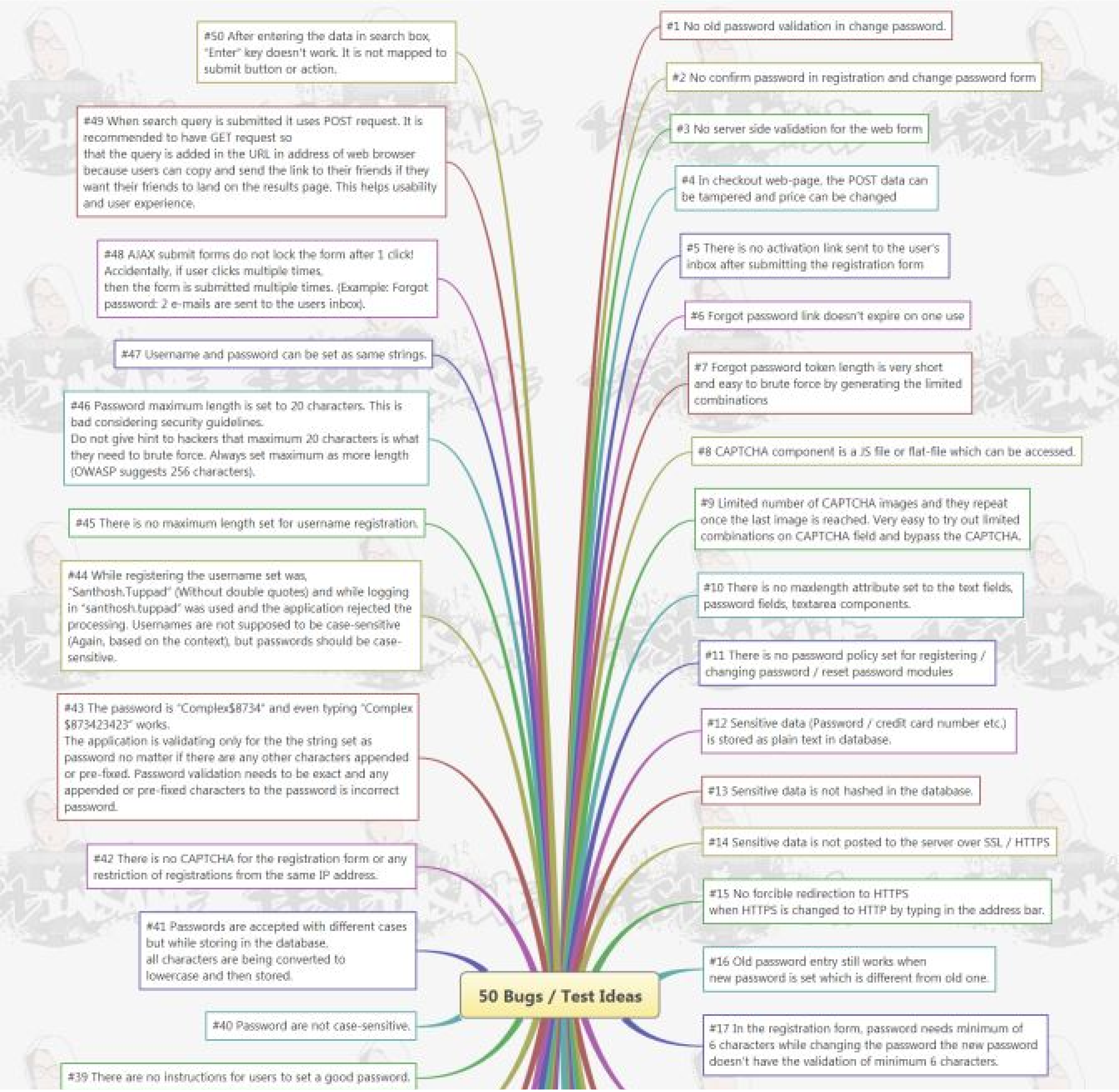
50 Bugs / Test Ideas



50 Bugs / Test Ideas

#50 After entering the data in search box, "Enter" key doesn't work. It is not mapped to submit button or action.

#49 When search query is submitted it uses POST request. It is recommended to have GET request so
that the query is added in the URL in address of web browser because users can copy and send the link to their friends if they want their friends to land on the results page. This helps usability and user experience.

#48 AJAX submit forms do not lock the form after 1 click! Accidentally, if user clicks multiple times,
then the form is submitted multiple times. (Example: Forgot password: 2 e-mails are sent to the users inbox).

#47 Username and password can be set as same strings.

#46 Password maximum length is set to 20 characters. This is bad considering security guidelines.
Do not give hint to hackers that maximum 20 characters is what they need to brute force. Always set maximum as more length (OWASP suggests 256 characters).

#45 There is no maximum length set for username registration.

#44 While registering the username set was, "Santhosh.Tuppad" (Without double quotes) and while logging in "santhosh.tuppad" was used and the application rejected the processing. Usernames are not supposed to be case-sensitive (Again, based on the context), but passwords should be case-sensitive.

#43 The password is "Complex$8734" and even typing "Complex $873423423" works.
The application is validating only for the the string set as password no matter if there are any other characters appended or pre-fixed. Password validation needs to be exact and any appended or pre-fixed characters to the password is incorrect password.

#42 There is no CAPTCHA for the registration form or any restriction of registrations from the same IP address.

#41 Passwords are accepted with different cases but while storing in the database.
all characters are being converted to lowercase and then stored.

#40 Password are not case-sensitive.

#39 There are no instructions for users to set a good password.

#1 No old password validation in change password.

#2 No confirm password in registration and change password form

#3 No server side validation for the web form

#4 In checkout web-page, the POST data can be tampered and price can be changed

#5 There is no activation link sent to the user's inbox after submitting the registration form

#6 Forgot password link doesn't expire on one use

#7 Forgot password token length is very short and easy to brute force by generating the limited combinations

#8 CAPTCHA component is a JS file or flat-file which can be accessed.

#9 Limited number of CAPTCHA images and they repeat once the last image is reached. Very easy to try out limited combinations on CAPTCHA field and bypass the CAPTCHA.

#10 There is no maxlength attribute set to the text fields, password fields, textarea components.

#11 There is no password policy set for registering / changing password / reset password modules

#12 Sensitive data (Password / credit card number etc.) is stored as plain text in database.

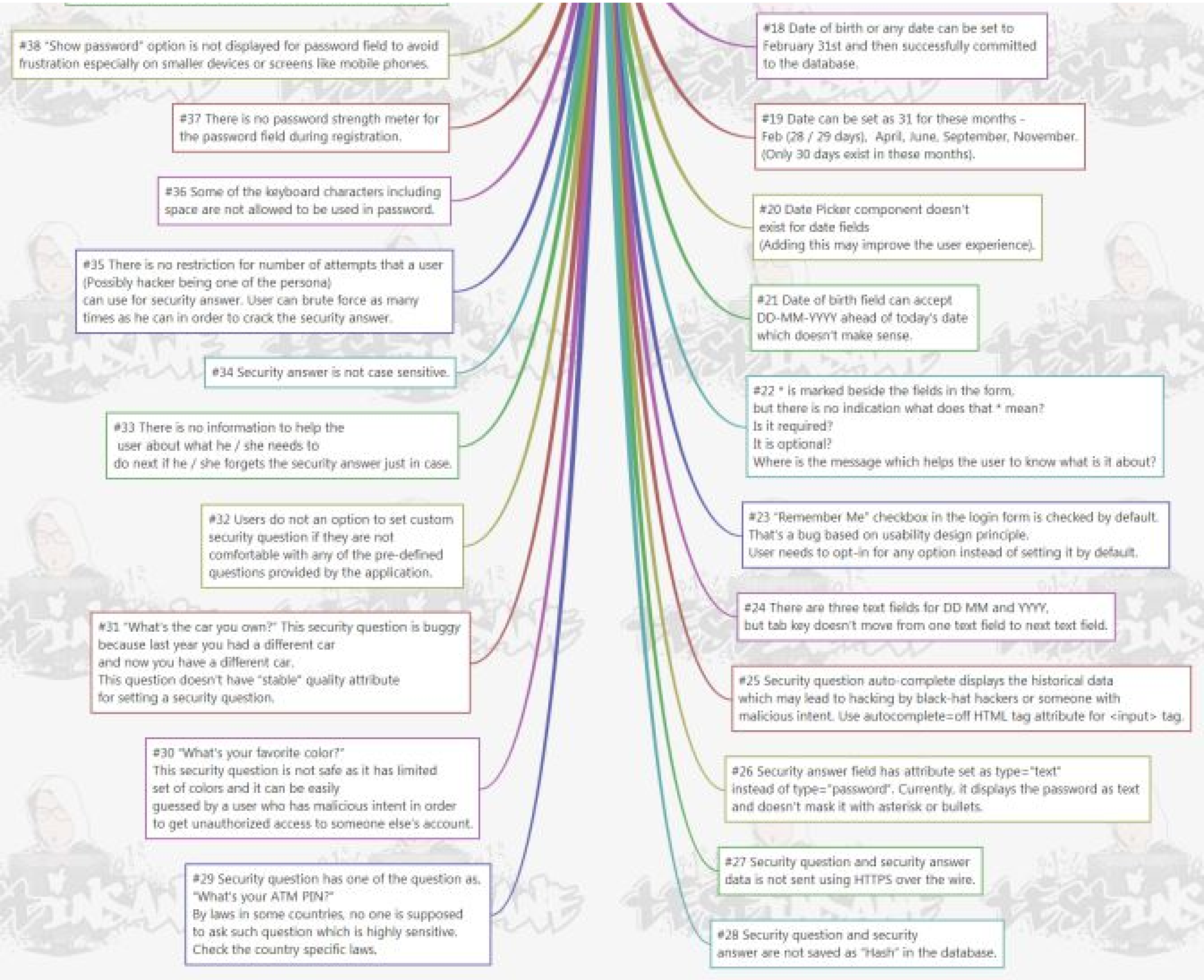
#13 Sensitive data is not hashed in the database.

#14 Sensitive data is not posted to the server over SSL / HTTPS

#15 No forcible redirection to HTTPS
when HTTPS is changed to HTTP by typing in the address bar.

#16 Old password entry still works when
new password is set which is different from old one.

#17 In the registration form, password needs minimum of 6 characters while changing the password the new password doesn't have the validation of minimum 6 characters.



#38 "Show password" option is not displayed for password field to avoid frustration especially on smaller devices or screens like mobile phones.

#37 There is no password strength meter for the password field during registration.

#36 Some of the keyboard characters including space are not allowed to be used in password.

#35 There is no restriction for number of attempts that a user
(Possibly hacker being one of the persona)
can use for security answer. User can brute force as many times as he can in order to crack the security answer.

#34 Security answer is not case sensitive.

#33 There is no information to help the
user about what he / she needs to
do next if he / she forgets the security answer just in case.

#32 Users do not an option to set custom security question if they are not comfortable with any of the pre-defined questions provided by the application.

#31 "What's the car you own?" This security question is buggy
because last year you had a different car
and now you have a different car.
This question doesn't have "stable" quality attribute
for setting a security question.

#30 "What's your favorite color?"
This security question is not safe as it has limited
set of colors and it can be easily
guessed by a user who has malicious intent in order
to get unauthorized access to someone else's account.

#29 Security question has one of the question as.
"What's your ATM PIN?"
By laws in some countries, no one is supposed
to ask such question which is highly sensitive.
Check the country specific laws.

#18 Date of birth or any date can be set to February 31st and then successfully committed to the database.

#19 Date can be set as 31 for these months -
Feb (28 / 29 days), April, June, September, November.
(Only 30 days exist in these months).

#20 Date Picker component doesn't
exist for date fields
(Adding this may improve the user experience).

#21 Date of birth field can accept DD-MM-YYYY ahead of today's date which doesn't make sense.

#22 * is marked beside the fields in the form,
but there is no indication what does that * mean?
Is it required?
It is optional?
Where is the message which helps the user to know what is it about?

#23 "Remember Me" checkbox in the login form is checked by default.
That's a bug based on usability design principle.
User needs to opt-in for any option instead of setting it by default.

#24 There are three text fields for DD MM and YYYY,
but tab key doesn't move from one text field to next text field.

#25 Security question auto-complete displays the historical data
which may lead to hacking by black-hat hackers or someone with
malicious intent. Use autocomplete=off HTML tag attribute for <input> tag.

#26 Security answer field has attribute set as type="text"
instead of type="password". Currently, it displays the password as text
and doesn't mask it with asterisk or bullets.

#27 Security question and security answer
data is not sent using HTTPS over the wire.

#28 Security question and security
answer are not saved as "Hash" in the database.

#1 No old password validation in change password.

#2 No confirm password in registration and change password form

#3 No server side validation for the web form

#4 In checkout web-page, the POST data can be tampered and price can be changed

#5 There is no activation link sent to the user's inbox after submitting the registration form

#6 Forgot password link doesn't expire on one use

#7 Forgot password token length is very short and easy to brute force by generating the limited combinations

#8 CAPTCHA component is a JS file or flat-file which can be accessed.

#9 Limited number of CAPTCHA images and they repeat once the last image is reached. Very easy to try out limited combinations on CAPTCHA field and bypass the CAPTCHA.

#10 There is no maxlength attribute set to the text fields, password fields, textarea components.

#11 There is no password policy set for registering / changing password / reset password modules

#12 Sensitive data (Password / credit card number etc.) is stored as plain text in database.

#13 Sensitive data is not hashed in the database.

#14 Sensitive data is not posted to the server over SSL / HTTPS

#15 No forcible redirection to HTTPS when HTTPS is changed to HTTP by typing in the address bar.

#16 Old password entry still works when new password is set which is different from old one.

#17 In the registration form, password needs minimum of 6 characters while changing the password the new password doesn't have the validation of minimum 6 characters.

#18 Date of birth or any date can be set to February 31st and then successfully committed to the database.

#19 Date can be set as 31 for these months - Feb (28 / 29 days), April, June, September, November. (Only 30 days exist in these months).

#20 Date Picker component doesn't exist for date fields (Adding this may improve the user experience).

#21 Date of birth field can accept DD-MM-YYYY ahead of today's date which doesn't make sense.

#22 * is marked beside the fields in the form, but there is no indication what does that * mean? Is it required? It is optional? Where is the message which helps the user to know what is it about?

#23 "Remember Me" checkbox in the login form is checked by default. That's a bug based on usability design principle. User needs to opt-in for any option instead of setting it by default.

#24 There are three text fields for DD MM and YYYY, but tab key doesn't move from one text field to next text field.

#25 Security question auto-complete displays the historical data which may lead to hacking by black-hat hackers or someone with malicious intent. Use autocomplete=off HTML tag attribute for <input> tag.

#26 Security answer field has attribute set as type=”text” instead of type=”password”. Currently, it displays the password as text and doesn’t mask it with asterisk or bullets.

#27 Security question and security answer data is not sent using HTTPS over the wire.

#28 Security question and security answer are not saved as “Hash” in the database.

#29 Security question has one of the question as, “What’s your ATM PIN?” By laws in some countries, no one is supposed to ask such question which is highly sensitive. Check the country specific laws.

#30 “What’s your favorite color?” This security question is not safe as it has limited set of colors and it can be easily guessed by a user who has malicious intent in order to get unauthorized access to someone else’s account.

#31 “What’s the car you own?” This security question is buggy because last year you had a different car and now you have a different car. This question doesn’t have “stable” quality attribute for setting a security question.

#32 Users do not an option to set custom security question if they are not comfortable with any of the pre-defined questions provided by the application.

#33 There is no information to help the user about what he / she needs to do next if he / she forgets the security answer just in case.

#34 Security answer is not case sensitive.

#35 There is no restriction for number of attempts that a user (Possibly hacker being one of the persona) can use for security answer. User can brute force as many times as he can in order to crack the security answer.

#36 Some of the keyboard characters including space are not allowed to be used in password.

#37 There is no password strength meter for the password field during registration.

#38 "Show password" option is not displayed for password field to avoid frustration especially on smaller devices or screens like mobile phones.

#39 There are no instructions for users to set a good password.

#40 Password are not case-sensitive.

#41 Passwords are accepted with different cases but while storing in the database, all characters are being converted to lowercase and then stored.

#42 There is no CAPTCHA for the registration form or any restriction of registrations from the same IP address.

#43 The password is "Complex$8734" and even typing "Complex$873423423" works. The application is validating only for the the string set as password no matter if there are any other characters appended or pre-fixed. Password validation needs to be exact and any appended or pre-fixed characters to the password is incorrect password.

#44 While registering the username set was, "Santhosh.Tuppad" (Without double quotes) and while logging in "santhosh.tuppad" was used and the application rejected the processing. Usernames are

not supposed to be case-sensitive (Again, based on the context), but passwords should be case-sensitive.

#45 There is no maximum length set for username registration.

#46 Password maximum length is set to 20 characters. This is bad considering security guidelines. Do not give hint to hackers that maximum 20 characters is what they need to brute force. Always set maximum as more length (OWASP suggests 256 characters).

#47 Username and password can be set as same strings.

#48 AJAX submit forms do not lock the form after 1 click! Accidentally, if user clicks multiple times, then the form is submitted multiple times. (Example: Forgot password: 2 e-mails are sent to the users inbox).

#49 When search query is submitted it uses POST request. It is recommended to have GET request so that the query is added in the URL in address of web browser because users can copy and send the link to their friends if they want their friends to land on the results page. This helps usability and user experience.

#50 After entering the data in search box, “Enter” key doesn’t work. It is not mapped to submit button or action.