker to modify information or details, identify theft and causes renouncement issues.

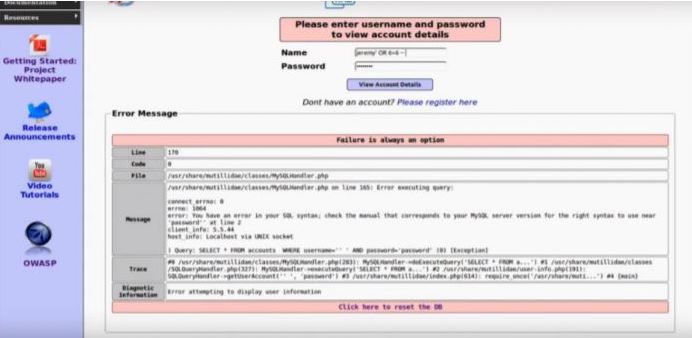
We will be executing an automatic tool based on open source command line in a kali linux operating system. Initially kali linux is to be opened and run sqlmap in it in order to execute sql injection tool using simple commands. Test team then opens the terminal and enters virtual box, starts scanning which will display the IP address. Then sqlmap is to be opened and give comments for hacking the login credentials through sqlmap python sqlmap.py-u <http://www.site.com/section.php?id=51>. Once this command is executed tables are identified in the database with the command –u <http://www.site.com/section.php?id=51> –tables –D safecosmetics and then data is extracted.



Then Mozilla firefox is opened in kali linux OS and enter 192.168.56.101, further clicking on multillidae gives the following

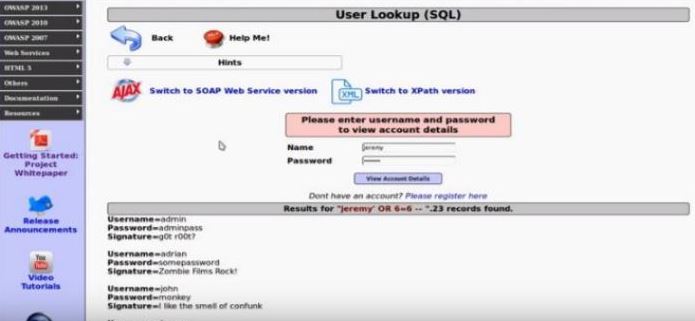


Now after entering valid login credentials and giving details to view account an error message with few queries will be displayed which will be injected into sql. After that we need to check in context for terminals in command as tmp/gedit/x. Further inject the error message’s query into firefox as shown



Query injection into terminal is done as follows

Select \* from accounts where username = ‘’ AND password = ’password’, now in order to hack all login credentials it is ‘select \* from accounts’ which gives a result of all login credentials.



With SQL injection every user detail can be dumped into tables with various commands.

**Vulnerability Analysis:**

**Broken Authentication:**

This is caused by poor session management which make the session data of user not protected hence could be easily compromised. Broken authentication can be exploited in the following ways; first, the attacker can impersonate an authorized user and perform user privileged actions on the system. This could be devastating especially when the user is a super user with critical permissions attached to it.

### Recommendation:

The following are some of the recommendations the test team has put forward to curb the exploitation of the broken authentication.

First, there is a need to improve the password length to 8 alphanumeric characters with the inclusion of special character to curb brute force attack chances of succeeding.

Second, there is a need to enforce account disabling after some number of attempts to login failed. The period of account disable should be sufficient to ensure brute force don’t succeed in gaining the user authentication credentials.

Third, there is a need to have Username/Password enumeration and failures on the authentication should not tell the user which part has failed but rather be vague as output invalid username and or password.

**Cross site scripting:**

This happens when a malicious user injects some scripts into the web application input framework. This scripts can be used to cause harm to the production database. The vulnerability can be used to steal information from the issue by faking for them some pages that may look familiar with their regular websites. This makes user data such as credit card information put in jeopardy. Handling accounts, session stealing, Trojan login panels, attacks on user browser and other client side attacks are also involved in cross site scripting.

### Recommendation:

Separating fraud data from user browser with the help of frameworks which helps to escape XSS with designs like react JS, Ruby on Rails. Knowing the limitations of this framework and using them in appropriate manner which are not mentioned in use cases.

**SQL Injection:**

This is a very common technique used by the attackers because it provides data base access to the attacker. SQL injection is nothing but putting fraudulent statements of sql with the help of web pages.

**Recommendation:**

In order to prevent this attack and minimise the attack on actual data base we need to create a duplicate database and get inputs to this database, data and input validation must be regularly sanitized.

# **Conclusion**

The report has given the various test to be conducted with regard to the OWASP vulnerability list. Three of those selected includes the XSS, broken authentication and SQL injection. The test team developed a methodology which involved steps of the whole pen test carried with each phase having its own output. The steps include information gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation and finally compilation and presentation of the report. More care still needed to be done to ensure the human elements and the physical security of the various key information resources are protected. This will increase the security of the resources.

**References:**

1)Tomasin, S., 2017. Analysis of Channel-Based User Authentication by Key-Less and Key-Based Approaches.

2)Saadat Beheshti, S.M.R., Liatsis, P. and Rajarajan, M., 2017. A CAPTCHA model based on visual psychophysics: Using the brain to distinguish between human users and automated computer bots. *Computers & Security*, 70, pp.596–617.

3)Mason, A., 2012. Caught in the cross-site scripting fire. *Network Security*, 2012(5), pp.5–9.

4)Jacobs, F., 2014. Providing better confidentiality and authentication on the Internet using Namecoin and MinimaLT.

5)Marashdih, A.W. and Zaaba, Z.F., 2017. Cross Site Scripting: Removing Approaches in Web Application. *Procedia Computer Science*, 124, pp.647–655.

6)Owasp.org. (2019). *Top 10-2017 Top 10 - OWASP*. [online] Available at: https://www.owasp.org/index.php/Top\_10-2017\_Top\_10 [Accessed 17 May 2019].

7)YouTube. (2019). *OWASP Broken Web Applications VM*. [online] Available at: https://www.youtube.com/watch?v=cwjcfAgKqcg [Accessed 17 May 2019].

8)Linuxhint.com. (2019). *SQL Injection with Kali Linux – Linux Hint*. [online] Available at: https://linuxhint.com/sql-injection-kali-linux/ [Accessed 17 May 2019].

# Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pass (D) | Credit (C) | Distinction (B) | High Distinction (A) |
| Self-Assessment |  |  |  |  |

Self-Assessment Statement

|  |  |
| --- | --- |
|  | Included |
| Learning Summary Report |  |
| Pass tasks complete |  |

Minimum Pass Checklist

|  |  |
| --- | --- |
|  | Included |
| All Credit Tasks are Complete on Ontrack |  |

Minimum Credit Checklist (in addition to Pass Checklist)

|  |  |
| --- | --- |
|  | Included |
| Distinction tasks (other than Custom Program) are Complete |  |
| Custom program meets Distinction criteria | NO |

Minimum Distinction Checklist (in addition to Credit Checklist)

# Declaration

I declare that this portfolio is my individual work. I have not copied from any other student’s work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: **SYAMALA PHANI DEEPIKA CHITTAJALLU**

# Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for SIT704 Unit Title to a **Distinction** level.

I really worked very hard in completing the tasks. In addition to self learning, the practical sessions which we attend weekly helped me to solve the queries with the tutor and complete tasks on time. I’m still working on few of the tasks ,based on my knowledge and understanding of concepts I can grade myself to distinction level. I have learned many topics in this unit especially the Linux authentication and hacking concepts gave me handful of knowledge. Implementing Linux authentications, Using kali Linux, Metasploit, cracking passwords, OWASP burp suite, SQL has given me hands of experience in ethical hacking environment of overall cyber security.

# Reflection

## The most important things I learnt:

Topics like Linux authentication, Default crypt, Basic linux commands, implementing cvss scores, OWASP exploration, Setting up ethical hacking environment, scanning websites topics has helped me in gaining knowledge in this course.

## The things that helped me most were:

Implementation of encryptions , Default crypt function, Crypt hash in linux authentication, Setting up ethical hacking environment and OWASP vulnerabilities were the things that helped me most

## I found the following topics particularly challenging:

The topic which I felt very challenging was Setting up ethical hacking environment because this task requires more patience and there should be no error occurred.

## I found the following topics particularly interesting:

Exploring OWASP three vulnerabilities and analyzing them in detail is very interesting for me.

## I feel I learnt these topics, concepts, and/or tools really well:

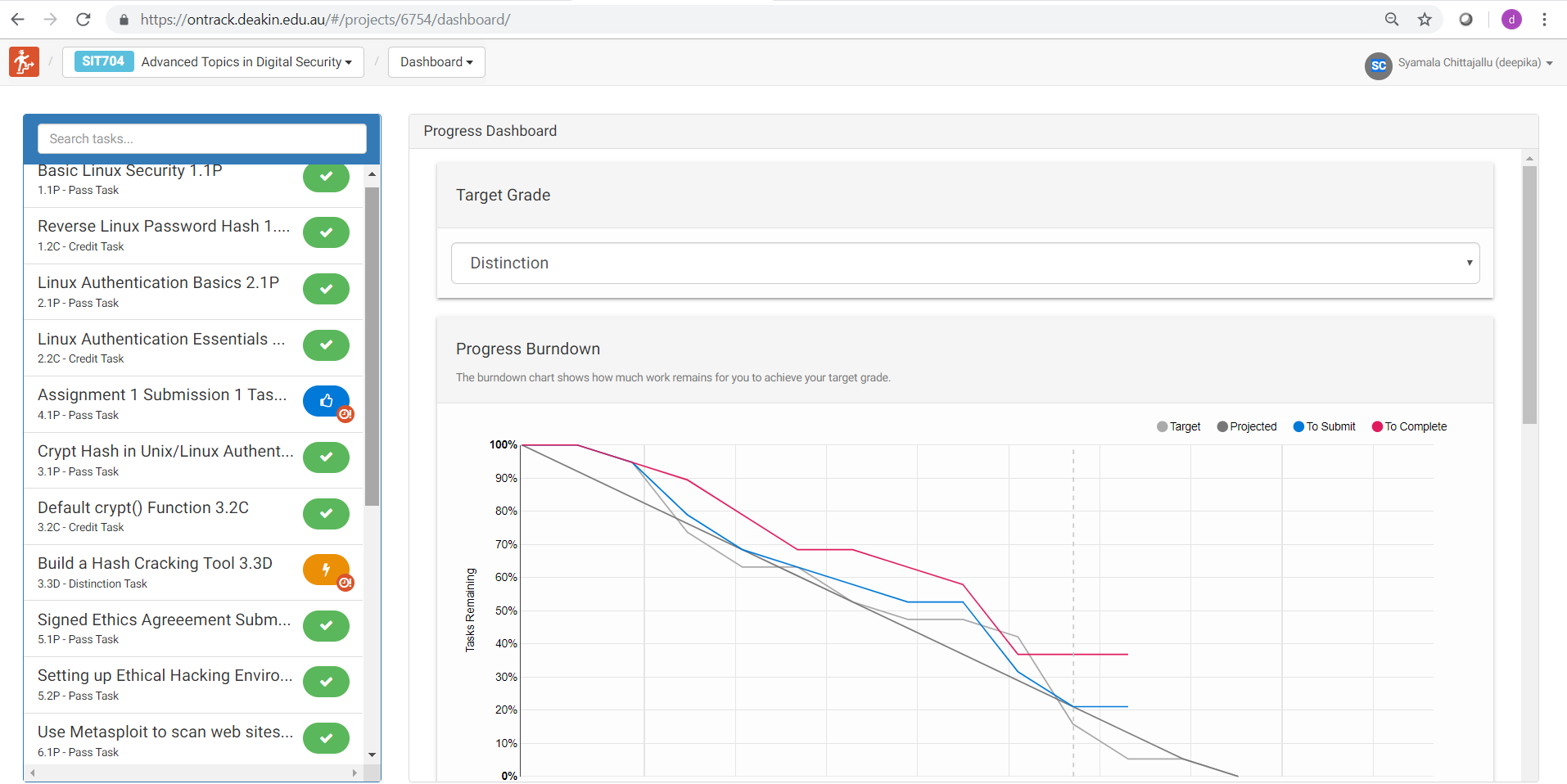
Using putty, Kali Linux, OWASP three vulnerabilities, Penetration testing of different web

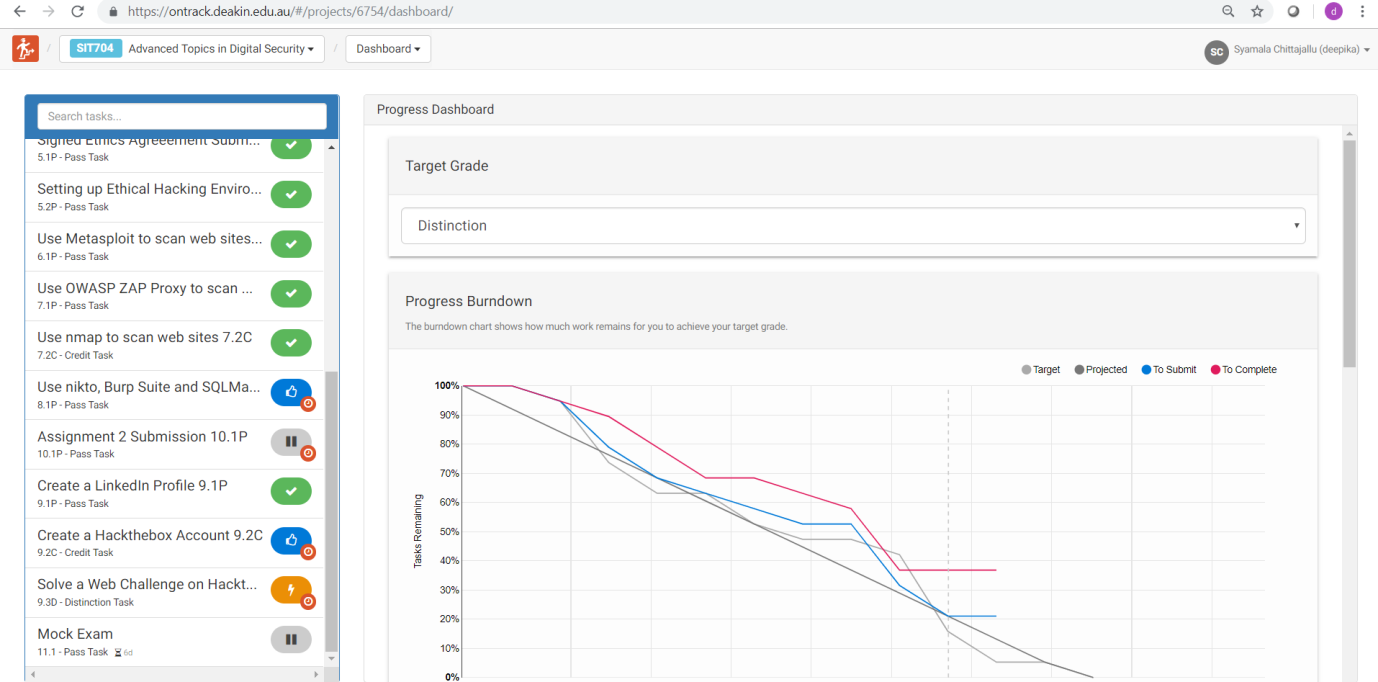
applications are the tools and topics that I feel learnt really well.

## I still need to work on the following areas:

I still need to work on Build a hash cracking tool and working on Solve a Web Challenge on HacktheBox.

## My progress in this unit was …:





## This unit will help me in the future:

This unit has given me in depth knowledge regarding awareness of security issues and to overcome the hacking attacks.

## If I did this unit again I would do the following things differently:

If I would do this unit again I will try my best to solve high distinction tasks and update my knowledge accordingly and explore the topics in various ways.

## Other…:

I have enjoyed doing this unit which gave me real time experience in exploiting the vulnerabilities. By doing this unit I came to know about many security measures we should take in this rapid development of cyber attacks.