```
         _         _                 _          _          _            _          _           _
        / /\      /\_\              / /\       /\ \       /\ \     _   /\ \       / /\        / /\
       / /  \    / / /         _   / /  \      \ \ \     /  \ \   /\_\/  \ \     / /  \      / /  \
      / / /\ \   \ \ \__      /\_\/ / /\ \__   /\ \_\   / /\ \ \_/ / // /\ \ \   / / /\ \__  / / /\ \__
     / / /\ \ \   \ \___\    / / // / /\ \___\/ /\/_/  / / /\ \___/ // / /\ \_\ / / /\ \___\/ / /\ \___\
    / / /\ \_\ \   \__  /   / / / \ \ \ \/___/ / /    / / /  \/____// /_/_ \/_/ \ \ \ \/___/\ \ \ \/___/
   / / /\ \ \___\  / / /   / / /   \ \ \   / / /     / / /    / / // /____/\     \ \ \        \ \ \
  / / /  \ \ \__/ / / /   / / /_    \ \ \ / / /     / / /    / / // /\____\/  _   \ \ \   _    \ \ \
 / / /____\_\ \  / / /___/ / //_/\__/ / // / /___  / / /    / / // / /______ /_/\__/ / /  /_/\__/ / /
/ / /__________\/ / /____\/ / \ \/___/ //\__\/_/___\/ / /    / / // / /_______\\ \/___/ /   \ \/___/ /
\/_____________/\/_________/   \_____\/ \/_________/\/_/     \/_/ \/__________/ \_____\/     \_____\/

         _            _           _              _          _
        _\ \         /\ \        /\ \           /\ \       /\ \
       /\__ \       /   \ \     /   \ \         \ \ \     /   \ \
      / /_ \_\     / /\ \ \    / /\ \_\         /\ \_\   / /\ \ \
     / / /\/_/    / / /\ \ \  / / /\/_/        / /\/_/  / / /\ \ \
    / / /        / / /  \ \_\ / / / ______    / / /    / / /  \ \_\
   / / /        / / /   / / // / / /\______\  / / /    / / /    \/_/
  / / / ____   / / /   / / // / /  \/_____ /  / / /    / / /
 / /_/_/ ___/\ / / /___/ / // / /_____/ / / /___/ / /__  / / /________
/_______/\__\// / /____\/ // / /______\/ //\__\/_/___\/ / /_________\
\_______\/    \/_________/ \/___________/ \/_________/\/____________/

         _                _         _           _                  _           _           _
        / /\             /\ \      /\ \        / /\               /\ \        /\_\        / /\
       / /  \            \_\ \     \_\ \      / /  \             /  \ \      / / /   _   / /  \
      / / /\ \           /\__ \    /\__ \    / / /\ \           / /\ \ \    / / /   /\_\/ / /\ \__
     / / /\ \ \         / /_ \ \  / /_ \ \  / / /\ \ \         / / /\ \ \  / / /___/ / // / /\ \___\
    / / /  \ \ \       / / /\ \ \/ / /\ \ \/ / /  \ \ \       / / /  \ \_\/ / /\______/ \ \ \ \/___/
   / / /___/ /\ \     / / /  \/_// / /  \/_// / /___/ /\ \     / / /    \/_/ / /\______/   \ \ \
  / / /_____/ /\ \   / / /      / / /      / / /_____/ /\ \   / / /        / / / \ \ \  _   \ \ \
 / /_________/\ \ \ / / /      / / /      / /_________/\ \ \ / / /________/ / /   \ \ \/_/\__/ / /
/ / /_       __\ \_\/_/ /      /_/ /      / / /_       __\ \_\/ / /_________\/ / /    \ \ \\ \/___/ /
\_\___\     /____/_/\_\/       \_\/       \_\___\     /____/_/\/____________/\/_/      \_\_\\_____\/
```

What is Business Logic?

- Business logic is a non-technical term generally used to describe the functional algorithms which handle information exchange between a database and a user interface. It is distinguished from input/output data validation and product logic. Business logic models real life business objects such as accounts, loans, itineraries, and inventories. It also prescribes how business objects interact with one another. Finally, business logic enforces the routes and the methods by which business objects are accessed and updated.
- Business logic comprises:
  - Business rules that express business policy (such as channels, location, logistics, prices, and products); and
  - Workflows that are the ordered tasks of passing documents or data from one participant (a person or a software system) to another."

What is a Business Logic Attack?

- A business logic attack is an attack on a business logic vulnerability.
- “Business logic vulnerability is one that allows the attacker to misuse an application by circumventing the business rules. Most security problems are weaknesses in an application that result from a broken or missing security control (authentication, access control, input validation, etc...). By contrast, business logic vulnerabilities are ways of using the legitimate processing flow of an application in a way that results in a negative consequence to the organization.”

  - OWASP

Simple Abstract Example

- There is a process that was developed with the intention of users going through step 1, then 2, then 3 in that order
- What if you go from step 1 straight to step 3?

Simple Real World Example

- A user is able to start a transaction linked to their loyalty account and the points are added to the account during checkout.
- The user goes through checkout but right at the end cancels out and their credit card isn’t charged
- The loyalty points were added during checkout, not after so the user gets the points for free

Simple Web Example

- Retrieving a profile
  - For example, Jack’s profile can be fetched with id=1001
  - If this value changed to 1089 we get another user’s information
  - Note: A scanner may go on and change the value from 1001 to ‘1001 to find SQL injection, but not to 1089 and would miss that the application is vulnerable to authorization bypass
- Note: this type of vulnerability exists and was exploited at a bank

Identification

- Controls are used to shape user interaction and enforce business rules.
- Identify and analyze controls to make sure they are:
  - In place to implement business rules
  - Implemented correctly and cannot be bypassed or tampered with
  - Used properly in all the necessary places

Testing

- Test business logic data validation
  - Verify that the application does not allow users to insert unvalidated data into the system
- Test ability to forge requests
  - Verify that the application does not allow users to submit or alter data to any component of the system that they should not have access to
- Test integrity checks
  - Verify that the application does not allow users to destroy the integrity of any part of the system or its data

Testing

- Test for process timing
  - Verify that the application does not allow users to manipulate a system or guess its behavior based on input or output timing
- Test limits on the number of times a function can be used
  - Verify that the application does not allow users to exercise portions of the application or its functions more times than required by the business process
- Test for the circumvention of work flows
  - Verify that the application does not allow users to perform actions outside of the “approved/required” business process flow

Testing

- Test defenses against application misuse
  - Verify that the application does not allow users to manipulate the application in an unintended manner
- Test upload of unexpected file types
  - Verify that the application does not allow users to upload file types that the system is not expecting or wanted per the business logic requirements
- Test upload of malicious files
  - Verify that the application does not allow users to upload files to the system that are malicious or potentially malicious to the system security

More Real Examples

- Winning an Online Auction
- "Interactive" T.V.
- Free MacWorld 2007 Platinum Passes
- Day Trading Contest
- Password Recovery
- Twitter Financial Details
- Burning Man Tickets

Real World Example

- Day Trading Contest
- Description
  - A fake stock trading platform allows players to use mock money to "buy" stocks based on their real market prices. In the game a user picks a stock, a number of shares, and sends a request to the system. The system then calculates the order price based on the current stock price and number of shares. This total is then sent back to the user to confirm the page. When they accept the trade the application executes based on the previously calculated information.
  - An attacker can watch for an earnings release and prior to it submit a buy and/or sell trade for the stock. After the system calculates the trade and requests the user to verify they simply leave the window open. When the earnings report is realeased and the stock goes up or down the user then only verifies the trade that earns them money. Additionally the trade is executed based on the pre-calculated price before the earnings report, earning the user even more money.

Real World Example

- Solution
  - When executing a trade the system should always calculate price based on the current share price.
  - A trade verification should have an expiration window of only 5 minutes or less.

Resources/References

- https://www.owasp.org/index.php/Business_Logic_Security_Cheat_Sheet
- https://www.owasp.org/index.php/Testing_for_business_logic_(OWASP-BL-001)
- https://www.whitehatsec.com/assets/WP_bizlogic092407.pdf