* **Low risk bugs**
  + **WEEK password POLICY in Register page**
    - **Summary**
    - **Remediation**
  + **MISSING RATE LIMIT in Login page**
    - **Summary**
    - **Remediation**
    - Allow for brute-forcing
  + **User/email enumeration** 
    - **Summary**
    - **Remediation**
  + **MISSING IMPORTANT SECURITY FLAGS** 
    - **Summary**
    - **Remediation**
    - Missing X-Frame-Options
  + **COOKIE WITHOUT HTTPONLY FLAG SET**
    - **Summary**
    - **Remediation**
  + **COOKIE WITHOUT secure FLAG SET**
    - **Summary**
    - **Remediation**
  + **Password Autocomplete**
    - **Summary**
      * An attacker with local access to the PC or by compromising the browser could obtain the clear text password from the browser saved password.
    - **Remediation** 
      * To prevent browsers from storing credentials entered into HTML forms, include the attribute autocomplete="off" within the FORM tag (to protect all form fields) or within the relevant INPUT tags (to protect specific individual fields)
  + **Check for secure Transmission**
    - **Check credentials only delivered over HTTPS**
    - **Check that the login/registration form is delivered over HTTPS**
    - **Check session tokens only delivered over HTTPS**
    - **Missing HTTP Strict Transport Security (HSTS)** 
      * **summary**
        + It has been discovered that the affected application is using HTTPS, however does not use the HSTS header. The HTTP protocol by itself is clear text, meaning that any data that is transmitted via HTTP can be captured and the contents viewed. To keep data private and prevent it from being intercepted, HTTP is often tunneled through either Secure Sockets Layer (SSL) or Transport Layer Security (TLS). When either of these encryption standards are used, it is referred to as HTTPS.
        + HTTP Strict Transport Security (HSTS) is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. This will be enforced by the browser even if the user requests a HTTP resource on the same server.
        + Cyber-criminals will often attempt to compromise sensitive information passed from the client to the server using HTTP. This can be conducted via various Man-in-The-Middle (MiTM) attacks or through network packet captures.
  + **click-jacking**
    - **summary**
      * If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.
    - **Remediation**
      * Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker
  + **SESSION FIXATION**
    - **Vulnerability description:**
    - **Remediation**
  + **Missing SPF Record**
    - **Vulnerability name:**
      * Missing SPF record
    - **Vulnerability description:**
      * An SPF record is a type of Domain Name Service (DNS) record that identify which mail servers are permitted to send email on behalf of your domain. The purpose of an SPF record is to prevent spammers from sending messages with forged from addresses at your domain
    - **checking Missing SPE**
      * There Are Various Ways of Checking Missing SPF Records on a website but the Most Common and Popular way mxtools.com
    - **Steps to Check SPF Records on**
      * 1- Go to <http://w.kitterman.com/spf/validate.html> or mxtools.com
      * 2-Enter Target Website “**targetsite.com**”
        + If you seem any SPF Record than Domain is Not Vulnerable But if you see Nothing Here then "! You Found a Bug"
    - **Attack Scenario &impact:**
      * If there is No SPF Records. An Attacker can spoof Email Via any Fake mailer Like “Anonymousmail.me” and Send Email From "Support or admin etc.."
        + Example Email: **support@targetsite.com**
      * And with some Social engineering He can takeover a user account Because The Email was sent from the Authorized Domain. So users can get tricked Easily.
    - **Remediation** 
      * As a website owner you should prevent your domains being used in spam mail by adopting both of the following approaches:
        + Implement the Sender Policy Framework (SPF): publish a DNS record to explicitly state which servers are allowed to send email from your domain.

Add an SPF record for your domain name at your DNS provider.

Create a new TXT record.

Set the Host field to the name of your subdomain (for example, mail if your email address is contact@mail.example.com), or to @ if you do not use a subdomain.

Enter your SPF record for the TXT Value field (for example, v=SPF1 a mx include:secureserver.net ~all)

* + - * + Implement Domain Key Identified Mail (DKIM): use a digital signature to prove that outgoing email was legitimately sent from your domain, and that it wasn’t modified in transit.
        + There is also an emerging umbrella standard called DMARC (“Domain-based Message Authentication, Reporting & Conformance”) that you should be aware of.
* **Medium Risk Bugs**
  + **Reflective XSS**
    - **Report Title:**
      * Reflective XSS
    - **Vulnerability description:**
      * URL redirection is sometimes used as a part of phishing attacks that confuse visitors about which web site they are visiting
    - **Impact**
      * whenever a user visit this URL it will redirect them To site.com It is used in phishing attacks
    - **Vulnerable Url:**
      * URL
        + <https://protect2.fireeye.com/url?k-171e4e52f8ee2b.8817lefe-1138524403bba8dfsushttps://www.site.com>
      * Param
    - **Payload**
    - **How to reproduce this vulnerability**
      * 1-> open this URL "https://protect2.fireeye.com/url?k=88171e45218ee2b.88171efe-1138524403bba@dfsu-https://www.bing.com
      * 2-> it will directly redirect to bing.com
    - **POC**
    - **Remediation**
      * Your script should properly sanitize user input.
  + **URL Redirection**
    - **Report Title:**
      * URL redirection
    - **Vulnerability Details**
      * **Vulnerability description:**
        + URL redirection is sometimes used as a part of phishing attacks that confuse visitors about which web site they are visiting
        + For more information You can check

https://owasp.org/www-community/attacks/xss/

* + - * **Impact**
        + whenever a user visit this URL it will redirect them To site.com It is used in phishing attacks
      * **Risk Breakdown**
        + Risk: High
        + Difficulty to Exploit: Medium
        + CVSS2 Score: 7.9 (AV:N/AC:M/Au:S/C:C/I:C/A:N)
      * **Vulnerable Url:**
        + https://protect2.fireeye.com/url?k-171e4e52f8ee2b.8817lefe-1138524403bba8dfsushttps://www.site.com
    - **How to reproduce this vulnerability**
      * 1-> open this URL "https://protect2.fireeye.com/url?k=88171e45218ee2b.88171efe-1138524403bba@dfsu-https://www.bing.com
      * 2-> it will directly redirect to bing.com
    - **POC**
      * Attach proof-of-concept scripts, screenshots, screen recordings, and so on
    - **Remediation**
      * Your script should properly sanitize user input.
      * Also you can check
        + https://cheatsheetseries.owasp.org/cheatsheets/Cross\_Site\_Scripting\_Prevention\_Cheat\_Sheet.html
* **Resources**