Bot or Not?
Mitigating Automated Threats to Web Applications

Bastian Braun
mgm security partners
14 November 2017
about: me

• IT Security Consultant @ mgm (Software House)
• me:= mgm security partners
  • security support for web development teams
  • seminars & trainings
  • security audits
  • security workshops
  • product & market analyses
  • penetration testing
Background: Automation in the Web

- web communication = requests + responses
- stateless HTTP allows uncontrolled repetitions of previous requests
Background: Automation in the Web

• practical
  • easily expandable
  • more robust / fail safe than stateful communication
  • business logic scalable & movable (see Angular, React, …)
• problematic
  • (in-)secure workflows
    • control-flow integrity
  • automated actions
Threats by Automation

- registration
  - e.g. email accounts for spammers, newsletters, username enumeration
- login
  - e.g. password brute-forcing, user lock-out
- password reset
  - e.g. email flooding, username enumeration
- parameterized search queries
  - data harvesting
Detection

• depends on feature logic
• approaches
  • detect massive requests from same IP
    • requires threshold → evade by spreading
  • generate client fingerprint to identify source
    • no fingerprint → suspicious
    • spoofed fingerprints → sanity check
  • device cookies
  • require authentication (login) before granting access
    • protect registration & login
Countermeasures: Theory

• CAPTCHAs
• additional knowledge
• tarpit
• SMS TANs
• proof-of-work systems
• IP locks
• user locks
## Countermeasures: Practice

<table>
<thead>
<tr>
<th>Countermeasure</th>
<th>Practical Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPTCHAs</td>
<td>annoying, bad usability, breakable</td>
</tr>
<tr>
<td>additional knowledge</td>
<td>annoying</td>
</tr>
<tr>
<td>tarpit</td>
<td>susceptible to DoS attacks, temporary user lockout</td>
</tr>
<tr>
<td>SMS TANs</td>
<td>automated triggers</td>
</tr>
<tr>
<td>proof-of-work systems</td>
<td>hard to scale</td>
</tr>
<tr>
<td>IP locks</td>
<td>false positives / collateral damage if NAT</td>
</tr>
<tr>
<td>user locks</td>
<td>massive user-lock out</td>
</tr>
</tbody>
</table>
## Countermeasures: Applicability

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Appropriate Detection</th>
<th>Applicable Anti-Automation</th>
<th>Unsuitable Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>Client IP, Client Fingerprint</td>
<td>CAPTCHA, Proof-of-Work, IP Locks</td>
<td>Additional Knowledge, Tarpit, SMS TAN, User Locks</td>
</tr>
<tr>
<td>Password Reset</td>
<td>Client IP, Client Fingerprint, Device Cookie</td>
<td>CAPTCHA, Additional Knowledge, SMS TAN, Proof-of-Work, IP Locks</td>
<td>Tarpit, User Locks</td>
</tr>
<tr>
<td>Login</td>
<td>Client IP, Client Fingerprint, Device Cookie</td>
<td>Additional Knowledge, Tarpit, SMS TAN, Proof-of-Work, IP Locks</td>
<td>CAPTCHA</td>
</tr>
<tr>
<td>Contact Form</td>
<td>Client IP, Client Fingerprint (Device Cookie, Authentication)</td>
<td>CAPTCHA, Proof-of-Work, IP Locks</td>
<td>Additional Knowledge, Tarpit, SMS TAN, User Locks</td>
</tr>
<tr>
<td>Newsletter Registration</td>
<td>Client IP, Client Fingerprint, Device Cookie (Authentication)</td>
<td>CAPTCHA, Proof-of-Work, IP Locks</td>
<td>Additional Knowledge, Tarpit, SMS TAN, User Locks</td>
</tr>
<tr>
<td>Parameterized Search Queries</td>
<td>Client IP, Client Fingerprint, Device Cookie, Authentication</td>
<td>Proof-of-Work, IP Locks</td>
<td>CAPTCHA, Additional Knowledge, Tarpit, SMS TAN, User Locks</td>
</tr>
</tbody>
</table>
Conclusion

• open issues
  • how to protect machine-2-machine APIs?
  • how to distinguish competitors from Google?
    • e.g. prevent automatic price analysis by competitors
      vs give Google crawler access to portfolio
• user acceptance still the biggest problem
• awareness during development processes often low