Challenge URL: /dvwa/vulnerabilities/captcha/
Objective: Change the current user's password in a automated manner because of the poor CAPTCHA system.
Tools needed: Burp Suite
Additional notes: This module requires keys for Captcha v2. V3 will not work.

Did you remember to read this section's README?

The Guide

Testing the Form

Let's begin by walking through the workflow as the developers intended.

From the basic form, we'll try to change our password to "test" and complete the Captcha ...

... which then sends us to a form saying we passed the Captcha check. If we click the "Change" button ...

We get a confirmation that the password changed:

Hm, we see three pages, but not any hint of how they pass data to each other, or how to exploit them.

Let's try using an incredibly-useful tool called Burp Suite to examine the requests as they go back and forth. Let's go to the "Proxy" module/tab, then the "Intercept" tab, making sure the button "Intercept is on" is enabled. If it isn't, click the "Intercept is off" button to change it. Let's then try running through the same execution flow, but watching the requests we send.

Note: You may want to temporarily disable interception until after you verify the Captcha. It's not required, but will make life a little easier.

First form request:

Second form request:

Beating the Challenge

After looking at the two Burp requests, we see a very interesting parameter, **step**, that gets POSTed to the server. In the first request, we see an additional Captcha paramater, but in the second, that parameter disappears. It's almost like the second request just assumes you've already successfully passed the Captcha check? Very interesting. Let's see if we can find a way to re-use the second request and check if it skips the Captcha completely.

In Burp, there's a useful module called the "Repeater". Shockingly enough, it allows us to repeat or replay a given HTTP request. Significantly, it also allows us to modify the request before we send it. That means we have control over all parameter values. Let's see why that matters.

In the "Proxy" module we're currently working in, let's navigate to the "HTTP history" tab. Let's then find our second HTTP request (It's a POST request to our challenge URL). Right click on it and click "Send to Repeater". You'll know this works when the "Repeater" tab/module along the top of the Burp window turns bright orange. Let's move to that module now.

Let's try tweaking the **password_new** and **password_conf** parameters. Change them to something else, like so:

```
step=2&password_new=hackedpw&password_conf=hackedpw&Change=Change
```

When you've modified the request, click the "Go" button on the top left of the Burp windows. Let's examine the response by clicking the "Render" tab:

We get the same "Password Changed." message as before! Let's go back to our browser, log out of DVWA ...

... and try logging into the "admin" account with our new password.

It works! We were able to bypass the Captcha completely by juat replaying the second "confirmation" HTTP request. To exploit this in the real world, we could try to steal the "admin" cookie and try to visit the malicious URL ourselves, or we could trick the end user into clicking the malicious link itself. Regardless, for our intents and purposes, the challenge is complete!