Web Application Penetration Test Checklist | Part – 02

In this article I am going to share a checklist which you can use when you are doing a penetration test on a website, you can also use this list as a reference in bug bounties. This list is made for intermediates, so they can look it for reference.

Before starting this list I want to make a request that this is my advice that you should complete the previous [checklist](), so in this process you will not get confused.

**You are not genius!** Remember this thing, so if you don’t understand something just Google about it and so some research, I also don’t know everything and there could be things that I have missed, so don’t worry and keep learning.

**The list**

1. Test for credentials transported over encryption.
   * When you submit your login/registration data try intercepting the request and changing the requests method. Post to Get/, and Get to Post. If any points of time you find the data submitted by user are transported without encryption you can make this as low-level bug.
2. Test for default credentials on admin page/console or any sign in panel.
   * Try submitting default username passwords like “admin”:”admin”, “admin”:”password”
3. Bypassing the authentication.
   * Forced browsing: Directly visiting the section of the website which requires authentication. For example, if you have to login at <https://testwebsite.com/login> to visit <https://testwebsite.com/information>, but if you can visit <https://testwebiste.com/information> directly just by typing this URL in the browser without authenticating then this will be known as forced browsing.
   * Parameter modification: Try changing response which comes from the server for example, if your server response <https://testwebsite.com/auth=false> then try changing the parameter auth=false to auth-true.
   * Session ID brute forcing.
   * SQL injection.
4. Check for broken access control.
5. Remember password checking.
   * Check that is password being stored in the cookies or being constantly transferred in every request of the website. The credentials should only be sent I login phase.
6. Check for directory traversal includes file input.
   * You have to check each and every input which your website and its directories take from user.
   * You can referrer to this [article](<https://medium.com/@nerdy_researcher/directory-traversal-aka-path-traversal-c76dc7bbe61#:~:text=What%20is%20Directory%20Traversal%3F,and%20sensitive%20operating%20system%20files)>).
7. Checking for privilege escalation.
   * You can check for this at some places like if user can make payment, adding something, sending message to someone.
   * You can intercept request of two different sets of account and try modifying parameters like grp, id, and role if they exist.
   * You can referrer to this [article]( <https://shahjerry33.medium.com/privilege-escalation-hello-admin-a53ac14fd388>).
8. Check for IDOR (Insecure Direct Object Reference).
   * You can try for getting access to other user data by changing parameters in URL.
9. Check for bypassing session management object.
   * Set-cookies are secure or not?
   * Are cookies transmitted in encrypted manner?
   * Make sure cookies are not same every time when your browse website.
   * Sometimes website can leak their token structure/information try to find it.
   * Session ID predictability.
   * Brute forcing session ID.
10. Check for CSRF.
11. Check for XSS (stored, reflected, blind).
12. Check for SQL injection (blind, In band, Out band, Error based etc.).
13. Check for XML injection.
14. Check for file upload.
15. Check for open redirection or client-side open redirection.
    * You can referrer to this [article]( <https://corneacristian.medium.com/top-25-open-redirect-bug-bounty-reports-5ffe11788794>).
16. Checking for web sockets vulnerabilities.
17. Check for code execution.
    * You can referrer to this [article]( <https://medium.com/@ashishrohra/remote-code-execution-explaination-writeups-and-tools-a8e4c3362259>).
18. Check for SSRF (Server Side Request Forgery).
19. Check fir command injection.
    * You can referrer to this [article](<https://medium.com/ax1al/os-command-injection-beginners-guide-637e1eed1fde>).
20. Checking for business logic flaws.
    * You can referrer to this [article](<https://medium.com/armourinfosec/exploiting-business-logic-vulnerabilities-234f97d6c4c0>).
21. Checking for LDAP injection.
    * You can referrer to this [article]( <https://medium.com/@hunter_55/ldap-admin-account-bypassed-2cc8b264d66e>).
22. Check for HTTP request smuggling.

So this was all about some more things to check while doing penetration test on a website or in a bug bounty program. Hope you liked it and learned something new from it.

If you have any doubt, question, quires related to this topic or just want to share something with me, than please feel free to contact me.