

УДК 004.02

Студ. Д.Д. Карнеева, В.В. Бугаенко
Науч. рук. доц. Д.М. Романенко, асс. С.А. Осоко
(кафедра информатики и веб-дизайна)

ИСПОЛЬЗОВАНИЕ СРЕДСТВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРОЦЕССА РЕГИСТРАЦИИ И АВТОРИЗАЦИИ НА ПЛАТФОРМЕ РНР

С древних времён перед людьми стояла довольно сложная задача – убедиться в достоверности важных сообщений. Придумывались речевые пароли, сложные печати. Появление методов аутентификации с применением механических устройств сильно упрощало задачу, например, обычный замок и ключ были придуманы очень давно.

В настоящее время в связи с обширным развитием сетевых технологий автоматическая аутентификация используется повсеместно.

На процессах аутентификации и авторизации основано разделения прав доступа, без которого не обходится ни одно более или менее серьезное приложение. Поэтому понимать, как они происходили раньше и происходят теперь, очень важно.

Идентификация – процесс определения, что за человек перед нами. Аутентификация – процесс подтверждения, что этот человек именно тот, за кого себя выдает. Авторизация – процесс принятия решения о том, что именно этой аутентифицированной персоне разрешается делать.

То есть, это три разных, последовательных и взаимно не заменяемых понятия. Идентификацию часто подразумевают в составе аутентификации. Важно помнить, что аутентификация и авторизация – это разные вещи.

В ходе аутентификации мы удостоверяемся, что человек, который к нам пришел, обладает доказательствами, подтверждающими личность. Одним из методов обеспечения безопасности является использование так называемых капч.

САРТСНА это аббревиатура от английских слов «Completely Automatic Public Turing Test to Tell Computers and Humans Apart», что означает – полностью автоматический тест Тьюринга для различения компьютеров и людей. Название содержит сразу и функции и основную цель: автоматический тест, который выявляет настоящих пользователей и устраняет злых роботов, которые распространяют спам и удаляют нужные материалы.

Какие основные задачи решает хорошая капча:

- капча – это задача, которую пользователь решает в любых условиях, но компьютер не может с ней справиться;
- капча – это задача с минимальным количеством вводимых данных;
- капча – это задача, легко выполняемая для любого пользователя;
- капча – это задача, которая решается человеком в миг, но она тяжелая для компьютера.

На сегодняшний день существует множество различных капч. Рассмотрим несколько самых популярных:

- **yaCAPTCHA**. Наиболее приспособлена для установки на сайтах и форумах. Не рекомендуется использовать его для блогов, поскольку вводить капчу достаточно сложно. Из-за этого многие посетители могут просто полениться ее распознать и оставлять комментарий.

- **WP-NOTCAPTCHA**. Это довольно таки забавная и простая капча для человека, но трудная для робота. Вместо того, чтобы разглядывать кашу из букв и цифр, разработчики NotCaptcha предлагают выставить картинки в вертикальное положение, передвигая ползунки.

- **reCAPTCHA**. Существует две версии капчи. Одна базируется на классическом вводе символов, а вторая дает задание с требованием выбрать соответствующие запросу картинки. Довольно удобная капча, но все же больше подходит для сайтов с регистрацией аккаунта. Также существует аналог DCaptcha - он выполняет такую же функцию как и reCAPTCHA.

- **Simple CAPTCHA**. Капча выводит различные символы, их нельзя прослушать, но если они не видны, то нажав на соседнюю кнопку символы на картинке можно заменить. При этом, не обновляя страницу, то есть, не теряя написанного комментария в поле.

- **Anti Spam Image**. Напоминает функционал DCaptcha, но является более сложной для роботов. Поскольку в описании картинки указано, что вводить нужно только определенные символы. Благодаря этому является надежной защитой против систем, обученных распознаванию классической капчи.

- **ImHuman**. Очень удобная капча для человека, но сложная для роботов. Здесь выводятся несколько картинок, и надо выбрать из них одну, которая написана в примечании. Похожей капчей является Checkbot. Это вряд ли подойдет более серьезным компаниям. Конечно, их пройти легче, чем неприятный закодированный текст классиче-

ской капчи. Но пользователям все также придется расшифровывать картинку или работать с изображениями.

• **Math Comment Spam Protection.** Капча представляет собой форму вопрос-ответ. Необходимо ввести сумму чисел, которые обозначены в вопросе. Как известно, автоматические боты по рассылке спама не обучены таким действиям. Опять же если робот сможет распознать цифры на картинке, то сложить их и вписать в поле их сумму, для робота уже проблематично.

• **СМС-капча.** Подтверждение по СМС с помощью привлечения личных девайсов, привязанных только к аккаунту, решает вопрос с хакерами. Это также сигнализирует о возможности для большей персональности в процессе обеспечения безопасности с сообщениями, которые могут обращаться к некоторым пользователям.

• **Игровая капча.** Как показано на видео, игровая капча использует сходный с капчей ImHuman принцип проверки. Их игры могут быть действительно приятной функцией для некоторых сайтов. В любом случае, это более интересный способ, чтобы доказать свою человечность (робот бы точно перепутал такси с едой).

Существует два способа установки Captcha:

- Первый предполагает введение в строке поиска названия плагина и его активацию.
- Второй заключается в непосредственном скачивании плагина на компьютер и последующей установке его в блоге.

Рассмотрим реализацию простой капчи с помощью PHP.

Для начала перечислим, что мы можем использовать:

1. В изображении должно использоваться хотя бы несколько цветов. Желательно всегда разные.
2. Должен присутствовать шум.
3. Буквы должны находиться на небольшом расстоянии друг от друга.
4. Размеры символов должны быть разными.
5. Отвратительный шрифт.
6. Символы под случайным углом.

Итак, что же мы сделаем:

Генерация шума;

Генерация текста;

Форма с возможностью обновления капчи;

Обработчик введенных данных.

```

// Пишем функцию генерации изображения
function img_code($code) // $code - код нашей кнопки, который мы укажем при вызове функции
{
    // Отправляем браузеру Header'ы
    header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");
    header("Last-Modified: " . gmdate("D, d M Y H:i:s", 10000) . " GMT");
    header("Cache-Control: no-store, no-cache, must-revalidate");
    header("Cache-Control: post-check=0, pre-check=0", false);
    header("Pragma: no-cache");
    header("Content-Type:image/png");
    // Количество линий. (За текстом и на текст). Поставим случайное значение, от 3 до 7.
    $linenum = rand(3, 7);
    // Задает фон для кнопки. Можно нарисовать свой и загрузить его в папку /img.
    $img_arr = array(
        "i.png"
    );
    // Шрифты для кнопки
    $font_arr = array();
    $font_arr[0]["fname"] = "DroidSans.ttf"; // Имя шрифта
    $font_arr[0]["size"] = rand(20, 30); // Размер в pt
    // Генерируем "подстилку" для кнопки со случайным фоном
    $n = rand(0, sizeof($font_arr)-1);
    $img_fn = $img_arr[rand(0, sizeof($img_arr)-1)];
    $im = imagecreatefrompng($img_dir . $img_fn);
    // Рисуем линии на подстилке
    for ($i=0; $i<$linenum; $i++)
    {
        $color = imagecolorallocate($im, rand(0, 150), rand(0, 100), rand(0, 150)); // Случайный цвет с изображения
        imageline($im, rand(0, 20), rand(1, 50), rand(150, 180), rand(1, 50), $color);
    }
    $color = imagecolorallocate($im, rand(0, 200), 0, rand(0, 200)); // Опять случайный цвет. Уже для текста.
    // Накладываем текст кнопки
    $x = rand(0, 35);
    for ($i = 0; $i < strlen($code); $i++) {
        $x+=15;
        $letter=substr($code, $i, 1);
        imagettftext ($im, $font_arr[$n]["size"], rand(2, 4), $x, rand(50, 55),
            $color, $img_dir.$font_arr[$n]["fname"], $letter);
    }

    // Опять линии, уже сверху текста
    for ($i=0; $i<$linenum; $i++)
    {
        $color = imagecolorallocate($im, rand(0, 255), rand(0, 200), rand(0, 255));
        imageline($im, rand(0, 20), rand(1, 50), rand(150, 180), rand(1, 50), $color);
    }
    // Возвращаем получившееся изображение
    ImagePNG ($im);
    ImageDestroy ($im);
}
img_code($captcha) // Выводим изображение
?>

```

Рисунок 1 – Листинг функции генерации изображения кнопки

В нынешнем времени сервисы по работе с каптчами все более актуальны и востребованы для использования их в различных интернет ресурсах и сервисах, они стремительно развиваются и наращивают свой функционал. Но даже если обеспечение безопасности улучшается, мир по-прежнему полон дизайнерских и других проблем, которые представляют множество возможностей для улучшения.

ЛИТЕРАТУРА

1. https://lifeha.ru/na-klikah-i-prosmotrah/kakie-sushhestvuyushhie-vidy-kapchi-2.html#Sposoby_ustanovki_kapchi
2. https://ru.wikipedia.org/wiki/Аутентификация#Способы_аутентификации
3. <http://deadsign.ru/ux/captcha/>
4. <http://cccp-blog.com/koding/kak-sdelat-kapchu-na-php#kak-rabotaet-kapcha>