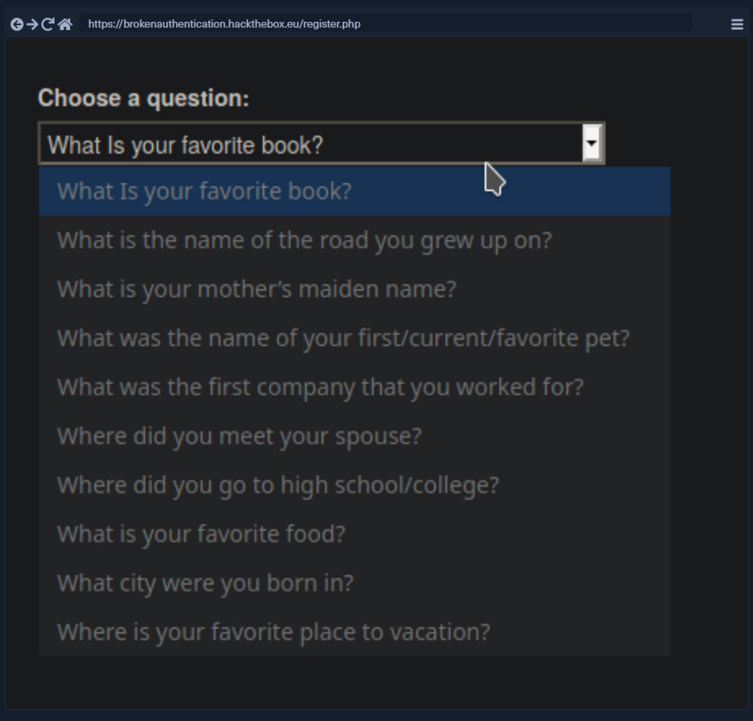
Guessable Answers

Often web applications authenticate users who lost their password by requesting that they answer one or multiple questions. Those questions, usually presented to the user during the registration phase, are mostly hardcoded and cannot be chosen by them. They are, therefore, quite generic.

Assuming we had found such functionality on a target website, we should try abusing it to bypass authentication. In these cases, the problem, or rather the weak point, is not the function per se but the predictability of questions and the users or employees themselves. It is common to find questions like the below.

- "`What is your mother's maiden name?`"
- "`What city were you born in?`"

The first one could be found using `OSINT`, while the answer to the second one could be identified again using `OSINT` or via a brute-force attack. Admittedly, answering both questions could be performed without knowing much about the target user.



We discourage the use of security answers because even when an application allows users to choose their questions, answers could still be predictable due to users' negligence. To raise the security level, a web application should keep repeating the first question until the user answers correctly. This way, an attacker who is not lucky enough to know the first answer or come across a question that can be easily brute-forced on the first shot cannot try the second one. When we find a web application that keeps rotating questions, we should collect them to identify the easiest to brute force and then mount the attack.

Scraping a website could be quite complicated because some web applications scramble form data or use JavaScript to populate forms. Some others keep all question details stored on the server-side. Therefore, we should build a brute force script utilizing a helper, like when there is an Anti-CSRF token present. We prepared a basic web page that rotates questions and a Python template that you can use to experiment with this attack. You can download the PHP file here and Python code here. Take the time to understand how the web application functions fully. We suggest trying manually and then writing your own script. Use someone else's script only as a last resort.

Start Instance

∞ / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Target: Click here to spawn the target system!

+1 Reset the htbadmin user's password by guessing one of the questions. What is the flag?

Submit your answer here... Submit Hint

Previous Next

Cheat Sheet
Resources
Go to Questions

Table of Contents

Broken Authentication
What is Authentication
Overview of Authentication Methods
Overview of Attacks Against Authentication

Login Bruteforcing
Default Credentials
Weak Bruteforce Protections
Bruteforcing Usernames
Bruteforcing Passwords
Predictable Reset Token

Password Attacks
Authentication Credentials Handling
Guessable Answers
Username Injection

Session Attacks
Bruteforcing Cookies
Insecure Token Handling

Skill Assessment
Skill Assessment - Broken Authentication

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left