Login Example

- Test Ideas
  - Checkable aspects
    - Username field
      - What is the set of acceptable characters?
      - What characters are acceptable, but must be escaped?
      - What characters are completely unacceptable?
        - If we're testing for these, test for one at a time, not for blanket rejection
      - If the username is an email address
        - Is there filtering for that?
        - automatically up- or down-cased?
      - Minimum and maximum lengths
    - Password field
      - Can it be revealed at the user's option?
      - What is the acceptable set of characters?
      - Which characters are unacceptable?
      - Case sensitivity?
      - Minimum and maximum lengths
    - Captcha
      - Image/text
      - Audio
      - Is it on a separate server?
    - Combinations of valid and invalid usernames/passwords/CAPTCHAs/security questions
      - Huge input data
      - Zero input data
      - Minimum length
      - Maximum length
      - Character assortments
    - Submit button
    - Is there a testable (scriptable) interface here?
  - Testing required
    - Does the system properly block lapsed accounts?
    - Does the system entitle the user to more than it should?
    - Are there appropriate lockout warnings?
      - Login attempt when locked out
    - What is the lockout strategy, if any?
    - Login after a timed-out session
    - Login after a rejected login
- Product Elements
  - Structure
    - Is there a flow model or diagram anywhere?
    - Authentication database
  - Function
    - Obtain credentials
      - User ID
      - Password
      - Other credential information?
    - Send credentials to server
    - Compare credentials to authentication database
  - Data
    - What information is gathered and tracked?
      - About the user
      - About individual login activity?
      - In aggregate
    - Rights
    - Access tokens
      - Token ID
      - Login session ID
      - User ID
      - Group identifiers
      - Restricting group identifiers
      - Privileges
    - Default owner, primary group, access control list for objects created by the subject associated with the token
  - Interfaces
    - Command line?
    - Browser?
    - Other GUI?
      - e.g. OAUTH
      - Through a third party?
  - Platform
    - Operating systems
    - User database
  - Operations
  - Time
- What do I know about?
  - my mission?
  - my context?
    - Am I the only tester on the project? The only tester focused on this feature? Am I qualified?
  - logging in?
    - What happens before, during, and after the process on the client?
    - What happens before, during and after the process on the server?
    - How do the client and server communicate?
      - What's the protocol?
        - Is it subject to some kind of standard?
        - Where can I learn about the protocol?
      - What encryption is in use?
      - You're not really using GET, are you?
    - What happens on intermediate systems?
      - Are simultaneous logins permitted?
        - Does the system preserve separate states for each one?
        - Does the user have access to resources across two sessions?
    - Cookies
      - Are cookies being used?
      - Where are they stored?
      - What do they contain?
      - Are their contents private? Should they be?
    - Connection to "external" systems
    - See "Product Elements" of the HTSM
  - this product?
    - Are there other systems accessible via single sign-on (SSO?)
      - Is the user able to access everything that he should?
      - Is the user prevented from accessing the things that he shouldn't?
  - related technologies?
    - See standards documents, RFCs, etc.
      - Bearer Token Usage
  - significant risks?
    - Review security literature for login-related problems
      - XSS, CSRF?
      - SQL Injection?
      - HTTP header injection?
      - Email injection?
- Risks
  - Authorised user will not be able to log in
    - Forgot username?
    - Forgot password?
    - Forgot both?
    - New user?
  - User may not have access to permitted resources
  - User may have access to forbidden resources
  - Unauthorised user will be able to log in
    - Hacker will be able to log in
      - Brute force approach
      - Identifiable vulnerability
    - Previously authorised user will be able to log in
  - Hacker will be able to get access to the system while evading the login process, or some aspect of it
  - System is vulnerable to some form of attack (other than unauthorized login) by exploiting vulnerabilities in the login interface
  - Username/password will be intercepted
  - Some version of the software will be out of date and unpatched or otherwise vulnerable
  - Login fields will be vulnerable to known security problems
    - SQL injection
    - XSS
    - CSRF
  - Error messages will reveal useful information to hackers
  - User will be fooled into revealing credentials
  - Username/password/personal information will be leaked via vulnerability
  - Logout or unsuccessful login will leave system in an unstable or insecure state
  - It may be possible to swamp the system with
    - simultaneous logins
    - dropped connections that are not cleaned up
    - going beyond limits of account management mechanisms
      - account registrations
  - Single sign-on issues
    - User may not be able to access other systems or organizations
  - Back doors of which we may be unaware
  - Record of user access will not be retained
  - Failed attempts to login will not be retained
    - Especially important WRT to tracking hack attempts
  - Site looks sufficiently amateurish that user won't risk a login attempt (original Verified by Visa was like this)
  - Rob Sab's thread on multiple simultaneous logins, logged-in sessions; changing rights.
- Issues
  - What is the test scope here?
    - Login is a process that happens on at least two machines, not just one.
    - To what degree are we monitoring the server?
    - What happens on intermediate machines?
      - Could login data be intercepted?
      - Are username and password encrypted?
    - How many factors?
      - Email only?
      - User ID only?
      - Both accepted?
      - Passworded?
      - Two-factor authentication?
      - Encryption?
    - Single sign-on?
      - OAUTH?
        - Will this be authenticating with another system (like those Twitter add-ins)?
  - When does a "logging in" test finish?
    - How do we determine when the user is logged in and the system is ready for use?
      - When a message box is displayed?
      - Has the login been recorded?
        - What information is kept in the record?
      - Other tests or checks?
    - What state should the system be in?
      - User correctly authenticated?
      - User appropriately authorized?
      - Login scripts finished?
        - Is anything being updated or pushed at login time?
          - Okay, not this time. Ever?
      - Login recorded in the access logs?
    - Does the server log the access?
      - Is that happening?
      - Where does the server store the logs?
      - What do the logs look like?
        - Are they well-structured, easily machine-readable?
      - Is the log secure?
      - Does the log contain sensitive information?
      - What happens under lots of load?
    - Are there extra security questions?
      - What triggers them?
      - What form do they take?
      - What might not get cleaned up if a user can get here but can't get past here?
  - Is there a testable interface available?
    - What happens in the GUI or browser, above that interface?
  - Has anyone else tested this?
    - Can that testing be trusted?
  - Are there limits on the number of logged-in users?
  - Are there limits on invalid login attempts?
  - How is the state of a logged-in user identified and maintained?
  - How about the state of an almost-logged-in user?
  - Character-based or field-based validation?
  - Where can login information be added, modified, and deleted?
    - Are validation routines consistent for all of these and for the login process?
- Quality Criteria
  - Capability
  - Reliability
  - Usability
    - Accessibility
      - Is the process accessible to
        - the blind
        - the deaf
        - limited mobility
  - Charisma
    - If there are requirements or restrictions on characters, can we see them up front, please?
    - Are error messages
      - unclear?
      - hostile on error?
      - failing to point to a resolution?
      - overly informative?
  - Security
    - Standard security issues
      - Confidentiality
      - Integrity
      - Authentication
      - Authorization
      - Availability
      - Non-repudiation
        - Non-repudiation is about making sure that the user cannot successfully claim "I didn't do it!"
        - Therefore: what record is being kept of attempted logins?
    - OWASP Top 10
  - Scalability
  - Compatibility
  - Performance
  - Installability
  - Development
    - Supportability
    - Testability
    - Maintainability
    - Portability
    - Localizability

This is an example; a set of rough notes from a one-hour brainstorm on "how would I test logging in". It's intended as reply to those examples of "testing a login page by checking for the 'you are now logged in' message".

Michael Bolton
2018