Web Based Parameter-Tampering on Shopping Site using BurpSuite Testing

Lijo Jose1, Dr. M. Rajesh Khanna, D.Meganathan, Praveen Kumar B.T

Vel Tech Multi Tech Dr.Rangarajan and Dr.Sakunthala Engineering College Chennai, India

Corresponding author: Lijo Jose, Email: lijomj2001@gmail.com

Client's web application's infrastructure was monitored using several footprinting techniques and scanning of ports where done using n-map and other tools to get the idea of total network infrastructure of our client and several tools to get close enough to the web application was done as part of bug bounty. Among all performed attacks for example SQL injection, remote access, Cross-site scripting, etc. Web Based Parameter tampering attack was successfully deployed in-order to affect the integrity of their web application. Parameter Tampering on a web application is a vulnerability which is found by analysing the site and its pages by using the BurpSuite tool, Freeze the payment request of Client's store web page using intercept action and try to alter the parameters such as price, quantity, product id or to develop parameters such as coupon code and forward the modified request to the web server by unfreezing the request. If there is Integrity and validation mechanism error in the server then the changes are reflected then it is vulnerable to parameter tampering attack and the web server redirects to payment gateway with modified parameter and if found so it is reported to the concern author of the site as bug bounty report.By performing this attack there was a huge impact on their business.

Keywords:Parameter Tampering, BurpSuite, intercept, Forward request, Unfreeze the request, Vulnerable, Report.

2023. In Saroj Hiranwal & Garima Mathur (eds.), *Artificial Intelligence and Communication Technologies*, 527–535. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-955020-5-9-51

1 Introduction

This paper speaks about Burp suite's role in bug bounty practice and its strength in Bug bounty hunting is a method for finding flaws and vulnerabilities in web applications; application sellers reward bounties. Application merchants pay programmers to distinguish and recognize weaknesses in their product, web applications, and portable applications..The bug abundance program, otherwise called the vulnerability rewards program (VRP),is a publicly supported component that permits organisations to award programmers exclusively for their contribution in distinguishing weaknesses in their product[1].

Considering the client, we undertook their web application for footprinting. Footprinting ought to be viewed as a bunch of techniques and cycles that overseers ought to methodically and fastidiously apply in the solidifying of their systems. These strategies, when applied, ought to at the very least, ensure that all data connecting with a particular eSystem or related innovation is recognized, altered, redacted or eliminated. Without the work of a methodical review and framework overview, key snippets of data connected with explicit advances or frameworks could spill into the public space. This might actually open that framework to undesirable consideration and interruption which thus might prompt honour acceleration by possible interlopers.[2], the initial step is to utilise web indexes to acquire beginning data about an objective fully intent on seeing what was accessible and how the information you find can direct your future endeavours[3].

Burp Suite has without a doubt turned into an instrument of decision for web application security testing. Likewise it has advanced such that it can be now utilised to track down weaknesses in API's and Mobile Apps also[4] utilising burp suite several attacks is performed to observe whether the client is vulnerable to OWASP top 10 attacks,. Burp Suite is a brilliant stage for Pentest and security sites. This instrument has awesome elements like:: Proxy Intercept, Spider (for "crawling "), Automatic detection of vulnerabilities , Repeat Tool, Ability to write own plugins[5]. After performing all the possible attacks it has been observed which attacks were vulnerable to the client, and how much the vulnerability affects the client's business is monitored. A detailed study report mentioning the vulnerability and impact study and mitigation steps with proof of concept such as images and videos of attacks are merged along the report and mailed to the client as part of the bug bounty process.

2 Footprinting

Footprinting incorporates an investigation of the primer data. Ordinarily, an analyzer doesn't have a lot of data other than the primer data[5]. It is the most preliminary stage of security testing. Usually websites and tools from search engines are used to identify sub-domains, IP_Address, mail servers, and infrastructure that is being runned(such as Programming language, web server, database, CMS, CDN, hosting). There are plenty of resources to gather the necessary information, below is the list of tools to be used for footprinting:

2.1 Virustotal

VirusTotal is a free instrument that can be downloaded on work area or got to online to dissect dubious documents, hashes or URLs[6]. www.virustotal.com is browsed in the search engine and domain id of the client's application feed and results are obtained which is mentioned in the Table 1 below.

Table 1. Result from Virustotal.

Relation	Result
IP_Address	104.18.21.251
No. of subdomain	60
Domain name	eccouncil.org

2.2 Wappalyzer

Wappalyzer is a program augmentation device accessible in the extension store. Wappalyzer utilises a data set of web application marks that assists with recognizing 900 above web innovations from in excess of 50 classifications. The client's application is loaded and the wappalyzer is initialised and following information is gathered that is mentioned in the Table 2 below.

Table 2. Result from Wapplyzer.

Technology	Deployed
CMS	Wordpress
Javascript frameworks	GSAP
Programming Language	PHP
Database	MySQL
CDN	cloudflare
Hosting	Kinsta
SEO	Yoast SEO Premium

3 Burpsuite Testing

Manual Testing is generally subject to two factors: the abilities of the analyzer and the apparatus utilised for testing. A device like Burp Suite fundamentally helps with satisfying the necessities of manual testing[4]. The testing is done using Burp Suite to search out whether the client's web application is vulnerable to OWASP Top10 vulnerabilities .One of the mindfulness records given by the Open Web Application Security Project (OWASP) people group is the ten most basic web application weaknesses. OWASP top 10 2017 is recorded beneath: [7]

- Injection
- Broken Authentication and Session Management
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging & Monitoring

To test the most common among these vulnerabilities present in the client's application using Burp Suite tool which is the main objective of the project.

3.1 Injection

As indicated by Lwin Khin Shar this attack happens when the engineer didn't matter sufficient limitation on the information boundaries in the web application and in outcome the attacker can embed and run his/her desired query.[8] For the essential distinguishing proof of SQL injection weaknesses, it is energetically prescribed to utilise the Intruder device included in Burp Suite.[1], The client's application was vulnerable to a variation of such Injection which will be mentioned later.

3.2 Broken Authentication and Session Management

It occurs where there will be session capturing or phony confirmation. It incorporates administration guaranteeing know conditions for affirming moreover sessions which could affect the web servers, requisition servers, web arrangement environmental factors and could be area of interest from asserting not right utilisation of privileges.[9]

3.3 XML External Entities (XXE)

XML External Entities licence the incorporation of information progressively from a given resource (local or remote) at the hour of parsing. This component can be taken advantage of by aggressors to incorporate malevolent information from outside URIs or classified information living on the local System.[10] It works because of the ongoing applications that permit clients to transfer XML data to the application. The server processes this data and sends a response.[1] Utilise the Burp Suite Collaborator to actually recognize out-of-the band weaknesses like XML External Entity Injection (XXE)[4]. The mentioned methodology shows that the presence of the XML External Entities (XXE) is negative in client's application.

3.4 Cross-Site Scripting (XSS)

Cross-Site Scripting otherwise called XSS is an application-layer threat that radiates from the security shortcomings of Client-side scripting languages, HTML and JavaScript (or all the more seldom VB Script, ActiveX or Flash). The motivation behind a XSS assault is to infuse vindictive code in a site, to sidestep security access controls and compromise information or perform session hijacking.[11] this XSS attack is named reflected, stored, DOM-based XSS. Fuzzing is a computerised attack strategy of producing an enormous number of solicitations containing attack strings to identify weaknesses like XSS.[12] Subsequently Intruder immensely helps in any of the test situations where we have two of the accompanying things A URL and a parameter to test, rundown of payloads to be submitted to the parameter[4]. Implementing it the result shows that the application of the client is not vulnerable to the XSS attack.

3.5 Insecure Deserialization

Insecure Deserialization happens when untrusted structured data is passed to be differentiated into an object. It tends to be utilised to perform different pernicious errands like remote code execution.[12] Java Deserialization Scanner is a Burp Suite extension to recognize Insecure Deserialization attacks.[13] The client's web application doesn't alter with this strategy.

4 Methodology

As mentioned earlier the client's application is vulnerable to parameter tampering attack which can be categorised as a form of Injection attack. The client operates an e-commerce sub-domain through his web application where business transactions are undergone.

4.1 Parameter tampering

Parameter tampering or parameter manipulation weaknesses permit the control of parameters traded between a client and the server to alter application data, like client certification, credentials and permission and authorizations, cost and amount of items, and others.[14] This is done by burp suite using the proxy options which is available within it. The Burp Suite proxy fills in as Man-in-the-Middle. While getting to an application over HTTPS through Burp Suite, the intermediary will produce a TLS authentication endorsed by its certificate authority and store it on the client framework.[4] this certificate is also known as Certificate Authority certificate or CA certificate. This Certificate is incorporated with the browser engine which would assist the burp suite with tooling to screen both the GET and POST requests, affirmation and message that chips away at the browser engine.

4.2 Working Process

As mentioned previously the client's e-commerce subdomain was vulnerable. In burp Suite, under the proxy tab which fills in as Man-in-the Middle attack has an option Intercept. To intercept a request. Explore to "Proxy ➤ Intercept" and Click on "Intercept if off"[4] When a request is intercepted, it very well may be seen in various ways. The choices accessible for a straightforward HTTP request are Raw, Params, Headers, and Hex in view of the sort of request[1] on burp suite. This request will contain host, cookie details, encoding, content-details and parameter such as currency, language, amount, quantity, product name, product id, purchase step, etc with its value which is shown in the fig.1

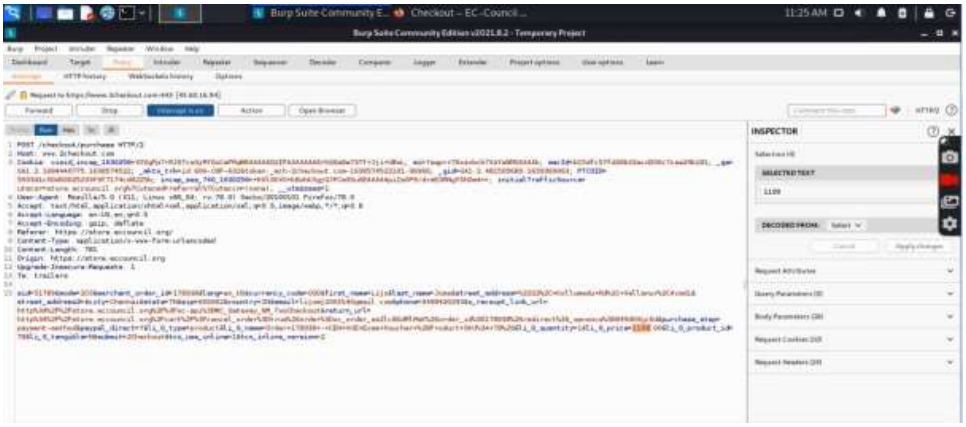

Fig1. POST request intercepted by BurpSuite

Since its an e-commerce the price parameter's value was manipulated to a new value and the manipulated POST request is forwarded by clicking on the forward button which is near to the intercept button in the proxy tab which would forward the request to the web server, the web server passess the information to the payment gateway where the information contains values that is given in the POST request, if the price of the product changed then application is vulnerable which is explained in the Fig.2
,

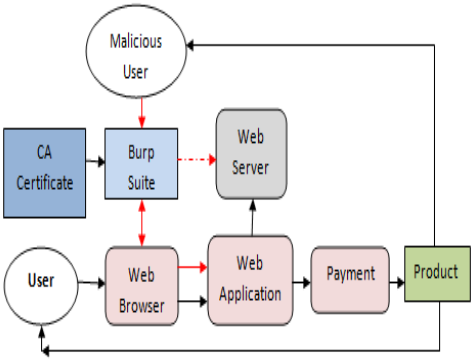

Fig 2. Architecture of the attack

In the Fig.2 the red arrow indicates the malicious users access and black coloured arrow indicates normal user request. The tester or malicious user integrates the burp suite with the browser which helps the burp suite to monitor the GET and POST request and is able to intercept the request. The modified POST request which is mentioned as a red dotted arrow in Fig.2 is forwarded to the server, the server forwarded it to payment. This vulnerability arises due to input validation mechanism error in the server. One of the principal reasons for Input Validation Mechanism mistakes is to endeavour to submit information which the application doesn't anticipate getting. Regularly, an application will play out some sort of second look for good measure on client input. This really take a look at attempts to guarantee that the information is valuable. More significant checks are important to distinguish the information from crashing the server.[15] This attack would cause huge business loss to the client.
On July 25,2016 Uber had a sql injection attack and was accounted for by Orange and was compensated total $ 4000 this attack initiates the sleep command(12)as the result postpones the reaction by 12 seconds[1]; Correspondingly on September 11,2018 Zomato endured with SQLi weakness and was accounted for by samengmg and was compensated with $1000. The client's application faces one such weakness and revealing it is significant since it can enormously influence business where $ 100 item is bought at $ 1 utilising the previously mentioned assault.

At the initial stage title must be mentioned which is then followed by specifying the origin of attack and severity and mentioning a detailed description of the attack. The next step is to mention how it would affect the client and their business and these vulnerabilities must be avoided and the measures are detailed in the report with images, videos and bills of the attack is attached as proof-of-concept and the most important is to mention the steps how to reproduce the attack again in an detailed description and the format is mentioned in the Table 3.

Table 3. Bug Report Format

Topic	Description
Vulnerability Title	Web Parameter Tampering(Injection attack)
Vulnerable Domain/URL	http://store.eccouncil.org/
Severity	medium
Description	The Web Parameter Tampering assault depends on the control of Parameters traded among client and server to alter application information, like user

	accreditations and consents, cost and amount of items, and so on. Typically, this data is put away in Cookies, stowed away form fields, or URL Query Strings, and is utilised to increment application usefulness and control. Using BurpSuite tool I intercepted the request for checkout and was able to alter the price of products available in the store.eccouncil.org,
Impact of the vulnerability	Contents can be changed such as price which can result in huge business loss and integrity issues.URL Parameters that an aggressor can change, including Attribute Parameters and inner Modules. An Attribute Parameters are unique parameters that characterize the behavior of the transferring page.
Steps to reproduce the issue	step 1) Visit store.eccouncil.org step 2) Select a product and proceed till billing and select place order step 3) It directs to the 2checkout page, now turn on the intercept in BurpSuite proxy. step 4) click on the 2checkout button in the browser. step 5) The BurpSuite captured the request, and in the request the price is mentioned, edit the price given and click Forward and turn off the intercept. step 6) The browser redirects to paypal payment where the attackers needs to pay the tampered amount
Remediation	1) The forms on the site should have some built-in protection, 2) Utilising regex to restrict or approve information 3) Server-side approval contrasted and all data sources 4) Don't allow interception
Proof-of-concept	JPEG/MP4

5 Conclusion

Developed an unmistakable comprehension of how to report defects in a program and which imperfections to report. This has driven me to figure out a configuration for revealing, yet this arrangement isn't all inclusive and it might fluctuate from one individual to another and case to case.[1] These reports might be rewarded or may not be but the standard format is pre-defined and it is followed. This paper mentioned the power of burp suite in preventing business and integrity loss to the clients and its contribution to bug bounty programs as well as mentions a great tool for attacks where giving a massive hit to the economy of the global market.

References

[1] Carlos A. Lozano and Shahmeer Amir “Bug Bounty Hunting Essentials: Quick-paced Guide to Help White-hat Hackers Get Through Bug Bounty Programs”N.p.: Packt Publishing

[2] C. Wren, D. Reilly and T. Berry, "Footprinting: A Methodology for Auditing eSystem Vulnerabilities," 2010 Developments in E-systems Engineering, 2010, pp. 263-267, doi: 10.1109/DeSE.2010.49.

[3] Oriyano, Sean P. 2014. CEH: Certified Ethical Hacker Version 8 Study Guide. N.p.: Wiley.

[4] Rahalkar, Sagar. 2020. A Complete Guide to Burp Suite: Learn to Detect Application Vulnerabilities. N.p.: Apress.

[5] R. E. López de Jiménez, "Pentesting on web applications using ethical - hacking," 2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI), 2016, pp. 1-6, doi: 10.1109/CONCAPAN.2016.7942364.

[6] R. Masri and M. Aldwairi, "Automated malicious advertisement detection using VirusTotal, URLVoid, and TrendMicro," 2017 8th International Conference on Information and Communication Systems (ICICS), 2017, pp. 336-341, doi: 10.1109/IACS.2017.7921994.

[7] H. Sϕhoel, M. G. Jaatun and C. Boyd, "OWASP Top 10 - Do Startups Care?," 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2018, pp. 1-8, doi: 10.1109/CyberSecPODS.2018.8560666.

[8] A. Sadeghian, M. Zamani and S. Ibrahim, "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques," 2013 International Conference on Informatics and Creative Multimedia, 2013, pp. 265-268, doi: 10.1109/ICICM.2013.52.

[9] H. Singh and M. Dua, "Website Attacks: Challenges and Preventive Methodologies," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 381-387, doi: 10.1109/ICIRCA.2018.8597259.

[10] S. Jan, C. D. Nguyen and L. Briand, "Known XML Vulnerabilities Are Still a Threat to Popular Parsers and Open Source Systems," 2015 IEEE International Conference on Software Quality, Reliability and Security, 2015, pp. 233-241, doi: 10.1109/QRS.2015.42.

[11]A. Stasinopoulos, C. Ntantogian and C. Xenakis, "Bypassing XSS Auditor: Taking advantage of badly written PHP code," 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2014, pp. 000290-000295, doi: 10.1109/ISSPIT.2014.7300602.

[12] Hands-on Penetration Testing for Web Applications: Run Web Security Testing on Modern Applications Using Nmap, Burp Suite and Wireshark (English Edition). 2021. N.p.: BPB Publications.

[13] Shah, Dhruv, Riyaz A. Walikar, and Carlos A. Lozano. 2019. Hands-On Application Penetration Testing with Burp Suite: Use Burp Suite and Its Features to Inspect, Detect, and Exploit Security Vulnerabilities in Your Web Applications. N.p.: Packt Publishing.

[14] Ackerman, Pascal. 2017. Industrial Cybersecurity: Efficiently Secure Critical Infrastructure Systems. N.p.: Packt Publishing

[15] Y. Park and J. Park, "Web Application Intrusion Detection System for Input Validation Attack," 2008 Third International Conference on Convergence and Hybrid Information Technology, 2008, pp. 498-504, doi: 10.1109/ICCIT.2008.338.