

LEVEL UP! WEEK

5 дней 10 спикеров





[СТАТЬИ](#) · 6 августа 2019, 13:00 37683  Svyat Login, QA gangsta Lead в EVO

Как провести тестирование на безопасность: руководство для Manual QA

Эта статья нацелена на подрастающее поколение QA и разработчиков, которым интересно узнать что-то об уязвимостях: с чего начать, какими инструментами можно пользоваться начинающему в этом деле (практические советы). В материале будет изложено то, что я хотел бы прочесть в начале своей карьеры Security QA.

Вступление

Когда я был Manual QA, мне всегда казалось, что искать уязвимости очень трудно, что этим могут заниматься только те люди, которые умеют программировать. Поэтому я выбрал сначала путь автоматизатора, так как зачастую QA развиваются именно в этом направлении. Но после более чем полутора лет в должности автомейшена мне стало скучно... Да-да, стало скучно, так как мне неинтересно было все время писать код и не общаться с командой девелоперов, продуктами и другими членами команды, как я делал это, когда был мануальщиком.

Недолго думая, в каком же направлении мне развиваться... Точнее, на это повлияло несколько атак ловцов уязвимостей на наш проект. После атак они предоставили нам репорты, и мне стало обидно: мы же могли предусмотреть эти кейсы своими силами, вроде же было не очень тяжело найти их. Так вот, после ознакомления с репортами, предоставленными этими ловцами, и прослушивания докладов по безопасности я решил двигаться в этом направлении.

Немного о себе: в тестировании я уже больше 7 лет, занимаюсь поиском уязвимостей больше 4 лет. Веду тренинги по [тестированию безопасности](#). Помимо основной работы провожу аудиты по QA. Веду [блог](#).

Начнем с того, с какими трудностями я сталкивался, когда начал изучать тестирование на безопасность. Наверное, с такими же столкнетесь и вы:

1. Проблема с тем, где без регистрации и СМС найти и бесплатно почитать нужную литературу.

Чтобы вот просто было написано понятным языком без заумных фраз, как любят описывать, чтобы все было структурировано и были примеры, где можно было бы попрактиковаться с прочитанным. При гуглении я получал огромный вброс инфы без разбору, что же надо для начала.

2. Проблема с нехваткой времени.

Согласитесь, трудно найти время на то, чтобы заставить себя учить что-то новое без надобности срочно применять эти навыки :-)

А вот и решения:

- С первой проблемой мне помогли... даже не поверите кто – те же самые ловцы уязвимостей, которые атаковали нас и писали нам репорты о наших уязвимостях. Читая эти репорты, я параллельно гуглил и искал в статьях информацию о причинах возникновения и критичности данных уязвимостей. В итоге я составил для себя чек-лист массовости (насколько распространен этот вид уязвимости среди продуктов) и критичности (насколько опасно иметь такую уязвимость в зависимости от того, сколько бед, которые могут привести к потере доверия к бренду, можно натворить с ней на проекте).
- Также решением первой и второй проблем стали мои выступления перед публикой на тему уязвимостей. За время подготовки к тому, чтобы учить кого-то чему-то, вы изучите материал



19 февраля – 12 марта
Почему нету везде подлинных делать продукты
Бесплатный курс по бизнес-анализу



16 февраля, Online
JStify Lightning Talks



16 февраля, Одесса
Курсы Salesforce Developer



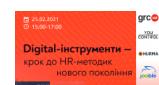
20 февраля, Online
Intellias DevOps Hiring Day



21 февраля, Online
Бесплатный урок по курсу «Английский для IT» в EnglishDom



24 февраля, Киев
или online
Курс FrontEnd Developer в CyberBionic Systematics



25 февраля, Online
Digital-Instrumenti – крок до HR-методик нового покоління



25 февраля, Online
Web crowd 8.0: Python meetup



13 марта – 11 апреля, Online
Тренінг “Project Management: Deep Dive”

25 марта – 10 июня, Online

настолько хорошо и настолько быстро, что сами удивитесь :-) У вас сразу появляется:

- дедлайн, к которому вы должны успеть подготовиться и хорошо изучить тему;
- стимул учить новое, так как перед вами стоит задача выступить перед публикой, а опозориться не сильно хочется, чтобы не показать, что вы чего-то не знаете в этой области, но что-то там вещаете;
- время. Скажете откуда? Да вы просто под психологическим воздействием, что нужно подготовиться к выступлению, меньше бездействуете (меньше времени уделяете просмотру видосов с котиками и остальными приколами). Да и после работы вместо просиживания штанишек перед сериалчиком вы готовитесь к своему дебюту на сцене. Нормальный такой лайфхак? :-)

3. И самое главное: после того как вы ознакомились с какой-либо теорией о уязвимости, немедленно идите применять ее на практике, так как теория в голове без практики имеет способность пропадать, не подавая виду, быстро и бесследно. И это будет реально напрасно потраченное время. Попрактиковаться можно у себя на проекте с разрешения руководства – а я думаю, оно по-любому разрешит вам, так как хорошо, когда есть человек, который может проверить хоть на какую-то безопасность. Либо можно прогулить Broken Web Application и по первой ссылке скачать себе приложение, запустить его на VirtualBox и применять там свои наработанные навыки.

The screenshot shows the OWASP BWA application catalog. At the top, there's a warning message: "!!! This VM has many serious security issues. We strongly recommend that you run it only on the 'host only' or 'NAT' network in the virtual machine settings !!!". Below this, the catalog is divided into three sections:

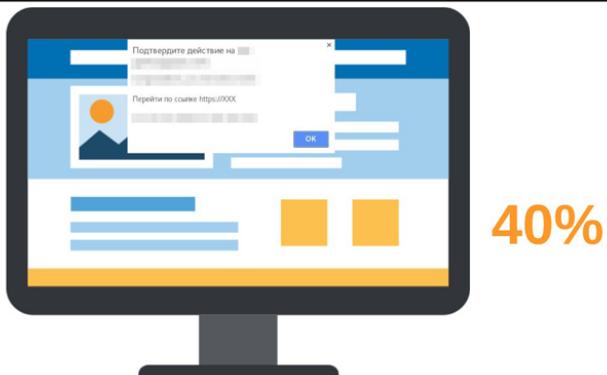
- TRAINING APPLICATIONS**: Includes OWASP WebGoat, OWASP F-SAP Java SwingGet Interactive, OWASP RailoGoat, OWASP Security Shepherd, OWASP Code Injection Rainbow, and Dynamic Vulnerable Web Application.
- REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS**: Includes OWASP Victim, Google Gavotte, StackOverflow, cyclone, OWASP I-Liner, Hacker, Bodil, and PenTestia.
- OLD (VULNERABLE) VERSIONS OF REAL APPLICATIONS**: Includes WordPress, Getbootstrap, Yod, Gallery2, Joomla!, OrangeHRM, GIGI-PHP, webCalendar, rtki-test, and overStats.

В этом приложении много разных лабораторий с самыми распространенными уязвимостями, так что вам должно хватить этого, чтобы хорошенько набраться опыта пентратации.

А теперь давайте поговорим о самых распространенных уязвимостях, которые могут встретиться вам на пути, и о том, как их искать...

XSS

A7:Cross-Site Scripting (XSS)



Это один из типов уязвимости web application, который дает скрипту возможность отрабатывать на страницу, написанную на JS. Такая уязвимость дает возможность злоумышленнику внедрить свой сценарий в работу вашего приложения. [По статистике](#), 40% компаний, прошедших через



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты

сканеры, имеют эту уязвимость. В рейтинге OWASP Top 10 она на 7-м месте. Причина появления этой уязвимости — это доверие со стороны разработчика, что пользователь не будет вносить разнообразные куски кода на сайт.

Что же может сделать в своих целях ловец уязвимостей?

- Изменить настройки (например, заменить бэкграунд сайта, всунуть, допустим, на задний фон скрин Joycasino, размещать рекламу через всплывающие попапы).
- Стырить куки пользователя.
- Фишинг (хакер может вставить поддельную форму для входа на страницу, которую вы посещаете, используя DOM, установив в атрибуты `action` этой формы отправку данных на свои собственные серверы).
- Кейлогер (хакер может внедрить что-то типа отслеживания действий, выполняемых на клавиатуре пользователем, используя `addEventListener`, а потом отправить все эти нажатия клавиш на свой серверок, записав конфиденциальную информацию пользователя. Например, это могут быть пароли и номера кредитных карт. Стремно, согласитесь.)

Бывает три типа XSS:

- Непостоянные (отраженные) XSS.
- Постоянныe (хранимые) XSS.
- XSS DOM-модели.

Рассмотрим каждую из них.

Непостоянныe (отраженные) XSS

Этот класс XSS является самым распространенным на сайтах. Давайте смоделируем вектор атаки с помощью этого класса. Рассмотрим это на примере специально подготовленной лаборатории по уязвимостям DVWA. У нас есть форма с инпутом, при этом инпут, как мы видим по исходному коду этой формы, не фильтруется никаким фильтратором. То есть переменная `$_GET['name']` принимает без перебора все, что в нее впишут. Соответственно, вписанный в этот инпут скрипт `<script>alert(1);</script>` вызовет попап на этой странице со значением в нем цифры 1, после того как мы отправим на сервер запрос с этим скриптом. В итоге то, что мы отправили через этот инпут на сервер, передается GET-параметром в урле, так как эта форма общается с сервером по GET.

```
<?php
if(!array_key_exists("name", $_GET) || $_GET['name'] == NULL || $_GET['name']==""){
    $isempty=true;
}
else{
    echo '<pre>';
    echo 'Hello' . $_GET['name'];
    echo '</pre>';
}
?>
```

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? Submit

More... http://www.owasp.org/index.php/Cross-site_scripting
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.csseasort.com/xss-test.html>

На следующей картинке мы можем увидеть результат:

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не в усих получится делать продукты](#)

The screenshot shows the DVWA application interface. On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (highlighted in green), and XSS stored. The main content area has a title 'Vulnerability: Reflected Cross Site Scripting'. It contains a form with a text input field labeled 'What's your name?' and a 'Submit' button. Below the form, the text 'Hello' is displayed. A large blue modal window is open, showing the value '1' and an 'OK' button. The URL in the browser's address bar is highlighted with a red box, showing the injected JavaScript code: '<script>alert(1)%3B<%2Fscript>'.

Как мы видим, в урле наш JS-код, вызывающий тот самый попап, в который мы можем вписать любую информацию на текущей странице, где есть такая уязвимость. Каковы методы применения такой уязвимости, спросите вы? Приведу их ниже...



Допустим, SEO-менеджер изучил новый способ распространения рекламы – через попапы этой уязвимости. Он находит сайт, где эта уязвимость срабатывает, в нашем случае это bank.ua. Он дописывает в аргумент поиска после названия скрипта ')%3Balert('XSS')%3Bvar b=' , при этом браузером будет экранироваться попап, отображающий то, что в него напишет наш менеджер, так как данные не подвергаются обработке. Затем паренек просто распространяет весь этот урл через разные социальные сети. Если пользователь, который получит этот скрипт, внимательный, он просто может удалить добавленный в урл скрипт, и попап не выпрыгнет при открытии страницы банка, но если не заметил, то он, помимо открытия страницы сайта, получит вывод попапа, в котором будет информация, которую написал туда наш менеджер.

Хранимые (постоянныи) XSS

Этот вид XSS уже считается более разрушительным типом атаки, чем предыдущий. Он возможен, когда ловцу уязвимостей удается закинуть на сервер скрипт JS, который будет выполняться в браузере каждый раз при заходе на запрашиваемую вами страницу. Самым простым примером этой уязвимости являются форумы, на которых разрешено оставлять комментарии в HTML-формате без ограничений. Другими словами, хранимый XSS возникает, когда разработчики осуществляют некорректную фильтрацию при сохранении входных данных в БД на сервере, а затем выводят эти же данные в браузер пользователя.

Примерчик на том же ресурсе DVWA:

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

```

<?php

if(isset($_POST['btnSign'])) {
    $message = trim($_POST['mtxMessage']);
    $name = trim($_POST['txtName']);

    // Sanitize message input
    $message = stripslashes($message);
    $message = mysql_real_escape_string($message);

    // Sanitize name input
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES
    ('$message','$name')";

    $result = mysql_query($query) or die('<pre>' . mysql_error() . '</pre>');

}

?>

```

В этот раз мы внедряем JS-скрипт (`<script>prompt(111);</script>`) в поле Message, которое, как мы видим в коде, тоже не фильтруется нормальным образом от пользовательского ввода. После отправки этого скрипта на сервер он положится в базу на сервер и каждый раз будет вызываться при посещении этой страницы. Ах да... в этом случае будет не просто попап с выводом текста «111», а будет задействован еще и инпут, в которой можно что-то вписывать. А если хорошенко подумать над реализацией этого скрипта, то можно так подхачить его, что как будто выпрыгивает форма авторизации, в которую пользователь введет свои данные, а они будут отправлены на сервер злоумышленнику. Как вам такой исход этой уязвимости?

И этот попап будет воспроизводиться до тех пор, пока кто-то не почистит значение в базе на сервере, куда отправили храниться этот скрипт, либо не пофиксит саму экранизацию XSS на веб-странице.

DOM-модели XSS

А вот этот вид XSS самый опасный из всех. XSS в DOM-модели появляются на стороне клиента во время обработки данных внутри самого JavaScript. Этот тип XSS получил такое название, потому что нам нужна Document Object Model, чтобы сделать его. Как вы поняли, DOM – это сокращение. Через него можно получать доступ к содержимому HTML- и XML-документов, даже изменять содержимое: либо структуру документа, либо его оформление.

Что можно сделать с помощью этой уязвимости:

- Изменить страницу сайта (можно добавлять/заменять картинки, добавлять куски нового функционала и так далее).
- Можно воровать сессии, куки пользователя (своровав эти данные, злоумышленник может воспользоваться ими в своих целях: например, можно авторизоваться под этими данными и быть как будто тем пользователем).
- Кейлогер (заузав скрипт на странице с уязвимостью, злоумышленник будет получать все данные, которые вводит пользователь на клавиатуре, находясь на зараженной странице).
- Мало того что можно сделать то, что описано выше, так еще можно и залезть в операционную систему с помощью уязвимости ms10_002_augora, которая доступна для старых браузеров Safari и Internet Explorer 7. Вы скажете: «Да это же древние браузеры, кто ими будет пользоваться?» Но нет, давайте вспомним банковские конторы, которые

Интересные статьи



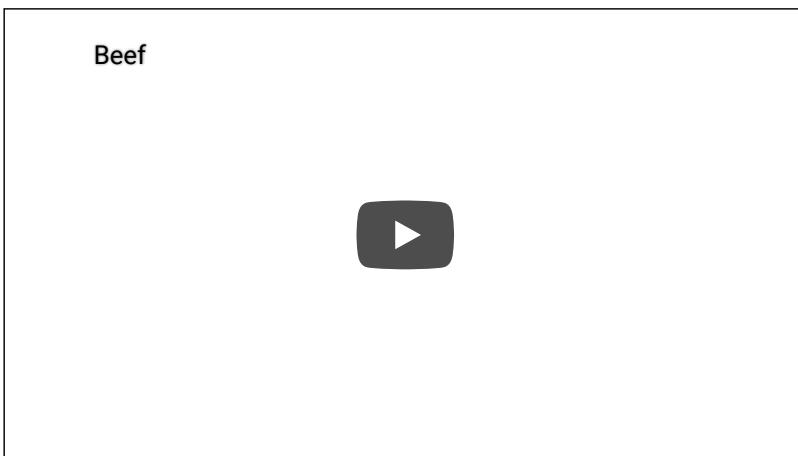
[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

заключили десятилетние контракты на использование этой версии браузера, – вот тут такие пользователи и попадутся на эту уязвимость.

Приведу пример, как можно своровать куки пользователя, а потом применить их у себя в браузере и стать этим пользователем, используя при этом инструмент BeEF:



Интересные статьи

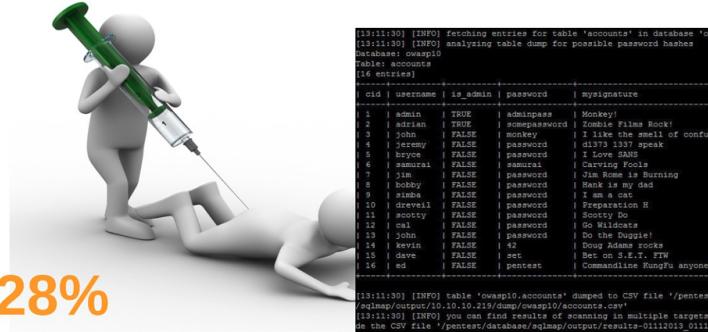


DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты

Injection



28%

А теперь перейдем к первой уязвимости из рейтинга OWASP Top 10 – к Injection. [По статистике](#), ею заражено 28% компаний... Я рассматриваю ее позднее, так как эта уязвимость с каждым годом становится все менее распространенной, но остается наиболее критичной из всех, потому что с ее помощью можно увести всю вашу базу данных. Эта уязвимость делится на такие векторы для атаки:

- инъекции через запросы SQL, LDAP, XPath;
- инъекции через команды OS command;
- инъекции через синтаксический анализ XML.

С помощью этих векторов атакующий может получить доступ как к одной учетной записи, так и ко всей базе данных клиентов этого ресурса. Для применения атаки используются всего лишь специальные символы и дополнительные операторы в зависимости от типа базы SQL.

Давайте рассмотрим один из векторов Injection – с использованием запроса для страницы авторизации. Если у вас в приложении такая есть, проверьте ее, чтобы с ней не было таких проблем, которые я вам покажу. При входе в систему через эту страницу мы будем искать запись пользователя в нашей базе данных SQL на основе введенного имени и пароля пользователя. Если мы получим какой-то результат, то он авторизует этого человека. Все довольно просто, на самом деле существуют такие страницы входа, которые работают таким же образом, имея такую же уязвимость.

Authorization	
User	<input type="text"/>
Pass	<input type="password"/>

Select *
From users
Where
user_name = '\$username'
AND password = '\$password'

Итак, тут мы видим форму, при заполнении которой введенные данные будут отправлены в SQL-запрос, и он вытянет нам ответ из базы данных. Если такой юзер есть в базе данных, то нас авторизует. Если нет, то нам скажет: сорян, зарегайся. То, что мы будем вводить в поле User , передается в переменную \$username , а то, что введут в поле Pass , передается в переменную \$password нашего SQL-запроса. А теперь рассмотрим то, что будет происходить при вводе значений в эти поля.

Authorization	
User	<input type="text" value="Svyat"/>
Pass	<input type="password" value="1234"/>

	id	username	password	email
1	Svyat	1234	svyat@com.ua	

Select *
From users
Where
user_name = 'Svyat'
AND password = '1234'

Давайте введем такие данные:

- в поле User – значение Svyat, которое, как мы видим (отмечено красной стрелкой), станет на место переменной \$username ;
- в поле Pass – значение 1234, которое, как мы видим (отмечено синей стрелкой), станет на место переменной \$password .

После запуска этого SQL-запроса будет выполнен поиск в базе данных, где будет найден этот пользователь со всей его информацией, соответственно, мы получим положительный ответ на странице авторизации и успешно авторизуемся.

Давайте теперь рассмотрим, что может сделать злоумышленник в нашем случае...

Authorization	
User	<input type="text" value="admin"/>
Pass	<input type="password" value="` OR 100=100 --"/>

	id	username	password	email
0	admin	_____	admin@com.ua	

Select *
From users
Where
user_name = 'admin'
AND password = ''
OR 100=100 --'

Тут злоумышленник совершает ход конем и целился в юзера admin. Давайте разберем подробнее, что он ввел:

- в поле User – значение admin , которое, как мы видим, станет на место переменной \$username ;
- в поле Pass – значение ' OR 100=100 -- , которое, как мы видим, станет на место переменной \$password .

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

А теперь давайте разберем, что происходит с переменной `$password`, когда в нее попадают данные такого рода, и как сервер обработает их.

- Символом кавычки ' из заданного значения ' OR 100=100 – мы закрываем дефолтную кавычку данной переменной, оставив пароль незаполненным. Это видно на картинке выше.
- Затем переходим к слову OR из нашего значения ' OR 100=100 -- – оно говорит о том, что мы вызываем оператор OR, который будет выполнять свою функцию в SQL-запросе, так как предыдущей кавычкой мы закрыли нашу переменную.
- А выражение 100=100 из нашего значения ' OR 100=100 -- говорит серверу: даже если пароль не подошел, все равно прими за true, как будто он подошел для тебя, так как 100=100 является true.
- И напоследок символы -- из нашего значения ' OR 100=100 -- закомментируют дефолтную кавычку. То есть весь код, который находится после символов -- , становится нерабочим. У каждой базы данных вызов операции, чтобы закомментировать код, происходит по-разному: бывает #, бывает –, бывает */.

Исходя из изложенного выше, SQL-запрос получил дополнительное условие, которое диктует серверу новые правила игры, заставляющие его авторизовать нас как пользователя системы, несмотря на то что мы не ввели правильный пароль. А так как это еще и пользователь-админ, мы имеем все привилегии управления приложением. В запрос добавлено значение OR 100=100, которое всегда будет оцениваться как true (так как 100 всегда равно 100, чтобы получить true, необязательно писать именно 100=100, можно взять любое значение, которое равно такому же значению). Теперь этот пользователь будет авторизован как администратор, даже если он не знает пароля. Это не прикольно, согласитесь. Но если есть такая уязвимость, то это может дойти и до такого случая, как на картинке ниже...



Authorization

User	admin
Pass	'; DROP TABLE users --

Если в поле Pass ввести значение '; DROP TABLE users -- , а затем отправить на сервер, то можно вообще лишиться данных юзера в нашей базе, но опять же это если мы не фильтруем в нашем приложении то, что вводит пользователь.

Если подытожить, эта уязвимость нацелена на применение дополнительных команд и операторов в запросах на сервер. И если в ней, как в XSS, нет фильтрации на это дело, то такая дыра будет присутствовать у вас в приложении.

Инструменты

Эту и другие уязвимости с OWASP Top 10 можно искать с помощью автоматизированного инструмента. С ним может справиться и Manual QA, достаточно знать, какие есть самые критические уязвимости, и уметь пользоваться инструментами для их поиска.

К инструментарию Penetration Tester можно отнести такие:

Бесплатные:

- OWASP ZAP;
- Nmap;
- Metasploit;
- SQLmap;
- Wireshark;
- Ettercap;

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

- BeEF.

Платные:

- Burp Suite;
- Acunetix;
- Charles;
- Veracode.

Так как по каждому из них можно рассказывать много чего, я уделил внимание лишь одному. Это будет бесплатный инструмент OWASP ZAP, чтобы вы могли поюзать его и понять всю суть поиска уязвимостей.

Что такое OWASP ZAP

Начнем с того, что OWASP ZAP – это в первую очередь инструмент, который достаточно прост в использовании для тестирования на проникновение в ваше приложение, а также для поиска уязвимостей в веб-приложениях. Программа предназначена как для пользователей, имеющих опыт работы в сфере информационной безопасности, таких как разработчики и мануальные QA, так и для начинающих.

Режимы OWASP ZAP

В ZAP есть несколько режимов работы, при которых он будет проводить сканирование:

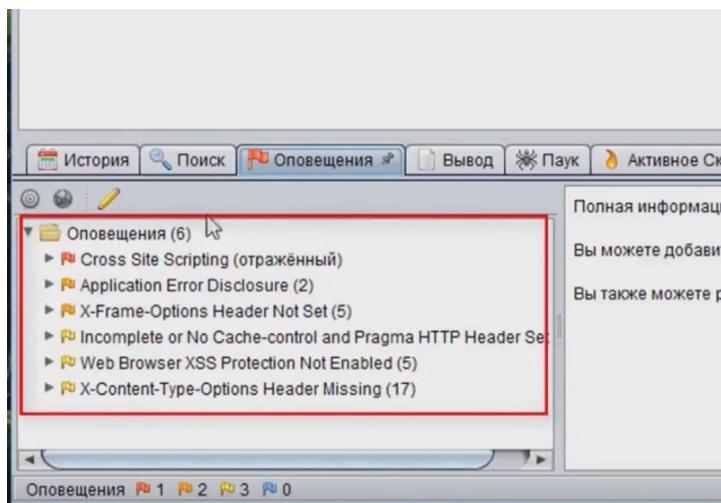
1) Безопасный режим – в этом режиме нельзя совершить что-либо потенциально опасное для приложения.

2) Защищенный режим – в этом режиме пользователь (бот) может выполнять только вредоносные действия по URL-адресам, указанным в области браузера.

3) Стандартный режим – в этом режиме пользователь (бот) может делать все, что имеет значение для приложения.

4) Режим Attack – при нахождении новых узлов в области действия шпиона они активно сканируются, как только обнаруживаются.

Оповещение



После того как вы запустите сканер, все найденные ошибки OWASP ZAP будут отсортированы по степени серьезности уязвимостей и будут находиться на вкладке «Оповещение». Красным флагом будут отмечены самые серьезные, такие как XSS, SQL Injection и так далее, оранжевым – менее серьезные уязвимости типа CSRF и остальные малозначимые.

Чтобы более подробно посмотреть найденные сканером уязвимости, достаточно несколько раз кликнуть по какой-то из них.

При этом откроется попап, в котором будет расписано:

- что это за уязвимость и на что она влияет;
- ее критичность;
- скрипт, с которым она воспроизводится;

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

- литература о том, как можно починить ее.

Изменить оповещение

Cross Site Scripting (отражённый)

URL-адрес: https://xss-game.appspot.com/level1/frame?query=%3C%2Fb%3E%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E%3C%3E

Риск: High

Confidence: Medium

Параметр: query

Атака: <script>alert(1);</script>

Evidence: <script>alert(1);</script>

CWE ID: 79

WASC ID: 8

Описание:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

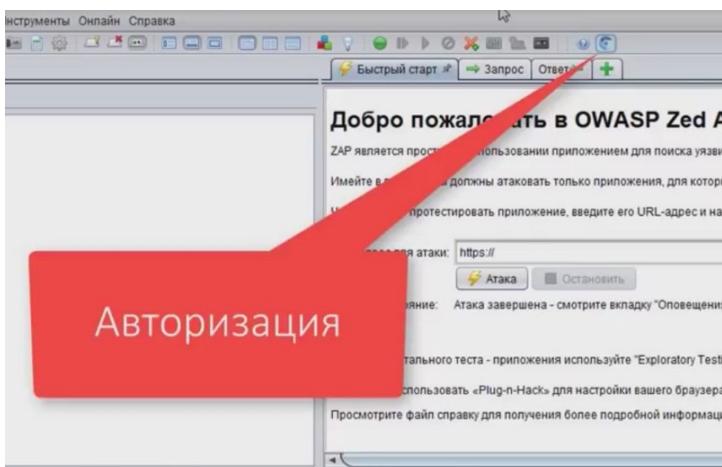
Дополнительно:

Решение:

Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI

Ссылка:

Авторизация



Если в вашем проекте присутствует авторизация, то те ссылки, которые доступны для авторизованного пользователя, не будут доступны для OWASP ZAP без выполненной настройки юзера (бота), под которым наш сканер сможет зайти для выполнения поиска уязвимостей на тех страницах, которые доступны для авторизованного юзера. Для того чтобы быстро настроить этого юзера, кликаем по иконке встроенного браузера. При этом откроется сам браузер, в котором введены настройки прокси для обмена данными (request и response). То есть OWASP ZAP становится посредником, запросы, которые вы посыпаете в браузере на сервер, сначала идут на OWASP ZAP, а с него – на сервер.

Так же происходит и в обратном порядке: ответ от сервера будет тоже сначала идти на OWASP ZAP, а уже с него – в браузер. Таким образом, вы сможете мониторить весь трафик. Затем выполните авторизацию на сайте, который собираетесь тестировать, через этот браузер.



Username

Password

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



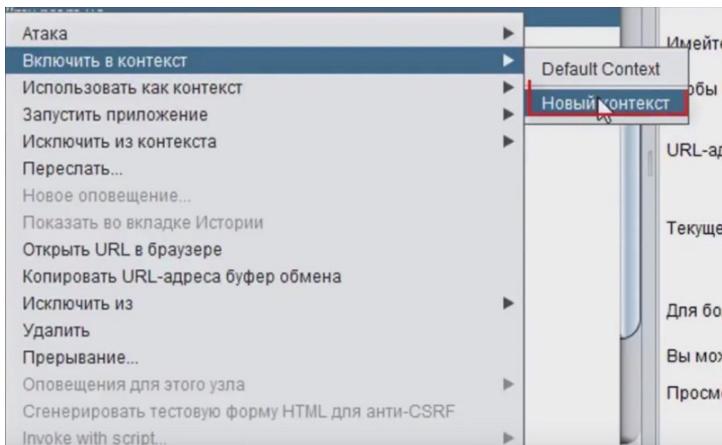
Почему не у всех получится делать продукты

Все данные, которые вы ввели, будут перехвачены нашим приложением OWASP ZAP. Это мы сможем увидеть в приложении ZAP, с левой стороны. Там появится папочка с названием нашего хоста (сайта), который мы прогнали через браузер с прокси.

The screenshot shows the OWASP ZAP 2.7.0 interface. On the left, the 'Сайты' (Sites) section lists contexts. A red arrow points from the text above to the 'http://192.168.17.218' entry, which is highlighted with a red box. To the right, the main panel displays the website's content with the title 'Добро пожаловать' (Welcome). Below it is a status message and a search bar. At the bottom, there's a table of network requests.

ID	Req. Timestamp	Метод	URL-адрес	Код	Причина	RTT	Size Re
1	24.07.19 8:51:06	GET	http://192.168.17.218/dvwa/login.php	200	OK	16 ms	1 224 б
5	24.07.19 8:51:06	GET	http://192.168.17.218/dvwa/dvwa/css/login.css	200	OK	14 ms	608 байт
9	24.07.19 8:51:12	POST	http://192.168.17.218/dvwa/login.php	302	Found	11 ms	0 байт
10	24.07.19 8:51:13	GET	http://192.168.17.218/dvwa/index.php	200	OK	7 ms	4 704 б
11	24.07.19 8:51:13	GET	http://192.168.17.218/dvwa/dvwa/css/main.css	200	OK	12 ms	3 945 б
13	24.07.19 8:51:13	GET	http://192.168.17.218/dvwa/dvwa/js/dvwaPag...	200	OK	24 ms	775 байт

Когда мы расхлопнем эту папку, мы увидим иерархию нашего приложения. Здесь мы и увидим наш запрос POST, с которым авторизовались. И если мы будем видеть, что, помимо нужного нам хоста, загрузились еще и другие ресурсы, которые интегрированы в наше веб-приложение, типа разных соцсетей, чтобы не тратить время на сканирование и их – так как нужно же проверять наш проект, а не другие :) – нам нужно задать «контекст» только для одной папки, которую мы хотим проверить. Для этого выбираем нужную папку и включаем ее в контекст.



После того как мы задали папке контекст, мы должны найти в ней запрос, который выполнялся на авторизацию, а затем создать юзера для этого метода, выбрав пункт «Использовать как контекст», а там – пункт Form-based.

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты

Сессия без названия - OWAS

Файл Правка Вид Анализ Отчет Инструменты Онлайн Справка

Стандартный режим

Сайты Сценарии

Контексты Default Context

Сайты

http://192.168.17.218 dvwa

dvwa

css images js

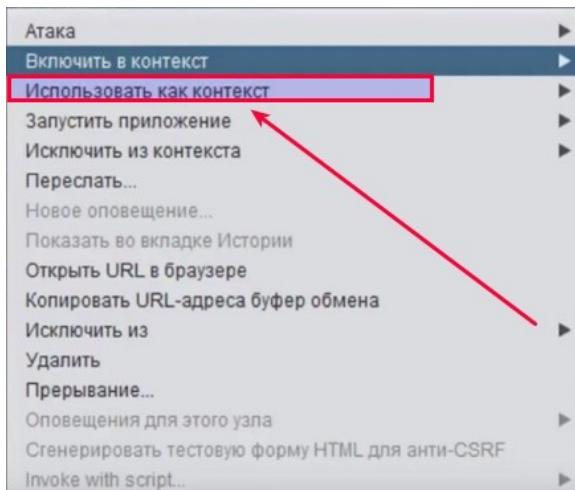
GET:index.php GET:login.php

POST:login.php(Login,password,username)

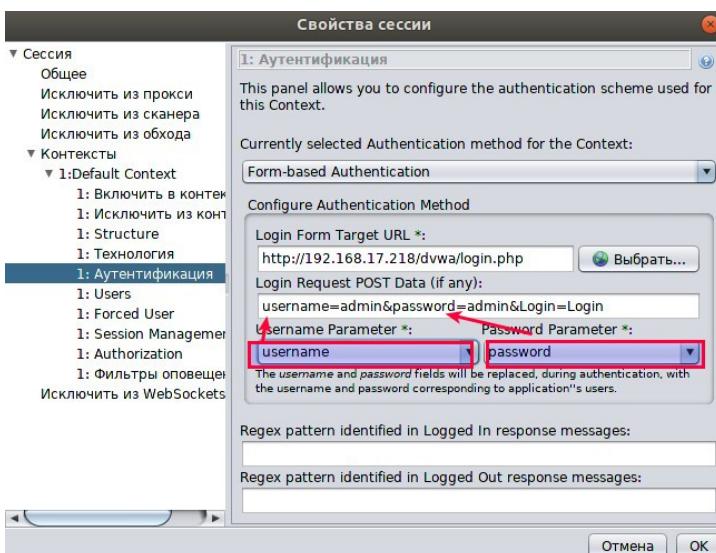
История Поиск Точки остановки Оповещения Активное Сканирование Паук

Фильтр: Выкл Экспорт

ID	Req. Timestamp	Метод	URL-адрес	Код	Причина
1	24.07.19 8:51:06	GET	http://192.168.17.218/dvwa/login.php	200	OK
5	24.07.19 8:51:06	GET	http://192.168.17.218/dvwa/css/login.css	200	OK
9	24.07.19 8:51:12	POST	http://192.168.17.218/dvwa/login.php	302	Found



После выбора этого пункта отобразится попап с настройками локаторов для заполнения. Нам надо определить, где локатор логина и где локатор пароля, дабы ZAP мог понимать, где какое поле ввода, чтобы он мог вписывать туда данные для авторизации. Тут нам надо выбрать из выпадающего списка, что будет применяться в качестве значения в локаторе.



В этом случае нам надо выбрать в первом ДД username, во втором – password. Нажимаем OK. На POST-запросе у вас появится иконка двери.

Интересные статьи

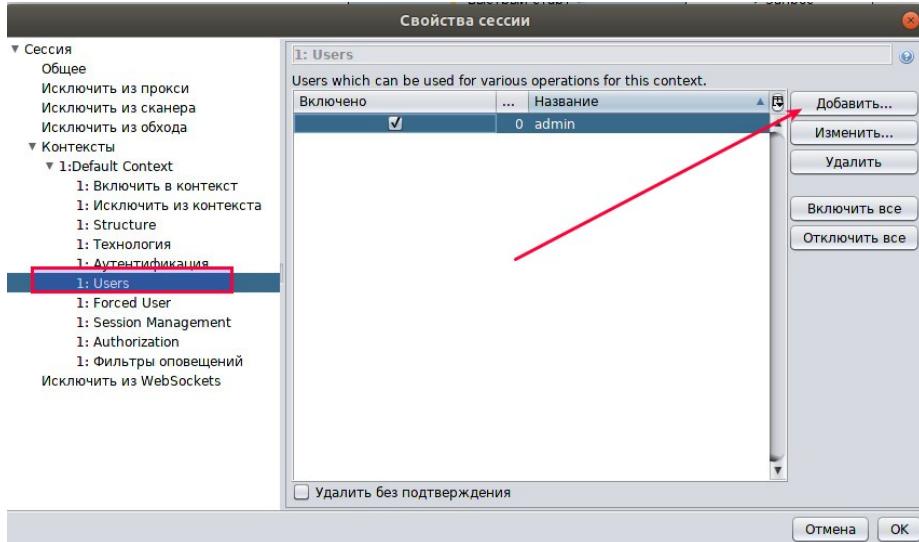


[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

Затем надо создать самого бота (юзера), задать ему креденшельы, которые он будет вводить в те инпуты, настроенные нами ранее. Для этого заходим в раздел Users и в этом разделе добавляем нашего пользователя, который сможет авторизоваться.



В открывшемся попапе задаем этому юзеру имя (какое – не имеет значения, это делается для вашего понимания, что это за юзер), пароль и логин (email), который подходит для авторизации в нашей системе.

Теперь нам надо показать OWASP ZAP какой-то локатор (XPath) на странице авторизованного юзера, чтобы сканер понимал, что авторизация прошла успешно. Для этого нам надо зайти на ответ, полученный от сервера, с HTML-разметкой, которую получает авторизованный юзер. Затем находим какой-то локатор, к которому привяжем наш сканер. Этот локатор OWASP будет искать после выполнения авторизации и понимать, успешно ли он прошел авторизацию на сайте.

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 07:47:02 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2~ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14
X-Powered-By: PHP/5.3.2~ubuntu4.30
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4704
Content-Type: text/html; charset=utf-8
```

```
</div>
</div>
<div id="main_body">
<div class="body_padded">
<h1>Welcome to Damn Vulnerable Web App!</h1>
<p>Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.</p>
```

Этот локатор вставляем в раздел аутентификации, в котором мы задавали параметры входа, – в инпут Regex pattern identified in Logged.

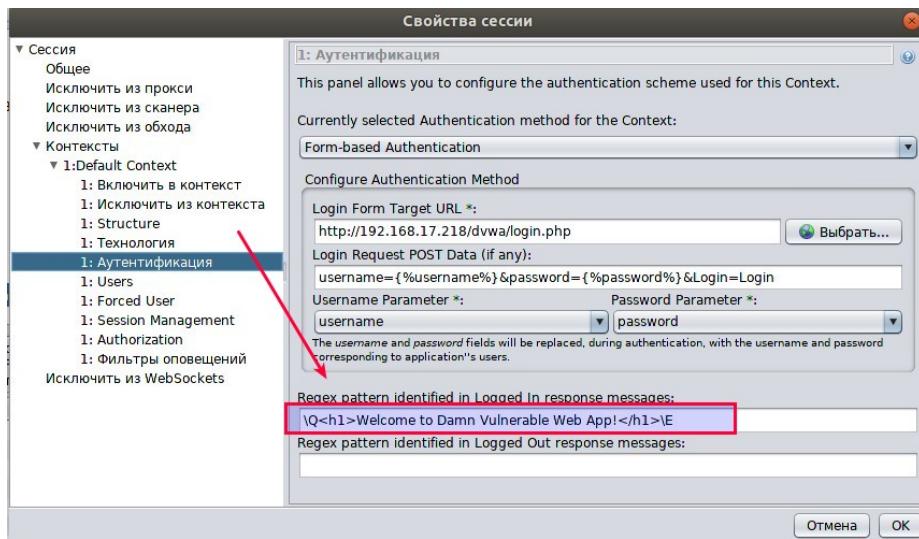
Интересные статьи



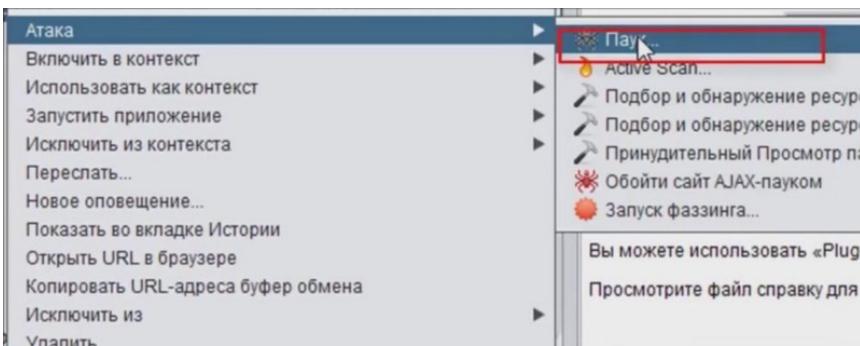
[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



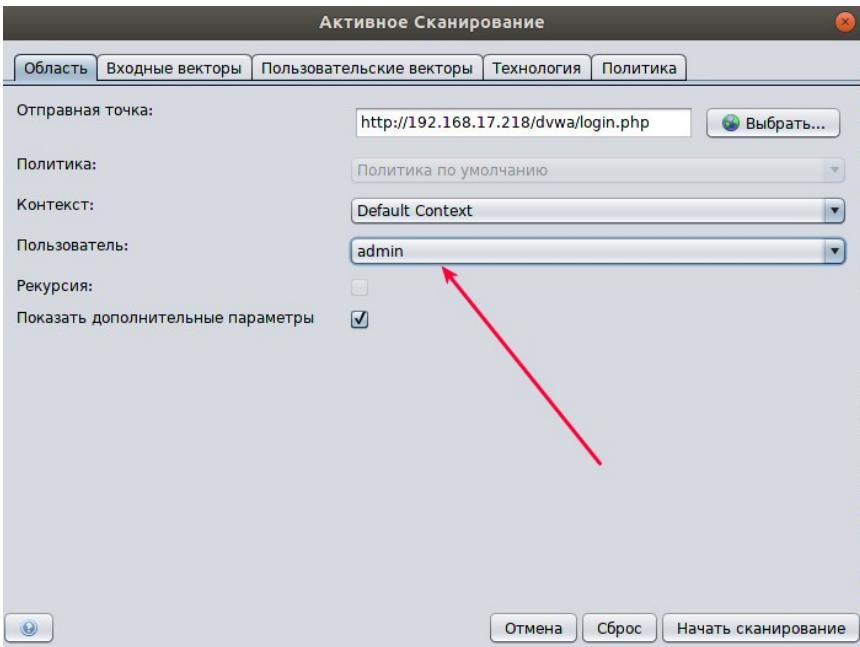
[Почему не у всех получится делать продукты](#)



Как только мы выполнили все, что написано выше, можем начинать атаку и проверку того, на что способно наше приложение. Для этого кликаем по нашей папке, которой задали контекст, и жмем на скан либо атаку.



После этого в выпадающем списке находим нашего пользователя, под которым зайдет сканер.



И вуала! Пошло сканирование вашего проекта на всевозможные уязвимости.

Но помните: никакая из программ не может гарантировать, что будут найдены все уязвимости. Найденные уязвимости могут быть и false-positive, то, что найдет сканер, по-любому надо проверять руками.

Подытожим

Теперь вы знаете:

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

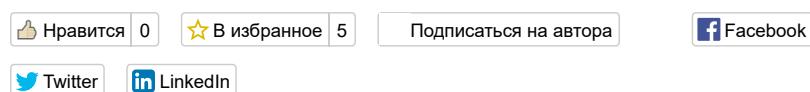
- о двух уязвимостях;
- с помощью какого инструмента можно искать уязвимости;
- где можно практиковаться в поиске уязвимостей.

Теперь ваша задача состоит в том, чтобы, когда сканер найдет новый вид уязвимости, ознакомиться с ним, узнать, как он воспроизводится и какую серьезность несет в себе. Таким образом, встречая все новую и новую уязвимость, обнаруженную сканером или найденную в какой-то статье, вы накопите опыт, с которым можете потом искать эти же дыры в других ваших проектах, дабы сделать продукт безопасным. А также со стороны менеджеров вас заметят и по-любому накинут бабосов за то, что вы такой крутой специалист.

И помните: все показанное выше сделано в целях обучения! Применять на своих проектах можно только с разрешения.

Читайте также: [Зарплаты украинских тестировщиков – июнь 2019](#)

Темы: junior, QA, безопасность, тестирование



Похожие статьи

[Ничего не забыть: универсальная схема для тестирования веб-приложений](#)

[От шока до принятия: пять стадий тестирования API](#)

[Советы сенюров: как прокачать знания junior QA](#)

[5 книжок для QA – початківців та досвідчених, від Юлії Пилипенко, QA Lead в MEGOGO](#)

[Что надо знать Manual QA Trainee, чтобы устроиться на работу.](#)

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты

Свежее

15 февраля · Колонки #C++

Що нового у C++20: можливості та перспективи

15 февраля · Статьи #аналитика

Рейтинг мов програмування 2021: частка Python зменшується, а TypeScript обійшов C++

12 февраля · Колонки #менеджмент

Как я справился с бойкотом от заказчика

12 февраля · Колонки #ментор

12 шагов, как подготовить Junior React Developer к работе на проекте на примере To-Do List App

11 февраля · Колонки #документация

Аудит технічної документації: кому, навіщо, як

Популярное за месяц

10 февраля · 53139

Де в Україні айтішнику жити добре. Рейтинг міст DOU

25 января · 39402

Зарплати українських PM, HR, DevOps, Data Science та інших IT-спеціалістів – зима 2021

11 февраля · 35776

«Демонтаж ФОП-моделі – це шок для IT-ринку України». Що думають IT-компанії про законопроект щодо посилення захисту працівників

11 февраля · 32174

Уряд схвалив проект закону щодо посилення захисту працівників. Що це означає для IT ФОПів

18 января · 28355

Зарплати українських тестувальників – зима 2021

[Все материалы](#) ➔ [Прислать статью](#) ➔



DOU Podcast

Профспілка в Google, офіс Lyft у Києві,...



Share



36 комментарів

Подписаться на комментарії

Коментарии могут оставлять только пользователи с [подтвержденными аккаунтами](#).



[anonymous](#)

08.08.2019 10:16

[Ответить](#)

[Поддержать](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

09.08.2019 17:49

На ютубе прикрыли видосы где показывают как «взломать соседа» , «взломы реальных ресурсов» , «взлом ВК» , «Взлом ФБ» и т.д.
Про ИБ видосы как были так и остались.

[Ответить](#)

[Поддержать](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

07.08.2019 03:10

И что часто встречаются сейчас такие вектора? Ну если серьезно? И не брать во внимание государственные сайты нашей страны. Ну я про реальные современные сайты в клаудах с вафами и секьюрити хидерами для привента XSS , современными браузерами с доп. уровнем защиты от XSS, а не про тренировочные DVWA , bWAPP и т.д.
Интересно услышать среднюю статистику среди людей которые тут пишут комментарии , про ваш опыт и т.д.

[Ответить](#)

[Поддержать](#)



Дмитрий Павлов Penetration tester в [Ciklum](#)

07.08.2019 12:17

Не частно, но встречаются.

[Ответить](#)

[Поддержать](#)



Евгений Толчинский QA Lead в [AMERIA](#)

07.08.2019 12:33

Иногда есть проекты, где единственный браузер пользователей IE, а там фильтрации XSS нет

[Ответить](#)

[Поддержать](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

07.08.2019 12:36

SilverLight и т.д. Проходили... Там в таких кейсах(когда обязуют юзать юзера IE) целая пачка векторов атак возможна)

[Ответить](#)

[Поддержать](#)



Евгений Толчинский QA Lead в [AMERIA](#)

07.08.2019 12:39

100%, но при этом XSS на основной странице, на проде. И находится стандартным скриптом, алерт, скрипт. Без танцев с бубном.

[Ответить](#)

[Поддержать](#)



Vlad Styran Co-founder & VP, Development в [Berezha Security Group](#)

08.08.2019 13:06

В 9/10 проектах есть XSS.

Больше ніж в половині їх більше ніж в одному місці.

[Ответить](#)

[Поддержать](#)

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

09.08.2019 17:18

Ну логично, что если нету валидации в одном месте то и в другом врятли будет) Хотя бывает) Знаю

[Ответить](#) [Поддержать](#)



Vlad Styran Co-founder & VP, Development в [Berezha Security Group](#)

09.08.2019 17:20

Ну да, якшо воно погано, то скрізь. За деякими виключеннями, де є ad hoc fix.

[Ответить](#) [Поддержать](#)



Kyrylo Romanenko QA Engineer

08.08.2019 18:27

И не брать во внимание государственные сайты нашей страны.

А почему не брать, и почему только нашей?

Попался мне не так давно в ненашей стране реальный коммерческий сайт по продаже билетов, который при регистрации пользователя все данные шлёт по HTTP открытым текстом.

[Ответить](#) [Поддержать](#)

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты



Vadim

06.08.2019 20:26

Добрый вечер, интересная статья, не имея никакого опыта работы, хотел бы спросить как с нуля войти в сферу безопасности? Скажем не с QA тестировщика, а к примеру в направлении penetration testing?

[Ответить](#) [Поддержать](#)



Egor Papyshov никто

06.08.2019 23:33

достаточно таких статей. Просто подпишитесь на автора и через пару месяцев станете полноценным пентестером, желанным кандидатом в яркую тиму хакеров. Возможно, в следующих выпусках автор расскажет по понятиям за отрицалово и другие аспекты жизни пентестера.

Бей актив реж-сук

[Ответить](#) [Поддержать](#)



Vlad Styran Co-founder & VP, Development в [Berezha Security Group](#)

07.08.2019 01:12

З нуля в безпеку не входять, це не ентрі-левел спеціальність. Вибачте, що розчарував :) Треба мати базис, в ідеалі, вже якусь IT-шну спеціальність і досвід. Як мінімум, знати, як працює комп'ютер, мережі, протоколи, софт – достатньо початкового рівня. З цієї платформи можна вже кудись рухатись. Якщо тягне в appsec, то логічно знайти найближче територіально відділення OWASP та почати заводити зв'язки серед його учасників. Інфо по українських чаптерах тут: www.owasp.org/...ndex.php/Category:Ukraine

Якщо тягне в щось менш спеціальне, як то пентестінг загалом, то тут все ще простіше: шукайте профільні конференції та мітапи, та починайте відвідувати. Далі все саме піде, якшо ґав не ловити.

[Ответить](#) [Поддержать](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

07.08.2019 02:51

Берешь и заходишь.

[Ответить](#) [Поддержать](#)



Vadim

07.08.2019 14:18

спасибо, рад это слышать)

[Ответить](#) [Поддержать](#)



Vlad Styran Co-founder & VP, Development в [Berezha Security Group](#)

[Berezha Security](#)

09.08.2019 17:41

Будь кстаті хороший приклад того, як це робиться ггг

[Ответить](#)

[Поддержать](#)



nea Alecu разрешение

Годнота на ДОУ!

[Ответить](#)

[Поддержать](#)

06.08.2019 18:12

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Vlad Styran Co-founder & VP, Development в [Berezha Security Group](#)

[Berezha Security Group](#)

06.08.2019 16:41

Красавчик. Продовжуй :)

[Ответить](#)

[Поддержать](#)



Почему не у всех получится делать продукты



Viktor Gogol QA в [Edvantis](#)

Спасибо за статью! Очень интересно!

[Ответить](#)

[Поддержать](#)

06.08.2019 14:34



Pavel Kryvko DevSecOps Engineer в [ЕРАМ](#)

Проблем найти нужную литературу вообще нет, есть 2 замечательных книги, которые охватывают достаточно большую область связанную с безопасностью веб-приложений.

1. Web Application Hacker's Handbook: www.amazon.com/...-Exploiting/dp/1118026470

2. The Browser Hacker's Handbook: www.amazon.com/...Wade-Alcorn/dp/1118662091

Также прочтение данных книг, помимо получения качественных знаний, может исцелить от мракобесия, которое распространяет автор.

Как вариант, узнать что «уязвимости» это не инфекция и «заразиться» ими не представляется возможным даже будучи в очень тесном контакте.

Nuff said.

[Ответить](#)

[Поддержать](#)

06.08.2019 14:18



Svyat Login QA gangsta Lead в [EVO](#)

06.08.2019 14:40

Полностью согласен с вами))) Читать статью DevOpsy в таком виде, которая подана для manual QA, конечно же не солидно. Прочесть книги можно)) но в книге не показано где можно потренироваться в применениях полученных навыков. Также скажу, что эти книги не рассчитаны для тех ребят, которые не понимают в тестировании безопасности. Спасибо за критику.

[Ответить](#)

[Поддержать](#)



Сергей Харюк Security Researcher

06.08.2019 14:50

Если если специалист по ручному тестированию хочет получить практические знания от людей которые действительно разбираются в этом, то лучше присоединится к комьюнити чатам в которых с вами поделятся знаниями бесплатно. Более подробно ознакомится с различными комьюнити можно тут:
dc8044.com
github.com/...8044/dc8044-useful-things

[Ответить](#)

[Поддержать](#)



Pavel Kryvko DevSecOps Engineer в [ЕРАМ](#)

06.08.2019 14:51

Эти книги как раз рассчитаны на тех кто ничего не понимает в безопасности веб приложений, что бы это понять достаточно прочитать оглавление и 2 первых главы. Практические навыки – это отдельная тема, которые, кстати, эта книга тоже затрагивает. И при чем тут DevOps в чем логика этого тезиса

Читать статью DevOpsy в таком виде

? :)

[Ответить](#)

[Поддержать](#)



Svyat Login QA gangsta Lead в [EVO](#)

06.08.2019 14:58

Когда, я читал эти книги 4 года назад, когда я был просто мануалом, мне было трудно воспринимать некоторые аспекты, когда я поднабрался опыта и прочел еще раз эти книги, мне уже стало понятнее, про что те аспекты были написаны. А про девопс, я имел ввиду, что такой формат статьи для девопса не интересен))

[Ответить](#)

[Поддержать](#)



Pavel Kryvko DevSecOps Engineer в [EPAM](#)

06.08.2019 15:06

Я думаю можно изменить DevOps на любую другую лычку и смысл не изменится, дело не в специализации а в информации. А какие именно аспекты было трудно воспринимать, стесняюсь спросить?

[Ответить](#)

[Поддержать](#)



Bohdan Lukin Security Engineer в [SoftServe](#)

07.08.2019 02:57

Книг овер дохера на просторах тырнетах , есть те которые можно читать как книгу а есть те которые обычно юзатся как настольный справочник , где вычитывается содержания и по необходимости делятся дип дайв в раздел книги который возможно поможет решить возникший кейс. Но как правило даже без книг все гуглится и находится. Единственная проблема это время. Если знакомишься с чем то новым – на познания уйдет больше времени , что впрочем как и везде, не только в ИБ. Эт мое имхо.

[Ответить](#)

[Поддержать](#)



Евгений Толчинский QA Lead в [AMERIA](#)

06.08.2019 15:13

Я читал WebApp hacker и есть много вещей не понятных. Когда читаешь статью Святослава, его блог, смотришь видосы мануальщиков, становится понятнее. Статья расчитана на мануальщика, у которого секьюрити 10% от его работы, а то и просто хобби и начинать с OWASP testing Guide или WebApp hacker скорее оттолкнет его. И стоит напомнить у части мануальщиков образование нетехническое. Я с более базовыми вещами по секьюрити выступал не один раз и после (в отзывах спустя пару недель) каждого доклада кто-то пишет нашли sql injection, XSS итд. П.С. не нравится статья – напишите лучше

[Ответить](#)

[Поддержать](#)



Pavel Kryvko DevSecOps Engineer в [EPAM](#)

06.08.2019 15:19

OWASP testing Guide или WebApp hacker скорее оттолкнет его

Если оттолкнет это к лучшему, правда. Будет меньше людей говорить об инфекционных веб уязвимостях и возможно цивилизованный мир начнет серьезно смотреть на экспертизу секьюрити в Украине.

[Ответить](#)

[Поддержать](#)



Евгений Толчинский QA Lead в [AMERIA](#)

06.08.2019 15:24

Если manual QA найдет простейшую уязвимость, это покажет его компетенцию клиенту. Это не замена секьюрити инженеру, это доп скил мануальщика.

[Ответить](#)

[Поддержать](#)



Pavel Kryvko DevSecOps Engineer в [EPAM](#)

06.08.2019 15:28

И что он скажет? «Я нашел sql-injection это инфекция которая...» а потом добавит «Йа хацкер...» Еще раз, дело в мракобесии, а не специализации.

[Ответить](#)

[Поддержать](#)



Евгений Толчинский QA Lead в [AMERIA](#)

06.08.2019 15:29

Интересные статьи



[DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»](#)



[Почему не у всех получится делать продукты](#)

Он скажет я нашел инъекцию, может мы ее пофиксим перед релизом? и может быть меньше говна будет в нете.

[Ответить](#)

[Поддержать](#)



Pavel Kryvko DevSecOps Engineer в

[EPAM](#)

06.08.2019 15:32

Сомневаюсь что будет именно так, на деле все обстоит иначе:) Я не совсем понял как «говно» стыкуется с уязвимостями?

[Ответить](#)

[Поддержать](#)

Интересные статьи



DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Почему не у всех получится делать продукты



Евгений Толчинский

QA Lead в

[AMERIA](#)

06.08.2019 15:36

1. от своих QA я слышу такую формулировку
2. Говно – сайты, которые взламывать особо не надо.
простейшая инъекция или XSS отрабатывает. Таких вагон. Смотришь путевку или гостишку, или интернет магазин за границей, не большой, на разных сайтах и оп <> не вырезаются, закрываешь сайт – я таким пользоваться не хочу.

[Ответить](#)

[Поддержать](#)



Pavel Kryvko

DevSecOps Engineer в

[EPAM](#)

06.08.2019 15:41

И много у вас опыта реальной эксплуатации веб приложений вот прям через XSS ?
Это не говно, а сайты которые тестировали коллеги, цитирую «фраеры-хацкеры» Святослава после прочтения подобных материалов.

[Ответить](#)

[Поддержать](#)



Kyrylo Romanenko

QA Engineer

06.08.2019 21:19

Для практического ознакомления с вопросом сойдет. Конечно, если это единственная мера безопасности на проекте, то все очень плохо.

[Ответить](#)

[Поддержать](#)

Советуем почитать

[Графические акселераторы для высокопроизводительных вычислений. Часть 2](#)

[Не просто PM, а консультант. Как эволюционирует менеджер проектов](#)

[Что может подстерегать новичков при работе с SQL Server](#)

[Перші кроки в NLP: розглядаємо Python-бібліотеку scikit-learn в реальному завданні](#)

От джуниора к лидеру. Какие навыки нужны для роста в профессии

Интересные статьи



Не поспішаємо з реалізацією. Чому співробітники не можуть працювати токсичними та що можна зробити

DOU Ревізор у Львові: «Кубрики для чотирьох в опенспейсі GlobalLogic»



Контакти
Почему не у всіх подується делать
продукты

Автоматизация процессов в Support: сокращаем время первого ответа на 30%

Много – не всегда хорошо. В чем суть достаточного тестирования и как его использовать правильно

Vert.x + Micronaut. Для чего нам Dependency Injection у світі мікросервісів

Резюме и собеседования: взгляд кандидата и интервьюера

© 2005–2021 [DOU.ua](#)

Українська · [Русский](#) · [English](#)

Нас уже 417 453. Мы в соцсетях:



[Поиск программистов на Джинне](#)