**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационной безопасности**

отчет

**по производственной практике в компании “Hacktory”**

Тема: Безопасность фреймворка Yii

|  |  |  |
| --- | --- | --- |
| Студент гр. 7361 |  | Егоров М. |
| Руководитель |  | Юшкевич И. А. |

Санкт-Петербург

2020

**ЗАДАНИЕ**

**на производственную практику**

|  |  |  |
| --- | --- | --- |
| Студент Егоров М. | | |
| Группа 7361 | | |
| Тема практики: Безопасность фреймворка Yii | | |
| Задание на практику:  Разобраться в теме и подготовить следующие материалы:  Общее описание безопасности фреймворка - основные ошибки и уязвимости, которые в нем встречаются и как избежать данных уязвимостей, и реализовать web-приложение с уязвимостью, используя данный фреймворк. | | |
| Сроки прохождения практики: 05.04.2020 – 18.07.2020 | | |
| Дата сдачи отчета: 17.07.2020 | | |
| Дата защиты отчета: 17.07.2020 | | |
|  | | |
| Студент |  | Егоров М. |
| Руководитель |  | Юшкевич И. А. |

**Аннотация**

Целью практики являлось нахождение уязвимости фрейморка Yii с описанием ее и способами устранения. В ходе работы с фрейморком было написано веб-приложение в среде разработки PHP. Приложение имеет удобный для пользователя интерфейс, разнообразную функциональность и содержит в себе уязвимость.

**Summary**

The purpose of the practice was to find the vulnerability of the Yii frontend framework, followed by its description and remedies. In the course of working with the framework, a web application was written in the PHP development environment. The application has a user-friendly interface, diverse functionality and contains a vulnerability.

СОДЕРЖАНИЕ

[введение 5](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845872)

[1. ОПИСАНИЕ Yii FRAMEWORK 6](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845873)

[2. ОСНОВНЫЕ УЯЗВИМОСТИ Yii 7](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845874)

[2.1. Уязвимости моделей 7](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845875)

[2.2. Уязвимости представлений 7](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845876)

[2.3. На что еще нужно обратить внимание 8](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845877)

[заключение 13](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845880)

[список использованных источников 14](file:///C:\Users\MadMax\Downloads\Egorov_M.docx#_Toc45845881)

# введение

Задачами практики является изучение фреймворка Yii, нахождение его уязвимости с последующим ее описанием и способами устранения. В ходе работы с фрейморком было написано веб-приложение в среде разработки PHP. Приложение имеет удобный для пользователя интерфейс, разнообразную функциональность и содержит в себе уязвимость.

# ОПИСАНИЕ Yii FRAMEWORK

Yii – объектно-ориентированный компонентный фреймворк, реализующий парадигму MVC. Yii имеет хорошую документацию. Она долгое время была англоязычной, но силами русскоязычных пользователей перевод все-таки увидел свет и теперь доступен на сайте проекта. Шаблон проектирования MVC предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: Модель (Model), Представление (View) и Контроллер (Controller) – таким образом, что модификация каждого компонента может осуществляться независимо. По результатам тестов phpmark Yii показал лучшую производительность. Yii имеет качественную подсистему отложенной инициализации (то есть, код загружается только тогда, когда он необходим). Несмотря на все плюсы этого фреймворка, в нем присутствуют и минусы. Часто в новых версиях появляются баги и уязвимости безопасности.

# 2. Основные уязвимости Yii

## 2.1. Уязвимости моделей

Посмотрим на метод update контроллера, генерируемого с помощью gii:

|  |  |
| --- | --- |
|  | |
|  | public function actionUpdate($id)      {          $model = $this->findModel($id);            if ($model->load(Yii::$app->request->post()) && $model->save())  {              return $this->redirect(['view', 'id' => $model->id]);          } else {              return $this->render('update', [                  'model' => $model,              ]);          }      } |

Подразумевается, что у нас есть форма, которая отправляет POST-запрос на данный action. Данные загружаются в модель, валидируются и сохраняются. Допустим, что данная форма нужна чтобы дать возможность пользователю обновить свое отображаемое имя (username). Но при неправильном указании safe-атрибутов в модели, злоумышленник сможет обновить и другие поля, такие как пароль или статус, причем не только для своего id. Для этого достаточно отправить POST запрос на update/чужой-id передав не только значение username, но и значение password.

Виной всему метод $model->load(), который позволяет массово присвоить значения в модель. Это гораздо удобнее, чем присваивать каждое значение по одному. Но не стоит забывать о safe-атрибутах (безопасных атрибутах).   
Изменить данное поведение можно лишь переопределив сценарий по умолчанию в модели:

|  |  |
| --- | --- |
|  | |
|  | public function scenarios()  {      $scenarios['default'] = ['login', 'username', '!password'];      $scenarios['example'] = ['login', 'phone'];      return $scenarios;  } |

## 2.2. Уязвимость представлений

В Yii по умолчанию используется шаблонизатор для VIEW чистый PHP. К сожалению, это может привести к ошибкам экранирования вывода. Разработчик выводимый контент должен экранировать, используя Html::encode(), либо аналог. Обычно, для поиска подобных ошибок, пентестер забивает во все возможные поля форм код вида:

<script>alert(document.cookie);</script>

Потом проходиться по всему проекту и смотрит что упало. Но иногда бывает, что форма выводиться не полностью. Если же атрибуты будут посланы явно отдельным запросом Post, то будут загружены в модель. Либо сейчас заполнить атрибут через форму пользователь не может, и все хорошо, а через месяц форма меняться, параметр становиться доступен для редактирования, и где-то в совсем другой части проекта проявляется XSS.

## 2.3. На что еще нужно обратить внимание

Есть несколько моментов, которые стоит учитывать при разработке на фреймворке Yii

1. **В версии 1.1.14 фреймворка Yii была обнаружена серьезная уязвимость**, позволяющая при определенных условиях выполнять произвольный PHP-код. Разработчики выпустили обновление (версия 1.1.15), полностью исправляющее проблему.

Уязвимость затрагивает `CDetailView`. Если ваше приложение берёт значение `value` от пользователя, то атакующий имеет возможность выполнить произвольный PHP скрипт на вашем сервере.

Пример уязвимого кода:

//Бажный код с отсутствием проверки

//Передаем имя любой php-функции и она успешно выполняется is\_callable($attribute[‘value’])? call\_user\_func($attribute[‘value’],$this->data)

2. Для предотвращения межсайтового скриптинга (XSS), необходимо использовать полезный компонент CHtmlPurifier, который может отфильтровать весь вредоносный код при помощи тщательно проверенного белого листа. Также компонент делает код совместным со стандартами.

3. Для того, чтобы не допустить CSRF, важно придерживаться простого правила: GET – только для получения данных. Ничего менять при GET – запросах нельзя. Для POST необходимо использовать случайное значение, которое можно проверить на сервере и убедиться, что запрос идѐт оттуда, откуда нужно. В Yii реализована защита от CSRF – атаки, проводимой через POST. По умолчанию, защита от CSRF отключена. Для еѐ включения необходимо настроить компонент CHttpRequest в файле конфигурации.

# заключение

Основными целями практики было изучение фреймворка Yii, нахождение его уязвимости с последующим ее описанием и способами устранения, также было реализовано web-приложение с уязвимостью, используя данный фреймворк.

# список использованных источников

1. Прогрессивный JavaScript-фреймворк // Vue.js. URL: <https://ru.vuejs.org/> (дата обращения: 06.07.2020).

2. Статья о XSS // URL: <https://dev.to/nandod1707/should-you-care-about-xss-in-vuejs-1ea6> (дата обращения: 08.07.2020).

3. Статья о XSS в Vue.js //URL: <https://blog.sqreen.com/xss-in-vue-js/> (дата обращения: 09.07.2020).