



# Business Logic Attacks – Bots and BATs

Eldad Chai  
Web Research Team Leader  
Imperva

[eldad@imperva.com](mailto:eldad@imperva.com)

OWASP  
EU09 Poland

Copyright © The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the OWASP License.



The OWASP Foundation  
<http://www.owasp.org>

# Agenda

- Business Logic Attacks (BATs) –  
The Whats and Whys
- Classes of BATs
- BATs and Bots
- Mitigating BATs
- Summary
- Q&A



# Business Logic Attacks (BATs) – The Whats and Whys



# Business Logic Attacks (BATs) – The Whats and Whys

## ■ What a BAT is / is not?

Is not	Is
Malformed requests	Legitimate requests
Illegal input values	Legitimate input values
Usually a single request	Usually multiple requests
Changing a module's functionality to inflict damage	Abusing a module's functionality to inflict damage
Direct damage to the application	Direct damage to the business
Indirect damage to the business	Indirect damage to the application



# Business Logic Attacks (BATs) – The Whats and Whys

## ■ Why BATs?

- ▶ Technical types of attacks are gaining more awareness
- ▶ Better coding practices and tools at the development phase
- ▶ In some cases an entire class of vulnerabilities was eliminated out of the box
- ▶ Better assessment tools for detecting vulnerabilities
- ▶ Better tools for detecting and blocking attacks



# Business Logic Attacks (BATs) – The Whats and Whys

## ■ So... why BATs?

- ▶ Require less understanding of the underlying technology
- ▶ More targets / opportunities out there
- ▶ Larger gain
- ▶ Can be “amplified” through automation



# Business Logic Attacks (BATs) – The Whats and Whys

- What damage can BATs do?
  - ▶ Financial damage
  - ▶ Damage to reputation
  - ▶ DoS, EDoS/FDoS
  - ▶ QoS degradation
  - ▶ Data leakage



# Classes of BATs

## ■ Classifying flaws

- ▶ Numerous examples of business logic flaws
- ▶ Both the business logic and its implementation might have flaws

## ■ Classifying attacks

- ▶ The attack techniques are not so diverse
- ▶ Mainly because automation is usually required
- ▶ Automation techniques have much in common



# Abuse of Functionality – Brute Force

- Multiple invocations of transactions with random or semi-random data
- Usually includes repeating a request to a certain module with the intention of guessing valid values



# Abuse of Functionality – Brute Force

## ■ Solution requirements

### ▶ Detection

- Identify failed transactions
- Measure the rate of failed transactions in real time
- Define acceptable thresholds for the rate of failed transactions in varying contexts: IP, session, user, system-wide, etc.

### ▶ Mitigation

- Anti-automation
  - Validate human interaction
  - Forced delays
- Reduce the effective rate of attacks to the point that they become futile



# Abuse of Functionality – Repetition Attacks

- Multiple invocations of requests with valid data
- Include repeating requests to a certain module with the intention of invoking a large number of business transactions

The screenshot shows a user profile for 'MagicOPromotion Genesis' with details: 'Joined: 29 Apr 2009', 'Posts: 2', and 'Location: England'. Below the profile is a promotional banner for '\$25 JUST FOR OPENING A NEW ACCOUNT?' with a green 'APPLY NOW' button. The banner also includes the text 'Open in Minutes' and a clock icon.

Rank	Name	Avg. Rating	Total Votes
1	Moot	90	16,794,368
2	Anwar Ibrahim	47	2,316,378
3	Rick Warren	45	1,902,383
4	Baitullah Mehsud	45	1,902,162
5	Larry Brilliant	44	2,005,310
6	Eric Holder	43	1,808,663
7	Carlos Slim	41	1,852,506
8	Angela Merkel	41	1,634,488
9	Kobe Bryant	39	1,976,880
10	Evo Morales	39	1,477,789
11	Alexander Lebedev	38	824,073
12	Lil' Wayne	37	939,993
13	Sheikh Ahmed bin Zayed Al Nahyan	36	838,578
14	Odell Barnes	35	916,836
15	Tina Fey	33	897,045
16	Hu Jintao	32	928,400

## PLEASE READ THE IMPORTANT DISCLOSURES BELOW.

1 Annual Percentage Yield is effective and is subject to change. A \$1 minimum deposit is required to open a new account. Withdrawal limits apply. Online statements required.

2 \$25 will be credited to your Complete Savings Account within 30 days of the account being funded with a minimum deposit of \$1. Payments will be reported as interest income. Accounts must be opened by to qualify for the \$25 offer. Must be a new account opened with new funds. Offer applies to one new account per customer. Not good with any other offer. This offer is not valid for E\*TRADE FINANCIAL employees.



# Abuse of Functionality – Repetition Attacks

## ■ Solution requirements

### ▶ Detection

- Identify successful transactions
- Measure the rate of successful transactions in real time
- Define acceptable thresholds for the rate of successful transactions in varying contexts: IP, session, user, system-wide, etc.

### ▶ Mitigation

- Anti-automation
  - Validate human interaction
  - Forced delays
- Reduce the effective rate of attacks to the point that they become futile



# Abuse of Functionality – Locking Business Resources



# Abuse of Functionality – Locking Business Resources

- Lock resources by doing nothing
- Invoke business transactions that (temporarily) consume / hold business resources
- Prevent the transactions from completing
- Resources are kept locked and pending, virtually unavailable to others



# Abuse of Functionality – Locking Business Resources

## ■ Solution requirements

### ▶ Detection

- Identify unresolved transactions / incomplete business flows
- Define acceptable thresholds for holding transactions in an undecided state, before they become “stale”
- Measure the quantity of “stale” transactions
- Define acceptable thresholds for the number of “stale” transactions in varying contexts

### ▶ Mitigation

- Enforcing number and time length restrictions
- Gracefully rollback “stale” transactions / business flows



# Application Flow Bypass

- Bypass or circumvent the intended flow of an application
- This includes invoking transactions out of their intended order



# Application Flow Bypass

## ■ Solution requirements

### ▶ Detection

- Identify a set of transactions that form a logical flow
- Define an acceptable order of transactions in a flow

### ▶ Mitigation

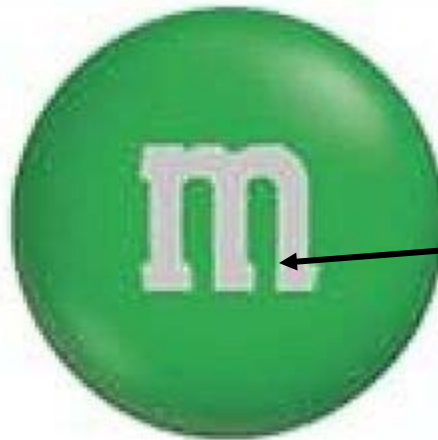
- Gracefully rollback transactions / business flows that do not comply with the flow definition



# Access Control Bypass

- Gaining access to parts of the application that otherwise would require increased access control
- Classic M&M security model

Hard from the outside



Soft from the inside



# Access Control Bypass

## ■ Solution requirements

### ▶ Detection

- Identify application users and their roles
- Associate roles with privileged transactions
- Identify access to privileged transactions by an unprivileged user

### ▶ Mitigation

- Deny access from unprivileged users



# Web Leeching

## ■ Leeching on business resources

- ▶ Contact forms (email spam)
- ▶ Sharing interfaces (comment spam)
- ▶ Redirect modules (phishing scams)
- ▶ Hotlinking (bandwidth leeching)
- ▶ Resource scraping (business data leeching)



# Web Leeching

## ■ Solution requirements

### ▶ Detection

- Identify textual abuse patterns (e.g., comment spam, email spam)
- Identify abusive sources (e.g., by Referer, by IP)
- Identify intensive automated access

### ▶ Mitigation

- Block abusive access
- Anti-automation
- Throttle down automated access



# BATs and Bots

- Automation's part of total attacks is growing
  - ▶ Spam bots
  - ▶ Email harvesters
  - ▶ RFI bots
  - ▶ SQLi bots
  - ▶ ...
- Automation is getting more sophisticated
  - ▶ Deployment techniques
  - ▶ Attack automation techniques



# BATs and Bots

- Online business transactions usually have a small effect on the business
- An attacker needs to invoke multiple transactions in order to have a significant effect
- Some examples of automated BATs:
  - ▶ Brute force
  - ▶ Repetition attacks
  - ▶ DoS (EDoS/FDoS)
  - ▶ Comment spam
  - ▶ ...



# Mitigating BATs

- First thing's first: How do you know that you are susceptible to BATs?
  - ▶ Use OWASP testing guide for business logic testing
  - ▶ Use assessment services
- Purely automated scans usually don't detect business logic flaws
  - ▶ Cannot differentiate between a positive and a negative business effect
  - ▶ Have a major disadvantage in assessing production environments



# Mitigating BATs

## ■ Validate human interaction

### ▶ CAPTCHA

- Generic
- Affects user experience
- Blocks legitimate bots

### ▶ Second level authentication

- Authentication based on personal data
- Validates specific user
- Requires additional data

### ▶ Integrated

- Requires prior knowledge of all validation points
- Hard to validate based on user behavior

### ▶ External

- Can be generic
- Might require changing application responses
- Easy to validate suspicious sessions only



# Mitigating BATs

## ■ Rolling back transactions

### ▶ Integrated

- Requires handling within several application modules
- Requires prior knowledge and definition of the protected flows and rollback sequences
- Implementation complexity is highly dependant on application architecture

### ▶ External

- Can be generic
- Easily integrated into existing detection and mitigation framework
- Might require additional session data
- Needs to be carefully designed and implemented



# Mitigating BATs

## ■ Real time measurements

### ▶ Integrated

- Requires prior knowledge of what is to be measured
- Added complexity to the application
- Might be more accurate when conditions cannot be detected by external tools

### ▶ External

- Generic
- Easily integrated into existing detection and mitigation framework
- Not everything can be measured externally



# Mitigating BATs

## ■ Force delays

### ▶ Integrated

- Require handling within several application modules
- In case of an attack, the application might be overwhelmed
- Not really a “classic” internal feature

### ▶ External

- Can be generic
- Easily integrated into existing detection and mitigation framework
- Might be too crude



# Mitigating BATs

- Implement the solution requirements
- Pre-emptive approach (integrated)
  - ▶ Implement solution requirements into the application code
  - ▶ For all existing applications and all new applications
  - ▶ Implementation can be tricky (measuring in real time, delays, abusive patterns, ...)
  - ▶ User experience is globally affected
- Alternative approach (external)
  - ▶ Use a Web Application Firewall
  - ▶ Change user experience only for suspicious sessions
  - ▶ Coordinated security across domains
  - ▶ Introduce fixes for crucial vulnerabilities with releases



# WAF Detection Requirements - BATs

- Identify business transactions
  - ▶ Define failure / success conditions for a request (user/WAF)
- Identify transaction flows
  - ▶ Define the set of transactions (user) and their order (user/WAF)
- Identify legitimate usage
  - ▶ Define thresholds for flows: number, time length (user/WAF)
  - ▶ Measure number, time length and rate in real time (WAF)
  - ▶ Multiple contexts: IP, session, user, system-wide (WAF)
- Identify access control policy
  - ▶ Define roles and associate users with roles (user)
  - ▶ Identify application users' transactions and compare against roles (WAF)



# WAF Detection Requirements - Bots

- Identify abuse patterns (WAF)
  - ▶ Signatures for textual patterns
  - ▶ Black lists for known malicious sources
  - ▶ Profile for unknown malicious sources
- Identify intensive automated access (WAF)
  - ▶ Malicious crawling patterns
  - ▶ Non-browser access
  - ▶ Automation patterns



# WAF Mitigation Requirements

- Block abuse access (easy...)
- Deny access from unprivileged users (easy...)
- Throttle down automated access (anti automation)
  - ▶ Human interaction validation (e.g., CAPTCHA)
  - ▶ Re-authentication
  - ▶ Delays
- Enforce business flows
  - ▶ Define rollback transactions for flows
  - ▶ Rollback transactions / business flows based on a pre-defined rollback sequence



# Summary

- The risk of business logic attacks is increasing
- Business logic attacks abuse the web application's functionalities
- Most classes of attacks require automation
- Detecting and mitigating BATs requires a new set of tools
- These tools can be implemented internally or externally
- WAFs can detect and mitigate BATs and Bots



# Thanks!

# Q&A

