Checklist

* + Understand the application, and its business need
  + Check how many user roles exist, and create 2 accounts for each role
  + Cross-site scripting (XSS) - any user input being reflected in the page (i.e. firstname, lastname, search text and so on)
  + Server-side request forgery (SSRF) - Any parameter that has a URL or path as a value (i.e. <https://www.site.com/page.aspx?Origin=google.com&url=https://www.google.com>)
  + Insecure Direct Object Reference (IDOR) - Check for any value that have a number or identifier, linked to user data or user activities
  + Server-side response manipulation - Check JSON responses, and manipulate any value that is being interpreted by the code (i.e. isadmin=false -- change it to isadmin=true and see if that would make you an admin)
  + For a misconfigured CORS to be a vulnerability, 3 things must be there. **If any of these conditions fails, it is not a vulnerability.**
    - Following 2 response headers must present:
      * access-control-allow-origin: attackercontroledsite.com
      * access-control-allow-credentials: true
    - Response must be 200 OK, AND must contain sensitive data
    - There should not be any extra headers (i.e. Authorization)
  + Host header injection - in the password reset page, set the hostname to your controlled hostname, and send a password reset to your account, then check your email if the password reset link includes your hostname.
  + XXE - any user input that contains XML data. Or any file upload page that would accept .xml or .svg