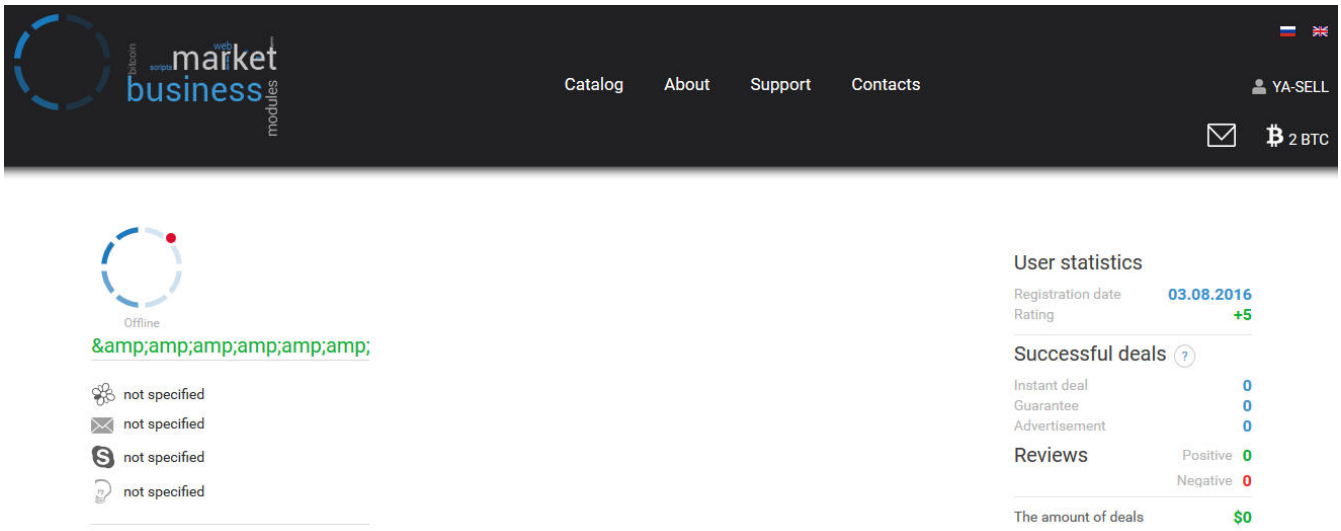
3.1.0 Логические ошибки SMARKET

Возможность порвать верстку страницы.
Проблема: отсутствие проверки **длины** введенного имени.

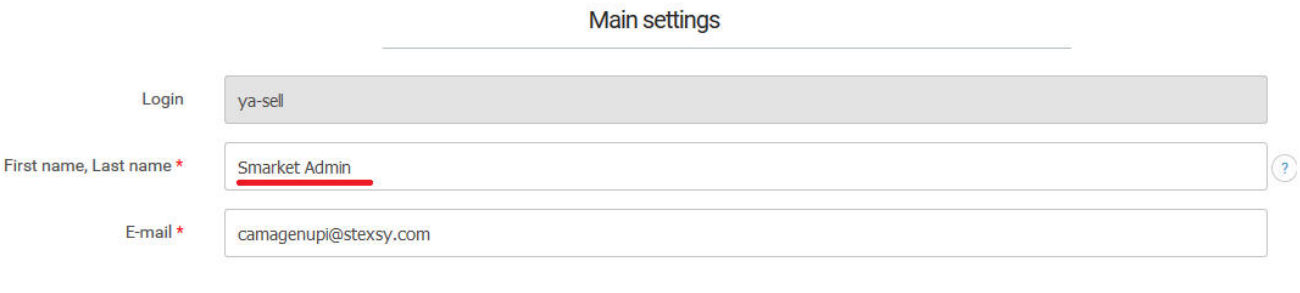
http://smarket.biz/user/BigBear

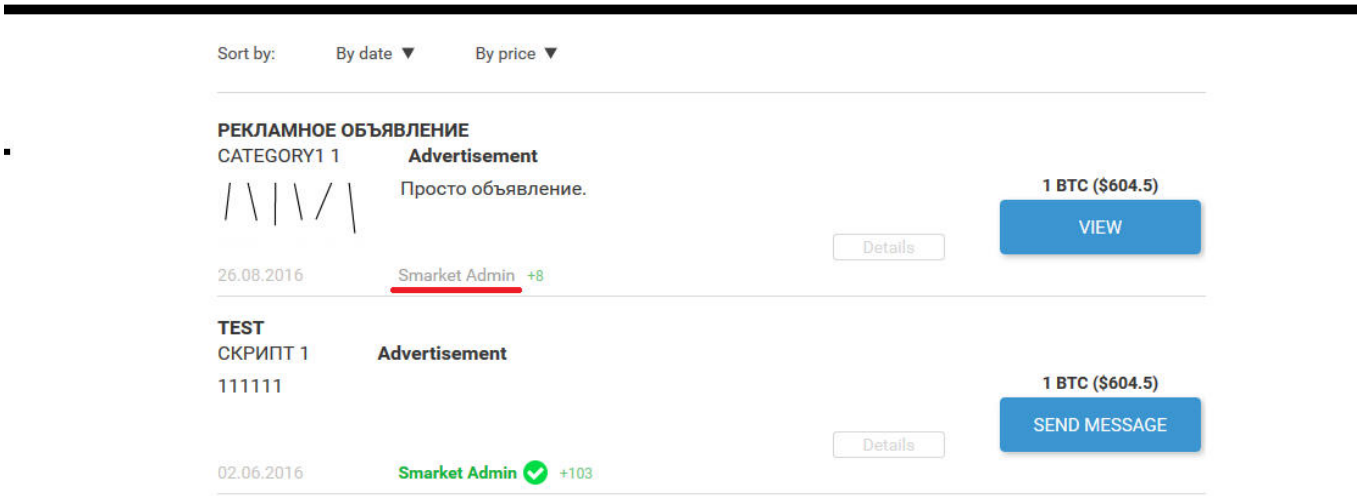


Рекомендация: добавить **LIMIT** в поле "Имя Фамилия". Органичение количества символов.

3.1.1 Обман пользователя (Социальная инженерия)

Проблема:
Поле "Имя, Фамилия" позволяет указать одинаковые имена, ники. Это дает 100%-ную возможность обмануть пользователя, представившись другим человеком.





Рекомендация: установить проверку уникальности введенных данных.

3.1.2 TS-SMARKET-XSS-01: Множество XSS уязвимостей

Критичность: medium

Требуется доступ в систему для обнаружения: нет, не требуется

Требуется доступ в систему для эксплуатации: нет, не требуется

Перечень уязвимых ресурсов:

http://smarket.biz

1. **Множественная** уязвимость внедрения JavaScript-кода (Reflected XSS) на страницах Веб-ресурса smarket.biz

Примечание: Многие современные браузеры (Например: Google Chrome, Yandex Browser) имеют встроенную защиту от атак класса XSS. Поэтому заданный вектор атаки в них работать не будет. Для демонстрации уязвимости, необходимо использовать браузер Internet Explorer 8.

Уязвимый параметр: небезопасная вставка URL в тело HTML тэга .

Последовательность действий для демонстрации уязвимости

Для демонстрации уязвимости достаточно перейти по нижеприведенной ссылке:

http://smarket.biz/catalog/filter"><script>alert(1)</script>

В результате сервер вернет страницу, в которую будет вставлено значение переданное в URL (см. Рисунок 2):

```
</div>                                <div class="pagination-wrapper"><div class="pagination"><ul
class="pagination-br"><a href="/catalog/filter"><script>alert(1)</script>?page=1"><li></li></a><a
href="/catalog/filter"><script>alert(1)</script>?page=1"><li class="active">1</li></a><a href=
"/catalog/filter"><script>alert(1)</script>?page=2"><li >2</li></a><a href="/catalog/filter"
><script>alert(1)</script>?page=3"><li >3</li></a><a href="/catalog/filter"><script>alert(1)
</script>?page=3"><li >></li></a></ul></div></div>                          </div>
                </div>
```

Рисунок 2 – Вывод параметра без кодирования спецсимволов в исходном коде процедуры навигации

На появившейся странице средствами HTML будет отображено диалоговое окно (см. Рисунок 3).

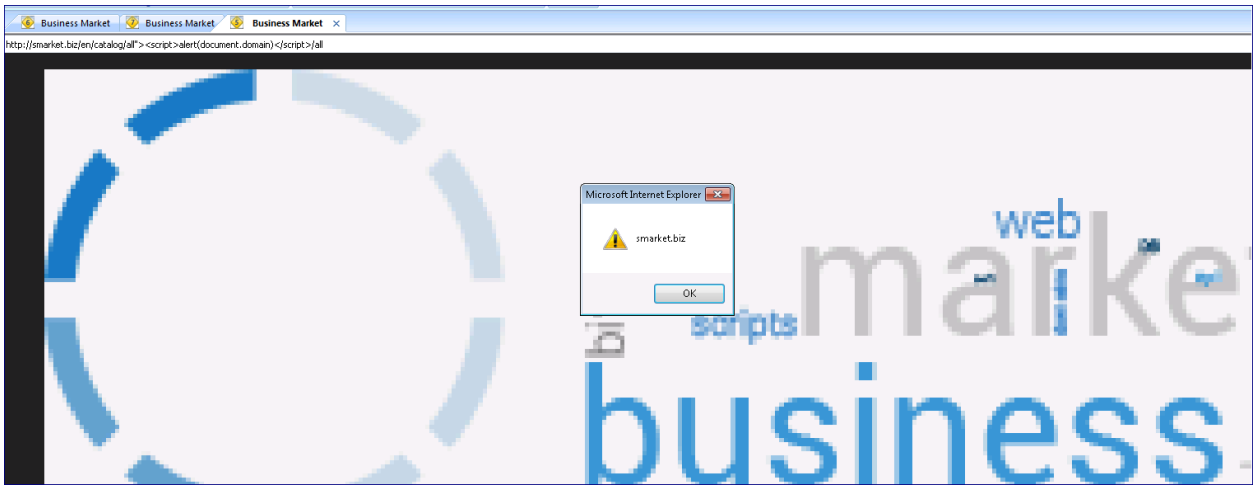


Рисунок 3 - Выполнение JavaScript-кода в теле HTML текста процедуры навигации

Рекомендации по устранению

Необходимо осуществлять валидацию полученных данных из URL и привести их к ожидаемому типу данных. Все данные, выводимые в HTML, обязательно должны кодироваться по правилам предотвращения XSS вне зависимости от наличия валидации данных при

получении. В данном случае пользовательские данные выводятся в тело HTML тэга (см. п.3.1.1 Правило №2).

Уязвимый параметр: dateto в сценарии

/en/catalog/filter

Последовательность действий для демонстрации уязвимости

Для демонстрации уязвимости достаточно послать POST запрос с следующим содержимым:

POST /en/catalog/filter HTTP/1.1

Content-Length: 101

Cookie:YII_CSRF_TOKEN=8fa8e7b3d47291994d671dd5e448e91ad2f15098s%3A40%3A%226b3094168352bc39c322ed39f0c0823a07027c49%22%3B

Host: smarket.biz

dateto=1<script>alert()<script>&YII_CSRF_TOKEN=6b3094168352bc39c322ed39f0c0823a07027c49

В результате сервер вернет страницу, в которой значение параметра **dateto** будет вставлено на страницу небезопасным способом (см. Рисунок 4):

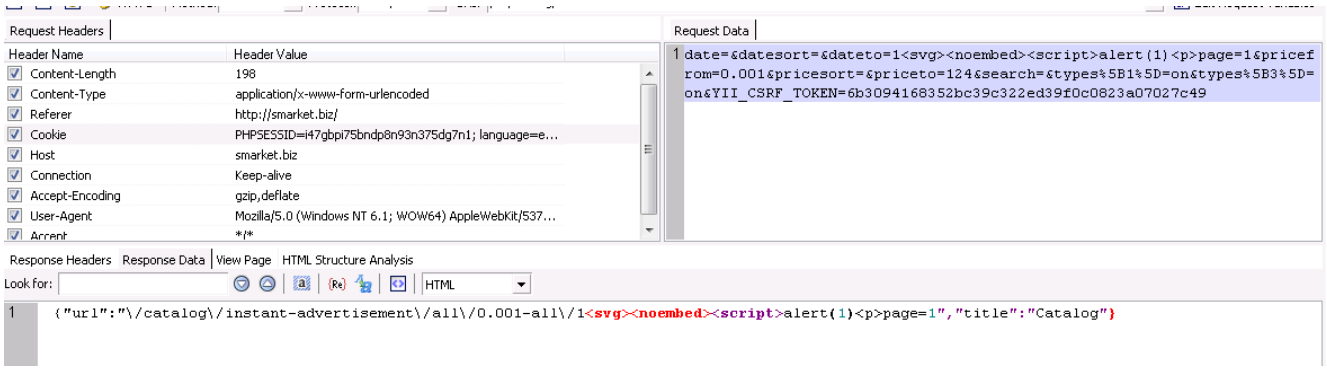


Рисунок 4 - Вывод параметра без кодирования спецсимволов в исходном коде страницы поиска товара

Так как возвращаемый Content-Type имеет значение text/html, то данная XSS успешно отработает в браузере Internet Explorer ниже версии 8. Для более поздних браузеров результат поиска будет возвращен в формате JSON, который воспрепятствует эксплуатации уязвимости.

Рекомендации по устранению

Необходимо осуществить валидацию полученных данных в параметре dateto и привести их к ожидаемому типу данных. Все данные, выводимые в HTML, обязательно должны кодироваться по правилам предотвращения XSS вне зависимости от наличия валидации данных при получении. В данном случае пользовательские данные выводятся в значение атрибута html-тэга (см. п.3.1.1 Правило №6).

Анализ критичности

Уязвимость на Веб-ресурсе может привести к различным последствиям, в зависимости от группы пользователей, попавших под действие этой уязвимости:

— Для посетителей Веб-ресурса (не авторизованных пользователей), а также для всех групп - может привести к внедрению произвольного HTML текста в браузерах Клиентов Веб-сервера, а также, перенаправлению на произвольную страницу сети Интернет и любым другим действия, реализуемые средствами JavaScript.

— Для пользователей Веб-ресурса (прошедших авторизацию) - дополнительно может привести к неконтролируем действиям (например, покупкам), а также эксплуатации любых других механизмов обмена данными в рамках пользовательской сессии.

— Для администраторов Веб-ресурса – перечень рисков аналогичен, что и для пользователей Веб-ресурса.

За счет того, что сессионные Cookie содержат флаг HttpOnly, получение пользовательских сессий, через описанную уязвимость – не возможен.

Уровень уязвимости оценивается, как **средний**.

3.2 Прочие исследования

3.2.1 TS-SMARKET-IA-01: Возможность автоматизации перебора паролей в форме авторизации на основе отсутствия значения Captcha

Критичность: medium

Требуется доступ в систему для обнаружения: нет, не требуется

Требуется доступ в систему для эксплуатации: нет, не требуется

Данный недостаток позволяет злоумышленнику автоматизировать перебор паролей в форме авторизации Веб-ресурса:

smarket.biz

Место уязвимости в рейтинге OWASP Top 10 2010: нет.

Идентификатор уязвимости в CWE: CWE-804.

Идентификатор атаки в WASC Threat Classification v2: WASC-21.

В данном случае злоумышленник может произвести подбор данных логина и пароля пользователя

Последовательность действий для демонстрации уязвимости

Для подтверждения наличия уязвимости достаточно убедиться, что отсутствуют изображения, выдаваемые технологией Captcha.

Пример авторизации:

http://smarket.biz/ru/user/login

Результат выполнения представлен на рисунке ниже (см. Рисунок 5).

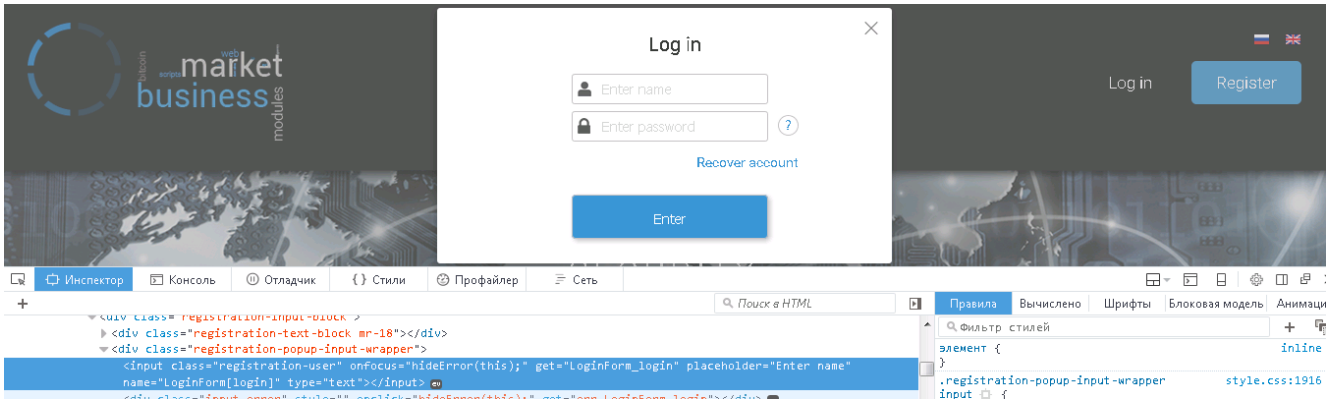


Рисунок 5 - Страница входа в пользовательский интерфейс

Анализ критичности

Исходя из возможности проведения процедуры прямого перебора данных для авторизации на Веб-ресурсе – статус критичности определен, как **средний**.

Рекомендации по устранению

Реализовать внедрение механизма динамической генерации Captcha (например, использовать любую публичную общедоступную технологию, такую как reCaptcha).

3.2.2 TS-SMARKET-WKN-01: Раскрытие версии Программного Обеспечения

Критичность:info

Требуется доступ в систему для обнаружения: нет, не требуется

Требуется доступ в систему для эксплуатации: нет, не требуется

Позволяет злоумышленнику раскрыть техническую информацию о Веб-сервере через получение заголовка с указанием установленной версии ПО, далее на основании полученных данных подобрать нужный вектор атаки на Веб-приложение.

Место уязвимости в рейтинге OWASP Top 10 2010: нет.

Идентификатор уязвимости в CWE: CWE-205.

Идентификатор атаки в WASC Threat Classification v2: WASC-45.

Последовательность действий для демонстрации уязвимости

Уязвимый сервис: smarket.biz

Уязвимый параметр: Server

Пример ответа Веб-сервера:

HTTP/1.1 200 OK

Date: Mon, 19 Sep 2016 14:22:12 GMT

Server: Apache/2.4.10 (Debian)

X-Frame-Options: SAMEORIGIN

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

Vary: Accept-Encoding

Content-Length: 11974

Connection: close

Content-Type: text/html; charset=UTF-8

Анализ критичности

Данная ошибка (конфигурации Веб-сервера) раскрывает конфиденциальную информацию, которая может повлечь за собой эксплуатацию актуальных уязвимостей под конкретные версии ПО. Однако, таких уязвимостей может и не быть, а их эксплуатация может затрудняться различными дополнительными факторами, например, внедренным межсетевым экраном. Исходя из этих фактов, статус критичности определен, как **низкий**.

Рекомендации по устранению

Удалить указанные заголовки из ответов Веб-сервера.

Данная проблема решается следующим образом:

Для Apache Tomcat не предусмотрено стандартных процедур удаления заголовка `Server`, но разрешено менять его содержимое через внесение произвольного значения переменной server в конфигурационный файл. Например, так: `<Connector port="80" protocol="HTTP/1.1" ... x-powered-by="false" ... server="Any"/>`. Также нужен перезапуск сервиса. После этого в заголовке ответа Веб-сервера вместо «Apache/2.4.10 (Debian)» будет показываться «Any».