Security Testing of Web Applications Using Burp Suite

Arvind Choudhary
20BCAR0239

A Vaibhav Jain
20BCAR0238

Suman Garai
20BCAR0246

***Abstract*— In recent years, utilization of web applications, web hacking exercises have grown exponentially. Organizations are confronting extremely critical difficulties in anchoring their web applications from rising cyber threats, as bargain with the assurance issues don't appear to be the right approach. Vulnerability Assessment and Penetration Testing (VAPT) methods help us find these vulnerabilities / security loopholes in our systems even before an intruder could find a way to get it. This helps avoid zero-day exploits. This paper aims to elucidate the overview of Vulnerability Assessment and Penetration Testing and introduce one of the most efficient tools used to perform these tests. This paper also presents the tool, with the goal to understand their utility and benefit the most from the tests.**

***Keywords*— cyber security, VAPT, zero-day exploits, vulnerabilities**

I. INTRODUCTION

Threats to integrity and confidentiality of information are constantly increasing. To protect our information, we are obliged to perform security tests. Attackers are always finding new ways to exploit a system this leads to evolution of new vulnerabilities. Security testing is a process which is intended to reveal flaws in the security mechanisms and find potential vulnerabilities to check if a system is compromised. If any system is not tested for security related issues it might end up with security loopholes which may result various risk factors including loss/ leakage of information which is confidential to the organization. Vulnerability Assessment and Penetration Screening (VAPT) are two forms of vulnerability testing. The assessments have different strengths and they are often combined to accomplish a far more complete vulnerability analysis [1]. In a nutshell, Penetration Testing and Vulnerability Assessments perform two different tasks, usually with different results, within the same area of focus. Vulnerability assessment comprises of identifying the various loopholes or weakness in a system through which an intruder or attacker may possibly attack the system. Through vulnerability assessment we can merely only identify the risks and cannot exploit them. This is where the role of penetration testing comes in. Unlike the vulnerability assessment pen-testing does not merely involve discovering the threats, we exploit the threats, or the loop holes reported by vulnerability assessment to ensure if they really exist or if those were false alarms [2]. Penetration tests find exploitable flaws and measure the severity of each exploited threat. It is always suggested to perform these two assessments together.

VAPT surveys the adequacy and inadequacy of the security courses of action of the web application to remain ensured against the rising Cyber dangers. The anticipated work builds up a flexible instrument which can discover vulnerabilities from web applications. In this way, identification of Vulnerabilities and cure of a comparable has turned out to be one among the prime issues for associations. With the developing between availability of frameworks and progression in Cyber Services, the degree of Cyber Attacks has conjointly misrepresented. In this manner as to remain resistant and for risk minimization, Vulnerability Assessment and Penetration Testing is led by the associations on customary premise.

The two sorts of security assessment are vulnerability assessment and penetration testing which can frequently be consolidated for accomplishing better powerlessness examination results.VA and PT are only two distinct errands giving diverse outcomes yet inside a similar workspace. We have Vulnerability appraisal devices for finding vulnerabilities, though no separation found between sorts of imperfections that reason harm on misuse and those that don't do as such. There are Vulnerability scanners which create caution for organizations about pre-presence of any blemishes in code and area of imperfections. Entrance tests are performed to abuse the vulnerabilities in a framework to get any method for unapproved access or probability of any malicious movement and utilized in intrusion detection of defects presenting the danger to the application. These tests discover exploitable blemishes and measure their seriousness. These are additionally useful for demonstrating the measure of harm it could cause amid the genuine assault. In this way, joined bundle of infiltration testing and weakness appraisal instruments give a point-by-point perspective of existing defects alongside the hazard related with it. For a security tester to completely test a web application for security threats, he cannot stop with performing only simple web security search. A complete analysis of the system under test has to be performed and accordingly the test has to be carried forward [3]. To perform this, just a vulnerability assessment or a penetration test is not sufficient.

Therefore, we are going to perform a deep security test for the web application, utilizing a robust tool designed for tests like port scanning, vulnerability assessments, penetration testing, network capturing etc., analyse the result obtained and draw conclusions accordingly.

II. VULNERABILITY ASSESSMENT AND PENETRATION TESTING

First, it is known that web applications or websites are in general vulnerable to security attacks, be it code based or network based. These vulnerabilities give the attackers an opportunity to take control of the system and its components. This is when we call a system compromised. Not only highly used banking sites or so even a basic website written in plain simple and static html needs a detailed vulnerability assessment and penetration testing. It is important to understand the seriousness of vulnerable websites or webservers. An attacker may potentially steal sensitive data from the server or disrupt website operations or simply deface pages of the website [4]. It is crucial to realize that protecting the web application with just firewalls is not enough hence we need a periodical detailed VAPT to ensure the system is fool proof.

Vulnerabilities are framework imperfections, bugs, misconfiguration that make it defenceless against the attacks. Evaluating of these framework vulnerabilities empower us to distinguish and introduce security patches, in order to shield the framework from the danger of being harmed. VAPT strategy is directed in two noteworthy parts. The main half manages the Analysis and Discovery of existing Vulnerabilities. The second half manages the Exploitation of the distinguished arrangement of Vulnerabilities, to assess their Severity and effect over the Target framework. Vulnerability assessment is a detached methodology though penetration testing is a functioning methodology where security experts recreate assault and test the objective site and its resilience control against assaults.

III. TYPES OF VULNERABILITIES

Open Web Application Security Project (OWASP) is an organization that provides unbiased and practical, cost-effective information about computer and Internet applications. This organization has taken data and surveys from the most powerful companies across the globe and put together an OWASP Top 10 Web Application Security Risks for web applications [5]. This was put together to provide guidance to developers and security professionals on the vulnerabilities that have most risk and are commonly discovered in software applications, these are also easily exploitable.

In spite of having such clear guidelines most organizations continue to fail in protecting their systems from these common attacks or vulnerabilities which are most often simple and easy to identify and resolve. Most organizations fall prey because they have misconceptions about what a web application is. A one-time vulnerability scan or penetration assessment of a handful of business-critical apps is not effective approach to application security. There should be an approach that continuously assesses the applications an organization develops for production is effective and recommended application security.

The following threats are the OWASP Top 10 Web Application Security Risks as of 2017 OWASP survey

- Injection
- Broken Authentication and Session Management
- Sensitive Data Exposure
- XML External Entity
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting
- Insecure deserialization
- Utilizing Components with Known Vulnerabilities
- Lacking Logging and Monitoring

IV. ABOUT THE TOOL

Burp suite is a java based integrated platform for performing security testing of web applications [6]. Burp suite in general is a web penetration testing framework. This tool is exclusively made for web applications. It is developed by the company named PortSwigger, which is also the alias of its founder Dafydd Stuttard. Burp Suite aims to be an all-in-one set of tools and its capabilities can be enhanced by installing add-ons that are called BApps.It is the most popular tool among professional web app security researchers and bug bounty hunters [17]. There is a community edition and a professional edition both of which prove to be extremely useful. Burp suite is now used by most of the professional testers as a part of their industry standard tools. Burp allows us to perform complex and customized tasks and also write up individual plugins as per our requirements. Burp suite is simply an intercepting proxy which helps a penetration tester to configure traffic to route through Burp. Burp suite performs in a way similar to man in the middle attack. It is placed between the web client and server, so this captures every request and response from and to the client and the server.

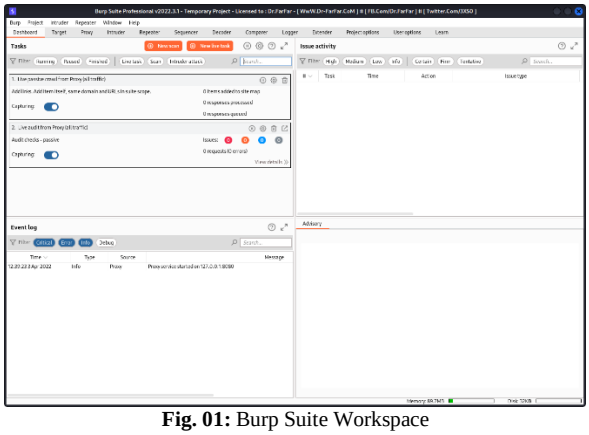

Fig. 01: Burp Suite Workspace

Burp has an option called the intercept, this helps the tester to pause the traffic and manipulate the data and test how it affects the data. It can do proxy it using its spidering and crawling options. Once Burp has visited each link then we can both actively and passively scan the website for vulnerabilities. Meanwhile it also has other options like the repeater, comparer and decoder which helps the tester to replay the requests or modify them and check the behaviour of the website to these requests. In addition to these it has an intruder option which allows us to perform customized brute force attacks and a sequencer which is mostly used to perform fuzzing find of operations, which is it sends the web page random data load to see how the page handles it and also to find unknown vulnerabilities.

V. TEST WORKFLOW

To perform a penetration test, a testing plan must be initially made. To perform an effective security test capturing most, if not all the vulnerabilities of a system, a number of security modules have been used to test a web application. These modules are all specifically designed keeping the OWASP security risks as a priority to perform unique tasks and hence putting different unique tools together gives us a wide perspective. Before a system is tested it should first be scanned for any open ports, these are the ports that will be tested as these will be the entry point for the attackers. The Proxy tool lies at the heart of Burp's workflow. It lets you use Burp's embedded browser, or our own external browser, to navigate the application, while Burp captures all relevant information and lets you easily initiate further actions. After completing your recon and analysis of the target application, and any necessary configuration of Burp, you can begin probing the application for common vulnerabilities. At this stage, it is often most effective to use several Burp tools at once, passing individual requests between tools to perform different tasks, as well as going back to our browser to perform additional tests. Throughout Burp, we can use the context menu to pass items between tools and carry out other actions [7].

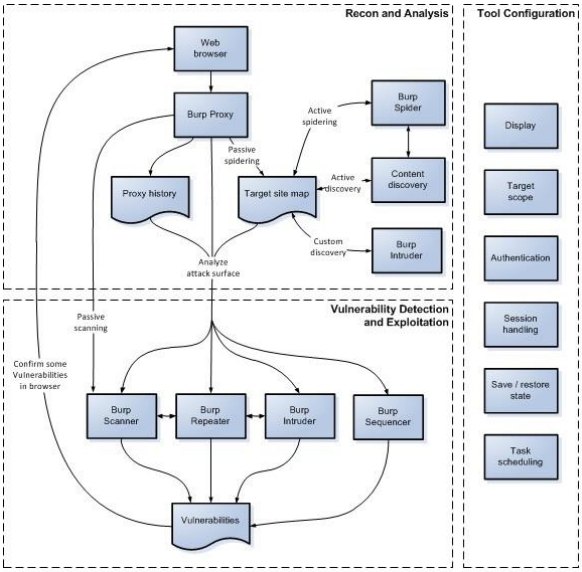

Fig. 02: Burp Suite Testing Workflow

The following modules were used to perform the entire procedure and the results were observed for the same. The testing platform used is Kali Linux v2022.1 and Burp Suite Professional v2022.3.1. The tests were conducted on a variety of web application and web servers.

A. Burp Scanner

Burp Scanner is a tool for performing automated scans of web sites, to discover content and audit for vulnerabilities. The work involved in performing a scan comprises two key phases: [8] The crawl phase of a scan involves navigating around the application, following links, submitting forms, and logging in where necessary, to catalogue the content of the application and the navigational paths within it. This seemingly simple task presents a variety of challenges that Burp's crawler is able to meet, to create an accurate map of the application [9]. The audit phase of a scan involves analysing the application's traffic and behaviour to identify security vulnerabilities and other issues. Burp Scanner employs a wide range of techniques to deliver a high-coverage, dead-accurate audit of the application being scanned Burp Scanner carries out several distinct audit phases. These are divided into three areas namely: Passive phases, Active phases and JavaScript analysis phases. Performing multiple phases within each area allows Burp to: effectively find and exploit functions that store and return user input., avoid duplication by handling frequently occurring issues and insertion points in an optimal manner and execute applicable work in parallel to make most efficient use of system resources [10].

B. Burp Repeater

Burp Repeater is a simple tool for manually manipulating and reissuing individual HTTP and WebSocket messages, and analysing the application's responses. You can use Repeater for all kinds of

purposes, such as changing parameter values to test for input-based vulnerabilities, issuing requests in a specific sequence to test for logic flaws, and reissuing requests from Burp Scanner issues to manually verify reported issues. The main Repeater UI lets you work on multiple different messages simultaneously, each in its own tab. When you send messages to Repeater, each one is opened in its own numbered tab. You can rename tabs by double-clicking the tab header [11]. To use Burp Repeater, we can send a request to it from one of Burp's other tools. Burp Repeater makes it much simpler to probe for vulnerabilities, or to manually confirm ones that were identified by Burp Scanner [12].

C. Burp Sequencer

Burp Sequencer is a tool for analysing the quality of randomness in a sample of data items. We can use it to test an application's session tokens or other important data items that are intended to be unpredictable, such as anti-CSRF tokens, password reset tokens, etc [13]. Burp Sequencer can operate on any sample size between 100 and 20,000. The analysis mainly uses significance-based statistical tests in which the assumption that the tokens are random is tested by computing the probability of the observed results arising if this assumption is true. If the probability falls below a particular level (the "significance level") then the assumption is rejected and the anomalous data is judged to be non-random.
This approach allows Burp Sequencer to give an intuitive overall verdict regarding the quality of randomness of the sample. To gain a deeper understanding of the properties of the sample, to identify the causes of any anomalies, and to assess the possibilities for token prediction, Burp Sequencer lets us drill down into the detail of each character- and bit-level test performed [14].

D. Burp Intruder

Burp Intruder is a tool for automating customized attacks against web applications. It is extremely powerful and configurable, and can be used to perform a huge range of tasks, from simple brute-force guessing of web directories through to active exploitation of complex blind SQL injection vulnerabilities. Burp Intruder works by taking an HTTP request (called the "base request"), modifying the request in various systematic ways, issuing each modified version of the request, and analysing the application's responses to identify interesting features. For each attack, you must specify one or more sets of payloads, and the positions in the base request where the payloads are to be placed. Numerous methods of generating payloads are available (including simple lists of strings, numbers, dates, brute force, bit flipping, and many others). Payloads can be placed into payload positions using different algorithms. Various tools are available to help analyse the results and identify interesting items for further investigation [15].

VI. RESULTS AND OBSERVATIONS

The below figures are the observed evaluation and results of the vulnerabilities during the various VAPT tests. These modules were run on various web applications. Some of the web applications are Google Gruyere webserver, Acunetix websites and OWASP vulnerable servers and websites.

Figure 03 is a snapshot of the results obtained from the Burp Scanner tested on https://google-gruyere.appspot.com/start website. On the left half, we can see the audit reports of the vulnerabilities present in the obtained. It's generated on the basis of the base request made and the response received. The left half Dashboard section, shows us the event log of the crawler visiting the sites and the actions taking by it, the issue activity window states the issues found in the site along with a brief summary of the same and timestamps of when it is found. In the advisory section, we can further expand on the issue details, background, remediations, references and classifications on any specific one, present in the issue activity.

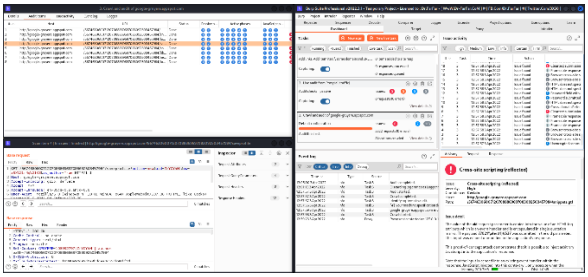

Fig. 03: Vulnerability scanning results from Burp Scanner

We can see a lot of Cross-Site Scripting (XSS) & Clickjacking issues; issues in communication encryption, transportation security, password fields, backup & cookies related, HTML and HTTPS response too.

Figure 05 shows us the result obtained from the Burp Sequencer when tests were conducted on http://testaspnet.vulnweb.com website. The packets are at first intercepted using Burp proxy and thereafter sent to the Sequencer with the packet containing Session ID. On the left half, n the summary section we can see overall result the randomness of the session id's generated, along with different types of analysis options and configurations. It also shows the number of random tokens being generated. On the right half, we see the input from the proxy along with token generation and live capture options.

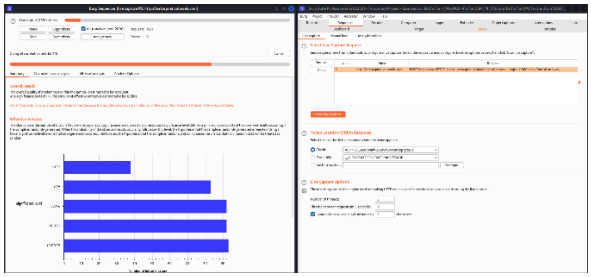

Fig. 05: Session ID randomness check results from Burp Sequencer

As stated, the overall result seems to be excellent, which means that it is quite difficult to crack the session ids generated by the site of any user when they are currently surfing on the site while staying logged in. The session ids generated by the site are of 83 bits which is quite close to the 128 bits length, required to prevent brute-force session guessing attacks, as mentioned in the OWASP site [16].

VII. CONCLUSION

Though the performed vulnerability assessment and penetration tests were able to find most of the known vulnerabilities, these tests are alone not enough to certify a system as risk-free. To explore all the aspects of more specialized and system-specific tests must be performed which can include manual code analysis and writing custom scripts.

REFERENCES

[1] AL-Ghamdi and Abdullah Saad AL-Malaise, "A Survey on Software Security Testing Techniques," proceedings of ijcst conference, 2016.

[2] AL-Ghamdi and Abdullah Saad AL-Malaise, "A Survey on Software Security Testing Techniques," proceedings of ijcst conference, 2016.

[3] Shah. Sugandh. & B.M. Mehtre, "A Modern Approach to CyberSecurity Analysis Using Vulnerability Assessment and Penetration Testing," Proceedings of 2013 NCRTCST, Hyderabad (A.P), India, 2014.

[4] Buja, G., Bin AbdJalil, K., BtHjMohd Ali, F.; Rahman, T.F.A., "Detection model for SQL injection attack: An approach for preventing a web application from the SQL injection attack," proceedings of 2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE).

[5] Buja, G., Bin AbdJalil, K., BtHjMohd Ali, F.; Rahman, T.F.A., "Detection model for SQL injection attack: An approach for preventing a web application from the SQL injection attack," proceedings of 2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE).

[6] International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019

[7] PortSwigger. (n.d.). Burp Suite, 2018, Information available from: <https://portswigger.net/burp/> accessed at 09:50 hrs on 17-01-2019.

[8] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/scanner> accessed at 01:10 hrs on 03-04-2022.

[9] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/scanner/crawling> accessed at 01:12 hrs on 03-04-2022.

[10] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/scanner/auditing> accessed at 01:15 hrs on 03-04-2022.

[11] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/desktop/tools/repeater/using> accessed at 05:13 hrs on 03-04-2022.

[12] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/desktop/tools/repeater/getting-started> accessed at 05:13 hrs on 03-04-2022.

[13] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/desktop/tools/sequencer> accessed at 05:07 hrs on 03-04-2022.

[14] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/blog/introducing-burp-sequencer> accessed at 05:37 hrs on 03-04-2022.

[15] PortSwigger. (n.d.). Burp Suite, 2022, Information available from: <https://portswigger.net/burp/documentation/desktop/tools/intruder/using> accessed at 05:42 hrs on 03-04-2022.

[16] OWASP. 2022, Information available from: <https://owasp.org/www-community/vulnerabilities/Insufficient_Session-ID_Length> accessed at 08:53 hrs on 03-04-2022.

[17] https://www.geeksforgeeks.org/what-is-burp-suite/#_=_