

Web Application Penetration Testing Using Burp Suite

Navjot Kaur Tanushree Das
Department of Electrical and Computer Engineering
Western university London, Ontario, Canada
nkaur23@uwo.ca tdas4@uwo.ca

Abstract—In recent years, due to the use of internet by government, companies etc, cyber security has become one of the most crucial and important aspect. Everyone is afraid of the word “hacking” during credit card transactions, sharing confidential information using internet etc. As an ethical hacker, the automated tools of penetration testing could be helpful to recognize the various flaws that occur during the deployment of web application.

This paper is a humble attempt to demonstrate the vulnerabilities of web applications using a penetration testing tool i.e. burp suite. We have done different experiments using demo sites, to analyze the traffic, packet transmission etc. The motive of the experiments is to show how the vulnerability testing coverage could be increase by using penetration tools.

Index terms--- testing security, vulnerabilities, hacking, burp suite

I. INTRODUCTION

Web services have become very important in information technology (IT) based services, industrial devices etc. Insecure web applications lead to have various security flaws. OWASP top 10 report [1] revealed the major threats which include cross site scripting, SQL injection, sensitive data exposure etc. Every web application has its own exploitable vulnerabilities. For example, the new reports show that word press and some other PHP applications are vulnerable to the GHOST Linux exploit [2]. Table 1 shows the overview of web services specific attacks.

A hacker is a person with a brilliant mind who can exploit the security of an application using different methods and can use the information either for good or bad. The motive of Penetration testing is to use the automated tools for finding the flaws in the web application so that the application will become make more secure. Although, none of the security tool is entirely complete in nature to identify the risks in web application, but it is possible to achieve rigorous vulnerability assessment on the basis of pool of functionality. [3]

Xml signature wrapping	Coercive parsing
Oversize payload	XML injection
Attack obsfunction	Instantiation flooding
Indirect flooding	Middleware hijacking
Oversized cryptography	Metadata spoofing
WsdL scanning	SOAPAction spoofing

Table 1. Overview of web services specific attacks.

We have used burp suite as a penetration testing tool. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities. Some of its feature are an intercepting proxy, an application-aware spider, web application scanner, an intruder tool is used for performing the Powerful customized attacks to find and exploit unusual vulnerabilities.

In this paper, we have performed the set of experiments using a penetration testing tool. Section III provides the related search associated with the web application vulnerability assessment. Section IV provides the experiments and their results that we have conducted for the vulnerability assessment. In Section V and section VI, the limitation and conclusion of the work is discussed respectively.

II. RELATED WORK

There are number of different penetration testing tools are available with different features and functionalities for inspecting the security aspects of web application. Shay Chan [5] compared 60 commercial and open source black box web application vulnerability scanners. The author just focused on the utilization of test tools instead of covering the wide range of vulnerability test cases. Rohan et al [3] provided a testing approach for vulnerability assessment of web application by means of analyzing and using combined set of tools to address a wide range of security issues, as shown in figure 1.

Christian et al [4] just evaluated different Web Services frameworks and their resistance against WS-Addressing spoofing and SOAPAction spoofing attacks, as they should also provide some solution for their research. Nuno Antunes et al[6] suggested that the penetration testing results are not necessary to be correct. Their study shows that the chances of getting false positives are always there. The limited search is the biggest limitation of their research due to which we cannot say if testing tools would show some false positive for other type of vulnerabilities also.

Most of the researchers agree with the fact that hacking itself is not a crime if we are not using it for the wrong purposes. To know the mindset of the black hat hackers, it is necessary to know that ways they have used to do it. To achieve this, different kind of tools and processes are used by the ethical hackers.

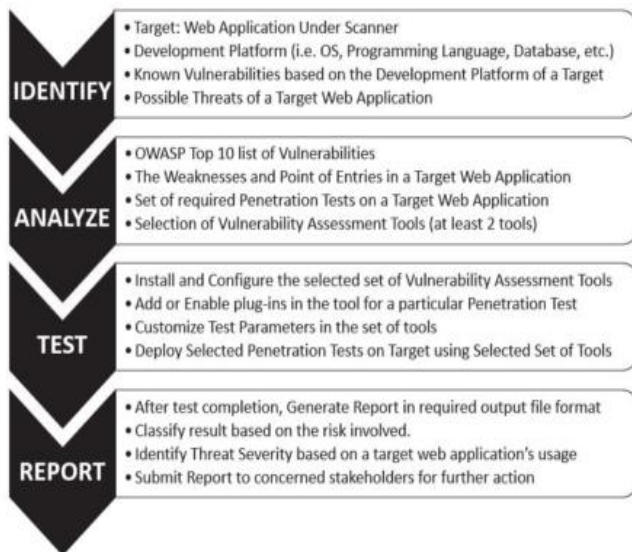


Figure1. Vulnerability testing approach [3]

III. EXPERIMENT AND OBSERVATION

In this section, we provide the description of the experiments performed by us for penetration testing using the modules of Burp Suite. We will first explain the process and then discuss the results on some commonly used Web Applications and some demo sites.

A. Experiment 1: Intercept Browser traffic

For performing this experiment, we have chosen to use Damn Vulnerability Web App(DVWA) [13] and XAMPP [14] to run DVWA. Burp Suite will be used as 'proxy-server'. First we run the Burp Suite and under the 'options' panel in 'Proxy' we will find the IP address and the port number. Now,

```

POST /dvwa/vulnerabilities/brute/ HTTP/1.1
Host: 127.0.0.1
Content-Length: 88
Cache-Control: max-age=0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://127.0.0.1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/50.0.2661.75 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://127.0.0.1/dvwa/vulnerabilities/brute/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: security=impossible;
PHPSESSID=fkpqeb1ij7ubmhtpjdj04jh2j1
Connection: close
username=admin&password=password&Login=Login&user_token=3d75da984067c2aed3f2968f7c618ece
  
```

Listing 1. Intercepted HTTP Request

to set up Burp Suite as proxy (Chrome Browser), we have to enable 'Proxy server' in LAN settings and provide the IP address and the port number of Proxy server. After this we can go to DVWA's Brute Force Module, enter the Username and Password. Once set up is complete we can find the HTTP request under the 'Intercept' panel of Burp Suite [9] as shown above in Listing 1. Also to intercept HTTPS traffic we have to import burp suite certificate [17].

Now, after going through the basic details we intercepted HTTPS traffic from commonly used web sites i.e Stack Overflow [20] and concluded some of the observations as listed:

1) Stack Overflow Login Request and Response

The HTTPS request provided details of the Login step:

- 1.1) The method used which in this case was 'POST'.
- 1.2) Cookie details such __cfduid,__qca etc.
- 1.3) Details whether the HTTPS request is for Signup request or Login.
- 1.4) User Id and password entered by the user.
- 1.5) The login protocol which was OAUTH 2.0 in this case.
- 1.6) Request string i.e the login Verification end point.

The HTTPS request provided details of the Login step for invalid Login credentials:

- 1.1) Server details, Date and time, content Length etc.
- 1.2) Response Code 200 OK.
- 1.3) A message: Login-OK.

The HTTPS request provided details of the Login step for valid Login credentials:

- 1.1) Server details, Date and time, content Length etc.
- 1.2) Response Code 302 FOUND.
- 1.3) Set-Cookie: fkey, uauth, domain, expiry time etc.
- 1.4) Location which was http://stackoverflow.com/
- 1.5) A message: Object moved

2) Observation

The key difference we could see in the Response code was 200 for invalid credentials and 302 for valid credentials and also there was difference in a message shown to user.

Therefore, we can conclude from this experiment that a hacker or malicious user can intercept browser traffic and can find out many relevant information after analyzing the traffic. In Stack overflow the website provided some warning that 'request appears to be suspicious' which can alert the user for taking necessary action.

3) Encrypted Request and Observation

After stack overflow [20] we later intercepted a leading bank's login request, it was much secure with only the user id visible the password was encrypted. It appears either SSL or some kind of encryption mechanism was being followed.

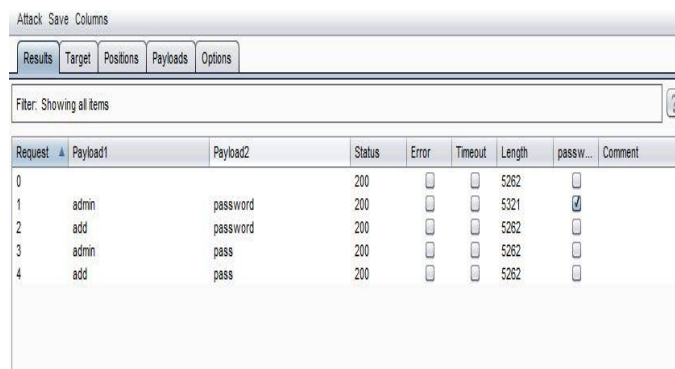
Also we got to know about:

- 1) `__cfduid` cookie [11] : It is used override any security restriction on the basis of IP Address. For example, if the user is performing Login from public place which can consist of infected machine, but if his machine is trusted the security issues can be overcome by this cookie.
- 2) `__qca` cookie [12]: It collects information like the IP address, HTTP location etc. for Data Analytics Purpose.

B. Experiment 2: Brute-Force Attack

After intercepting browser traffic, the next step can be to use the parameters present in the request and perform 'Brute-Force' attack to find out the valid Login Credentials or other Details. This experiment will require the use of the 'Intruder' module of the Burp Suite [13].

First we right click on the intercepted request in our previous experiment and then click on the option 'Send it to Intruder'. In Intruder panel, we select the attack type as 'Cluster bomb' and select the username and password as parameters through 'Add' button. Now under the "Payload" tab, we select the payload number and enter a list of username and password in the "Payload Options" respectively. Finally, after setting up, we click on the 'Start Attack' button.



The screenshot shows the Burp Suite Intruder tool interface. At the top, there are tabs for 'Results', 'Target', 'Positions', 'Payloads', and 'Options'. Below these is a filter bar that says 'Filter: Showing all items'. The main area displays a table with the following columns: Request, Payload1, Payload2, Status, Error, Timeout, Length, passw..., and Comment. The table contains five rows of data, all with a status of 200.

Request	Payload1	Payload2	Status	Error	Timeout	Length	passw...	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	5262	<input type="checkbox"/>	
1	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	5321	<input checked="" type="checkbox"/>	
2	add	password	200	<input type="checkbox"/>	<input type="checkbox"/>	5262	<input type="checkbox"/>	
3	admin	pass	200	<input type="checkbox"/>	<input type="checkbox"/>	5262	<input type="checkbox"/>	
4	add	pass	200	<input type="checkbox"/>	<input type="checkbox"/>	5262	<input type="checkbox"/>	

Figure 2: Brute Force attack

The Username and Password combination will be tried by the Intruder tool and we can examine the process results in the attack window as shown in Figure 2. The request and response of the Attack can be used to analyze whether we have cracked the correct Login Credentials.

Now we will be using 'Brute-Force' attack on Stack Overflow. [20]

1) Stack Overflow Request and Response

As discussed above, after intercepting the request we can manipulate and send request to server from burp suite 'Intruder' to analyze and see if we can find out the valid credentials. We performed this experiment with only two sets of Username and Password for Payload and used the 'Cluster Bomb' Module of 'Intruder'.

2) Observation

After going through the HTTPS Request and Response we observed that three of Login request, response status was '200-Ok' but for one of the pair response was '302-Found'. So as per our observation in Experiment 1, '302-Found' is displayed when the Login is successful. Therefore, we came to conclusion that the valid Login Credentials were cracked through this experiment.

3) Limitation

This experiment however has several limitations, as we did Brute Force attack only with four sets of Username and password to prevent any sort of attack to the actual site, therefore we cannot make any observation about if there is a threshold when the rate of Login request from a particular IP exceeds, and in case there is a threshold what are the actions taken by the site such as permanent blocking or blocking for some duration.

C. Experiment 3: Overcoming Client Site Validation

This is very small and simple experiment based on the work "Manual Security Testing Using Interception" by Ahmed Ibrahim [17] to check if we can overcome the client site validations by modifying request from burp suite and send it to server.

Client site validations are very much useful for web application where first level of input is checked at the client browser itself before sending any anonymous request to server. Burp Suite can be used here to manipulate the request and check if we are able to pull out any information by from the Server or can observe any reaction from server side. First we have to intercept the browser request using proxy and then manipulate the desired input field and then use the 'forward' button to send request to server and observe the response.

1) Zero Bank Request and Response

We have performed this experiment on the demo banking site [14] as per the reference [17]. After login into the site (credentials: username=username and password=password), we go to the option 'Add New Payee' under the tab 'Pay bills'. While entering the details if we leave the 'Payee Name' blank and click on 'Add', then the request will not proceed further. To check for the vulnerability, we will intercept the request through Burp Proxy and make the 'Payee Name' field blank and then forward the request to server as shown in Figure 3.

The request without the 'Payee Name' was accepted by the Server and Success message was shown in browser.

2) Observation

Now after going through the demo site we performed this experiment on Stack Overflow [20] signup and a bunch of other live websites, but we were unable to overcome client side scripting proving that these web applications has, well defined server side validation.

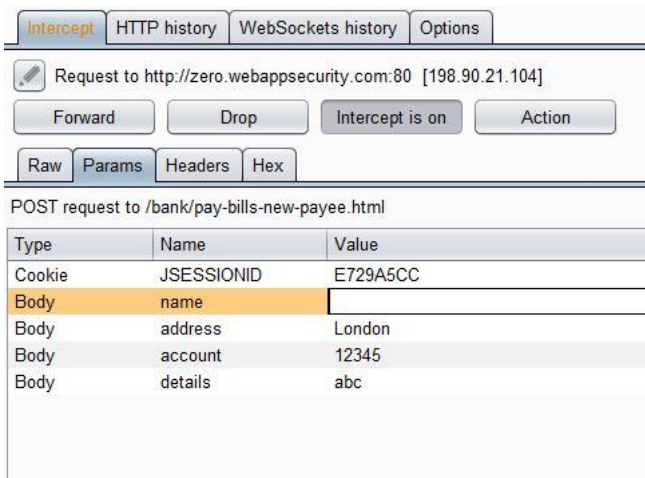


Figure 3: Forward Blank field request

D. Experiment 4 : Modified Request to server

This segment would require the use of burp suite's Repeater [22] module. After intercepting the request and sending to Repeater module we can change some of the request parameters such as cookie or username and observe the corresponding server response and check if we can find any vulnerability or restricted information from the server. Figure 4 shows request response data of an intercepted request in Repeater module.

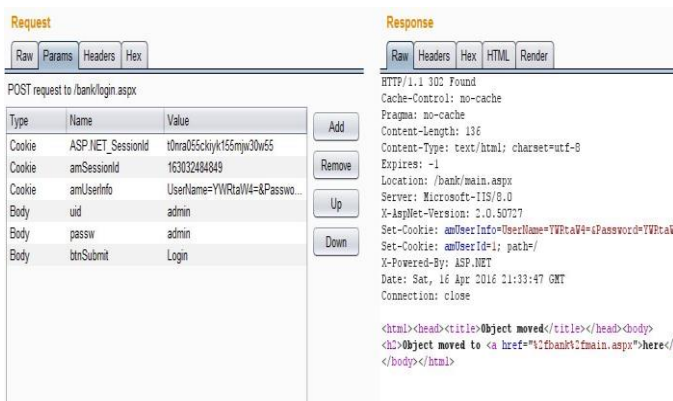


Figure 4: Repeater Tool to inspect Request and corresponding Response

This experiment has been performed on Faking News website. [21]

1) Faking News Login Request and Response

After intercepting the request, we forwarded the request to repeater and then tried to change some of the existing field such as user name or password.

2) Observation

When we removed the username and forwarded the request to server and observed the response as shown in Figure 5. The

response was as expected i.e. registration was unsuccessful with an error message which means the site is properly tested.

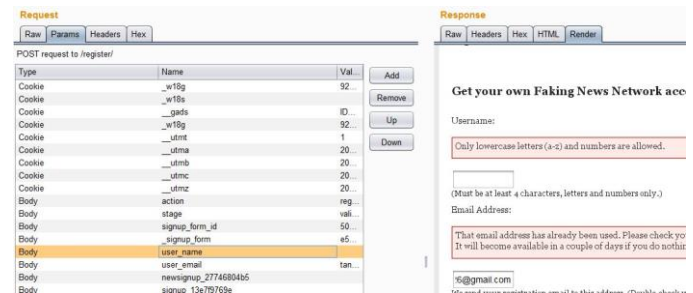


Figure 5: Request and Response with Blank Username

After this we tried to change to a valid username and forwarded the request to server, this time we were able to register successfully (Figure 6).



Figure 6: Request and Response with valid username.

3) Conclusion

Although this experiment was conducted at a minimal level of changing username during registration, in case some financial application is not properly tested and reported to developer, an intruder can intercept the request and change any of the field and drive the request to his favor.

E. Experiment 5: Analyzing randomness of Session Token/Cookies

Intruder can impersonate a token or cookies that are often generated by web application to identify or provide authority to a user [23]. The burp suite's Sequencer module can be used to indicate the randomness and reliability of this generated token. We performed our experiment on Western student Centre account [24] in which we analyzed the randomness of PS_TOKEN [25]. PS_TOKEN cookie provides free navigation within the application without the authenticating again and again.

1) Western Student Central Request and Response

After intercepting the request in the proxy server we have to send it to the sequencer module (burp suite) and select the desired token to be analyzed and then click on the start 'Analyze Now' button. The result of the PS_TOKEN analysis as per burp suite is given in Figure 7 below.

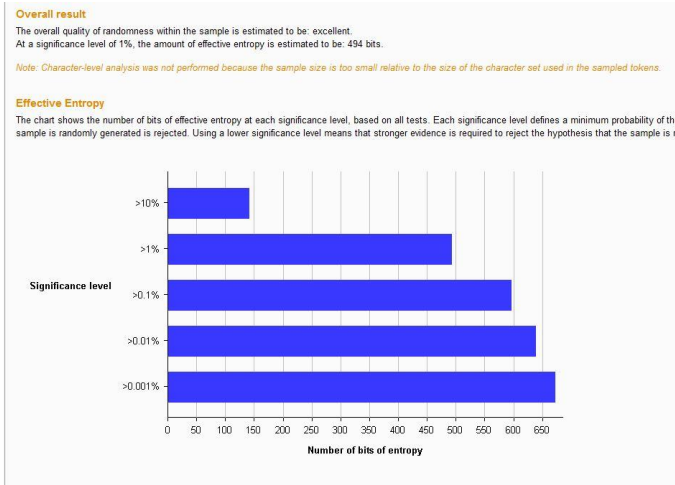


Figure 7: PS_TOKEN Randomness

2) Observation

As we can the overall randomness is indicated to be excellent for the PS_TOKEN within 100 samples requested from the tool. We can conclude that the mechanism behind this token generation is reliable and efficient. Sequencer can be an effective tool to determine efficiency of token generation so that it can be prevented from the attack of intruder.

IV. LIMITATIONS OF PENETRATION TESTING

Penetration Testing only performs black box testing of a web application, without going deep into the organization's network. If the penetration testing is not able to recover any vulnerability that won't classify the application as secure [14]. Any change in the internal source code can result in changes in Request and Response formation, resulting change in intercepted traffic. Penetration testing of a web application is also limited by the fact that it performs only the known exploit already present in public and may be they cannot think of an attack from hacker's point of view [16]. Due to limited time and resource allocated to penetration testing, focus is often given to fixing critical vulnerability than that of low priority vulnerability which can be intelligently used to create a complex scenario and break into the organization's network by intruders [18]. Another serious limitation is that penetration testing is often performed at the end stages of software development life cycle making it difficult for coders to change their design and coding styles and logic [19].

V. CONCLUSION AND FUTURE WORK

Penetration Testing largely benefits the software developers and quality management team before moving the software product or application to production. This project was undertaken to go through the Penetration Testing fundamentals and analyze security aspects of various Web Applications. During our experimentation we learned many features of Burp Suite tool and HTTP/HTTPS Request/Response. We carried the experiments on various

demo vulnerable site and actual sites used in our daily lives. The experiments using Burp Suites provided a deep insight into web application security and vulnerability. While conducting the experiment we realized very well that how important is to sanitize the request and response in the server side. A slight mishandling in this area can be exploited by the intruder to perform various attacks such as Man in the Middle Attack, Denial of Service attack or even get the valid credentials for the trapped user. Now, after successful completion of this project we would further like to perform more powerful penetration attack and at the same time as a software developer learn how we can detect and prevent these attacks at the server side. With keeping in mind the benefits of penetration testing, focus should also be given on selecting the tool for performing the test, a large variety of open source tools are present in the market, selecting among them can be a tedious task for the organization. We have used Burp Suite tool to perform our interception of request and response, we would further like to intervene with other open source tools available for our experimentation.

VI. REFERENCES

- [1] Top 10 2013- Top 10 , [Online], Available : https://www.owasp.org/index.php/Top_10_2013-Top_10
- [2] Ghost Linux bug update: WordPress , other PHP applications vulnerable,[Online],Available: <http://searchsecurity.techtarget.com/news/2240239231/GHOST-Linux-bug-update-WordPress-other-PHP-applications-vulnerable>
- [3] U. Ylekdqgln, L. Q. Dee, D. Vlv, R. I. Wkh, Z. H. E. Dssoldwlrq, and Y. Dvvhvphqw, "Vulnerability assessment web applications- A testing approach," pp. 16–21, 2015.
- [4] C. Mainka, J. Somorovsky, and J. Schwenk, "Penetration Testing Tool for Web Services Security," 2012 IEEE Eighth World Congr. Serv., pp. 163–170, 2012.
- [5] S.Chen,,"The scanning legion: Web application scanners accuracy assessmen t& feature comparison commercial & open source scanners",[Online],Available: <http://sectooladdict.blogspot.ca/2011/08/commercial-web-application-scanner.html>
- [6] N. Antunes and M. Vieira, "Penetration Testing for Web Services," pp. 30–36, 2014.
- [7] Damn Vulnerable Web Application (DVWA), [Online], Available: <http://www.dvwa.co.uk/>
- [8] XAMPP, [Online], Available: <https://www.apachefriends.org/index.html>
- [9] Using Burp Proxy ,[Online] , Available : https://portswigger.net/burp/help/proxy_using.html
- [10] Burp Suite Free Edition,[Online], Available : <http://burp/>
- [11] Cloudfare support , "What does the CloudFare cfuid cookie do?," [Online], Available : <https://support.cloudflare.com/hc/en-us/articles/200170156-What-does-the-CloudFlare-cfuid-cookie-do->
- [12] _qca , [Online] , Available: http://cookie-cat.co.uk/cookie_category/_qca-2/
- [13] Using Burp Suite to Brute Force a Login Page, [Online], Available: <https://support.portswigger.net/customer/portal/articles/1964020-using-burp-to-brute-force-a-login-page>
- [14] Zero Web App Security, [Online], Available: <http://zero.webappsecurity.com/online-banking.html>
- [15] SANS Institute InfoSec Reading Room," Conducting a Penetration Test on an Organization", [Online], Available: <https://www.sans.org/reading-room/whitepapers/auditing/conducting-penetration-test-organization-67>
- [16] Penetration Testing-Limitations, [Online] , [Available] : http://www.tutorialspoint.com/penetration_testing/penetration_testing_limitations.htm
- [17] Ahmed Ibrahim, "Manual Security Testing Using Interception", [Online], Aavailable : <https://www.linkedin.com/pulse/20140621040334-24040973-manual-security-testing-using-interception>

- [18] Limitation of Manual Penetration Testing, cronus,[Online],Available:
<http://www.cronus-cyber.com/limitations-of-manual-penetration-testing/>
- [19] Penetration Testing , CSCE 548 Secure Software Development,[Online],Available: <https://cse.sc.edu/~farkas/csce548-2012/lectures/csce548-lect9.ppt>
- [20] Stack Overflow, [Online], Available :
<https://stackoverflow.com/users/login?ssrc=head&returnurl=http%3a%2f%2fstackoverflow.com%2f>
- [21] Faking News , [Online] , Available:
<http://www.fakingnews.firstpost.com/>
- [22] Burp Suite Repeater , [Online],Available:
https://portswigger.net/burp/help/repeater_using.html
- [23] Dawid Czagan,Session Randomness Analysis with Burp Suite Sequencer,[Online],Available:
<http://resources.infosecinstitute.com/session-randomness-analysis-burp-suite-sequencer/>
- [24] Student Centre, [Online] ,Available:
<https://student.uwo.ca/psp/heprdweb/?cmd=login>
- [25] Implementing Single Signon ,[Online], Available :
https://docs.oracle.com/cd/E15645_01/pt850pbr0/eng/psbooks/tsec/chapter.htm?File=tsec/htm/tsec10.htm