CSE4028 report

Advanced Cybersecurity - Lab

18BCE7147
V. Hari Praneeth
Lab L57+58 slot

Problem Identification

- The problem identified is

  Brute-force attack using bots

- This has 2 subproblems - brute-force attack and bot
- These attacks are performed everywhere including account passwords, SSH accounts, database admin accounts, and WordPress sites.

How they work

- Bots are automated programs that perform a specified action.
- Brute-force attack uses trial-and-error to guess login info (both username and password). Hackers work through all possible combinations hoping to guess correctly.

    Like abcdef_123, abcdeg_123, abcdeh_123, ...

Detection

- Backend algorithms can detect both of them
- Bot detection can be done easily using user-agent, and the behaviour. We can enable captcha to prevent bots.
- **Brute force Detection**: For a username, if a similar password is entered 4 times in a row, it is a brute force attempt.
- A dictionary-based attack can be detected by storing dictionary files
- On detection, the user password should be reset automatically

Prevention

- If multiple brute force attempts are detected from the same IP address, the IP should be blocked
- Multi-Factor Authentication(MFA) helps prevent unauthorized login
- In SSH, Databases, and WordPress, default username or password should not be used.
- Secure password
  - User should be forced to have a strong password during registration.
  - Password must contain various characters including numbers.
  - Length should be at least 9.
  - Passwords should not be commonly used

Approach for Brute force detection (not full code)

```
def block(ip_addr):
    blocked_ips.add(ip_addr)
    print("Blocked IP: " + ip_addr)

def is_brute_force(ip_addr):
    ' Detect whether the request is brute force or not '

    # Find how many failed attempts from same IP
    if failed_attempts.get(ip_addr, 0) > attempts_limit:
        block(ip_addr)
        return True  # yes. it is brute-force
    else:
        return False

@app.route('/', methods = ['POST', 'GET'])
def home():
    ip_address = request.remote_addr
    if ip_address in blocked_ips:
        return 'Your IP is blocked'

    user_agent = request.headers.get('User-Agent')
    if not user_agent.startswith('Mozilla'):
        return 'Bot detected. This website is not for bots'
```

Approach for Bot detection

- Check User-agent. User-agent of browsers start with ‘Mozilla/’

  If it is not a browser, we detect as a bot.

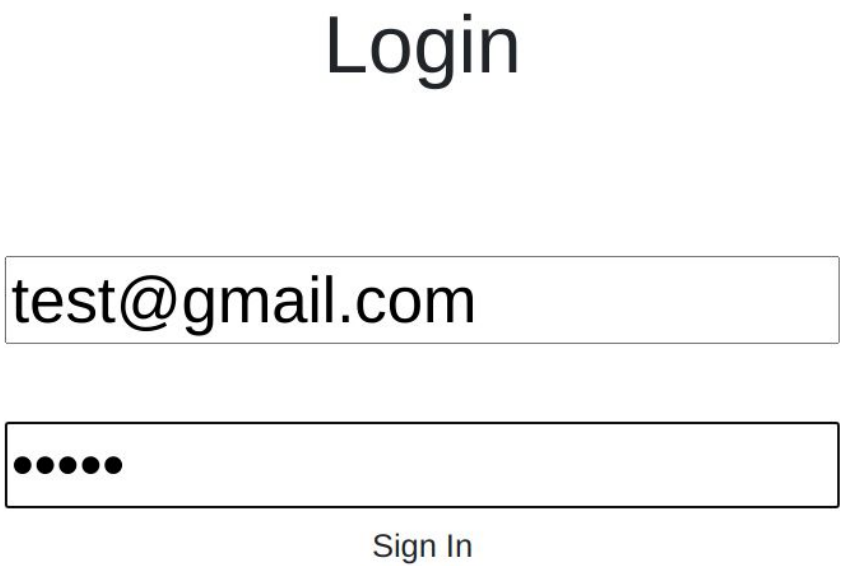
  There are useful bots used by search engines. Such bots should be allowed.

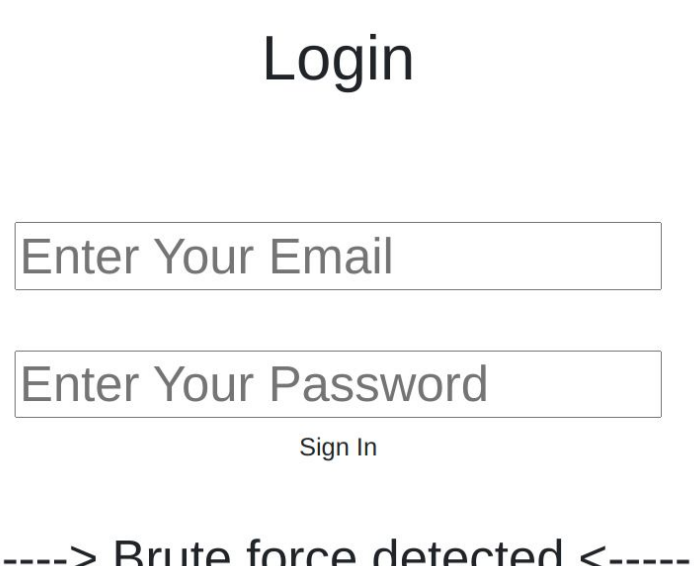
- Use captcha in Page
- Detect the movement of mouse pointer. Bots move mouse pointer in a straight line. Humans don’t move in straight line. On phones, we detect the scrolling speed and finger position.
- Use browser fingerprinting to detect whether it is a VM. VM can be frequently used by bots.
- After detecting as a bot, display a captcha in the page. If failed multiple times, the IP address will be blocked.

Basic implementation

brutefo.herokuapp.com

On multiple attempts with similar password, we detect brute force

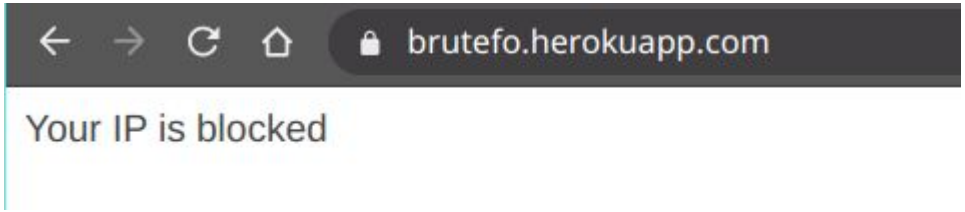




Implementation (contd.)

IP is blocked after

Brute force detection



Bot detection

```
18BCE7147 $ curl brutefo.herokuapp.com
This website is not for bots

18BCE7147 $
```

Thank you