Инструменты.

Любой анализ кода конечно же можно проводить вручную, но в большинстве случаев несколько нижеприведённых инструментов могут сильно облегчить исследование приложения. Рассмотрим сначала инструменты для наружного исследования веб-приложений.

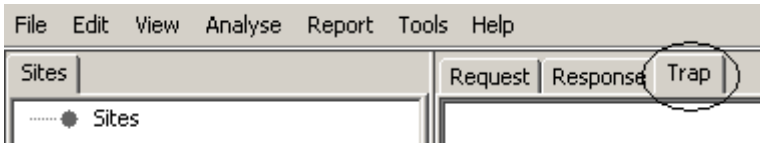
Paros.

Paros – это HTTP/HTTPS proxy-сервер написанный на языке Java и предназначенный для анализа и изменения HTTP-пакетов "на лету". Он может сильно пригодится при анализе веб-приложений на наличие в них уязвимостей. Далее будет рассмотрено большое количество примеров с использованием именно этой программы. Paros Вы можете скачать с официального сайта:
http://www.parosproxy.org/download.shtml
или же взять на диске в папке /programs/. Самая последняя версия, на момент написания этой книги, 3.2.13.

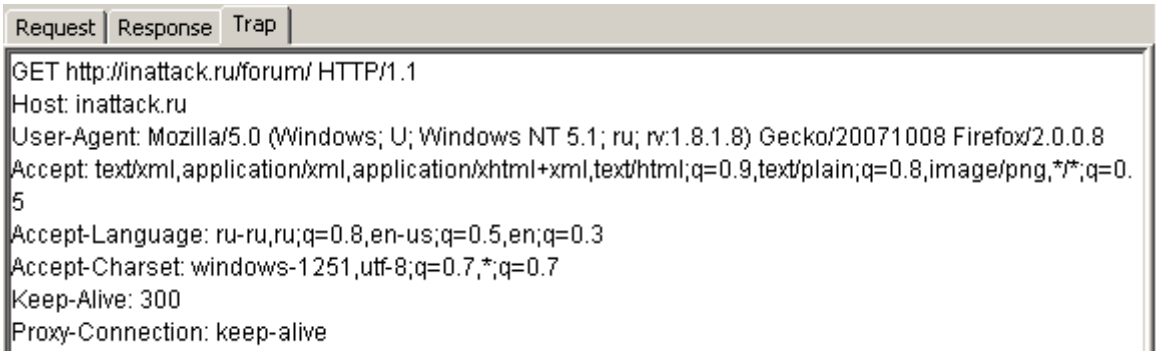
Paros написан на Java поэтому для запуска понадобится соотвествующий Java-пакет. Его Вы можете найти в том же каталоге что и Paros. Файл установки называется "jre-1_5_0_07-windows-i586-p.exe". Процесс установки достаточно простой и проблем не создаст.

Paros имеет множество функций, но мы будем использовать только перехват запросов к серверу и от него. Эта функция, как мне кажется, является самой полезной в Paros. В каких же ситуациях бывает полезна эта возможность? Таких ситуаций довольно много, но в основном это либо работа с большим количеством данных передающихся методом POST, либо попытки ручной эксплуатации сложной уязвимости. Например Вам понадобилось проверить на определённом сайте наличие фильтрации полей формы которая содержит 7 видимых полей и 8 скрытых. В похожих ситуациях новички чаще всего просто сохраняют страничку на жёсткий диск и редактируют значение полей. Вариант тоже правильный но если сервер проверяет заголовок "Referer" то все старания не принесут пользы. К тому же этот метод довольно долгий и нудный. Если же в такой ситуации использовать Paros то Вы сможете задержать http-запрос, перед отправкой на сервер, отредактировать в нём значения всех переданных полей уже отправить дальше.

Сейчас я опишу способ перехвата http-запроса на сервер. Запустите Paros и в правой части открывшегося окна выберите закладку "Trap":



Снизу Вы увидите 2 пункта – "Trap request" и "Trap response". При отметке первого пункта Paros станет перехватывать все пользовательские запросы проходящие через него и ждать Ваших действий. Вторая галочка – перехват ответа сервера. Чаще всего данная функция используется для анализа пере-направлений, приходящих с веб-сервера . Поставьте галочку в пункте "Trap request". Затем настройте браузер на работу через Paros (адрес 127.0.0.1, порт 8080) и пройдите по любому адресу. Тут же выскочит окошко Paros`a. Вот пример запроса на адрес http://inattack.ru/forum/:



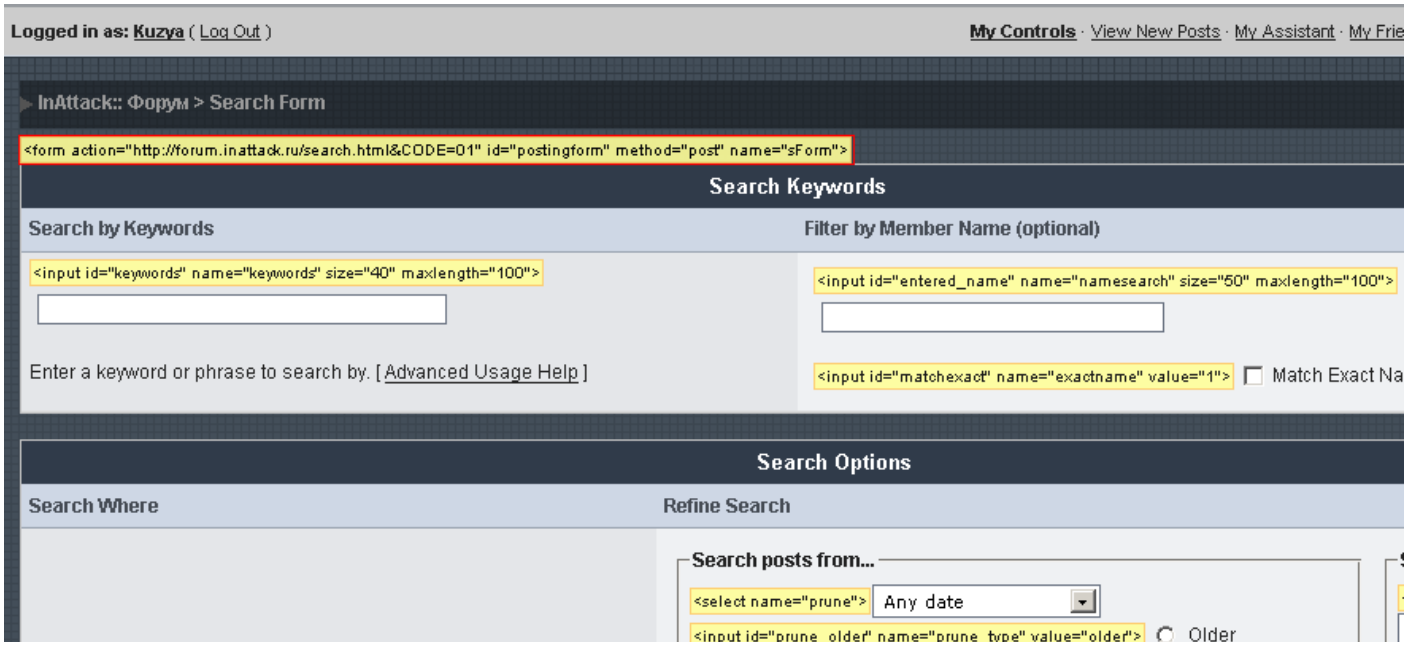
Как видите – здесь представлены все заголовки которые браузер отправляет серверу. Вы можете полностью их редактировать, добавлять или убирать любой заголовок. После окончания редактирования нажмите кнопку "Continue" (кнопка "Drop" обрывает запрос) и запрос уйдёт на сервер. Обратите внимание на то что при отправке браузером POST-данных их содержимое появится снизу, в поле которое находится под окном редактирования заголовков. С перехватом ответа сервера всё полностью аналогично. В этой книге мы рассмотрим пример с использованием Paros всего один раз, но данным продуктом Вы обязаны уметь пользоваться если хотите всерьёз заниматься анализом веб-приложений.

WebDeveloper

WebDeveloper представляет из себя плагин к браузеру MozillaFirefox. Предназначен он, конечно же, для мирных целей веб-разработчиков, но и взломщикам от него может быть не мало пользы. Скачать его Вы можете пройдя по этой ссылке:
http://chrispederick.com/work/web-developer/
После установки в Mozilla Firefox появится дополнительная панель состоящая из 12 меню. Сейчас мы рассмотрим только

те которые могут как-то помочь при поиске уязвимостей. Первое такое меню - "cookies". В нём Вы сможете посмотреть все cookies относящиеся к текущей странице,отредактировать их или добавить новые:

Изменение cookies с помощью WebDeveloper намного удобнее чем, например, с помощью Paros. Следующая интересная закладка — forms. Зайдите на любую страничку с формой и включите в меню forms пункт "Display Form Details". Вы сразу же увидите прямо на страничке все данные об этой форме:

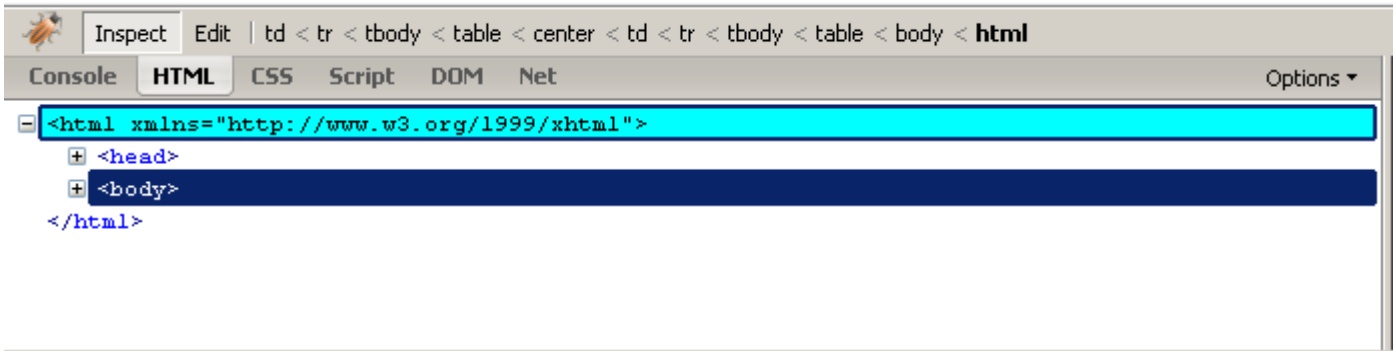


Так же очень много полезного Вы сможете найти во вкладке "Information".

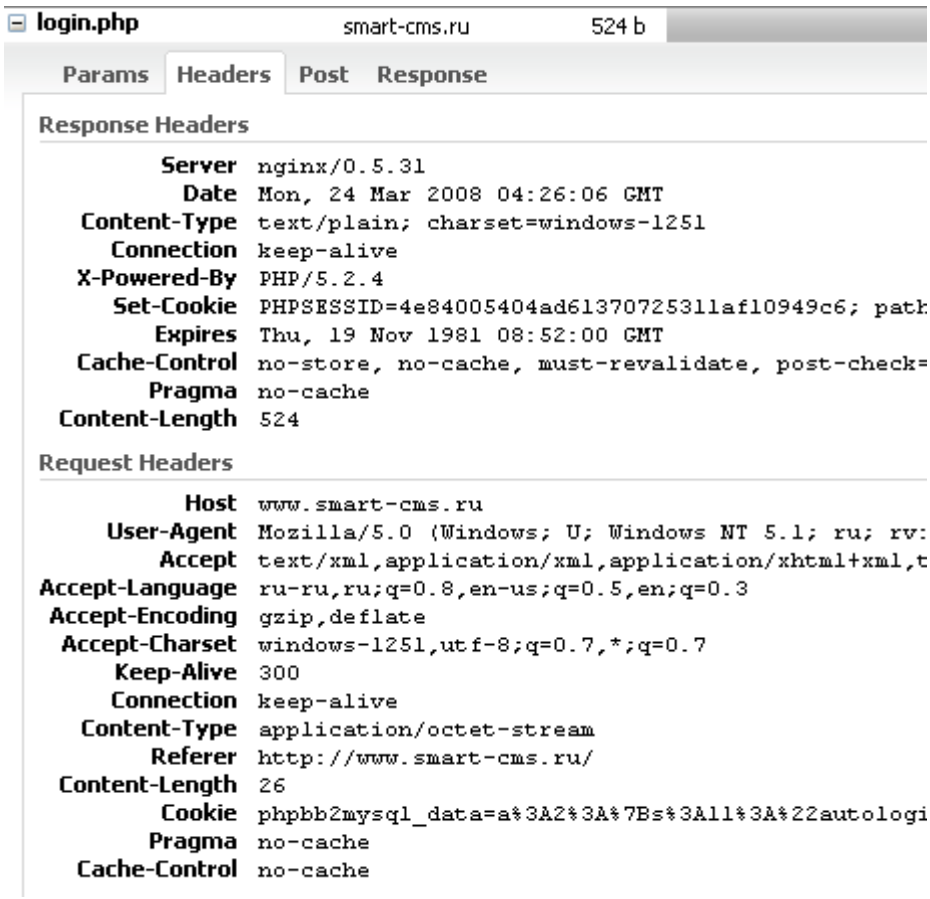
Firebug.

Firebug (http://www.getfirebug.com/), так же, представляет из себя плагин к браузеру MozillaFirefox. Он, как и WebDeveloper, позволяет просматривать JS/HTML-код текущей страницы, изменять их и т.д. Но самое интересное в нём — полный контроль сетевой активности браузера. То есть Firebug фиксирует любые обращения к каким-либо серверам, в том числе и через компонент XMLHTTP, что может очень сильно помочь в ситуациях когда большая часть сайта работает с технологией AJAX.

После установке Firebug Вы можете открыть его окно зайдя в меню "Инструменты->Firebug->Open Firebug". В низу у Вас появится основное окно данного плагина:



Как видите — здесь есть множество закладок. Перейдя на закладку "Net", а затем на вкладку XHR(XmlHttpRequest) Вы сможете наблюдать всю активность Вашего браузера связанную с этим компонентом. Для примера зайдите на сайт smert-cms.ru и попробуйте авторизироваться. В то же время наблюдайте за вкладкой XHR. Вы увидите что появилось одно обращение к login.php. Если Вы откроете список связанный с этим обращением то увидите все его подробности:



На этих четырёх вкладках Вы найдёте всё что Вам нужно.

Документация.

При анализе любого приложения очень может помочь документация по нему. А именно — по его внутреннему устройству. Это может быть, например, документация по API приложения. Даже если документации по нужному продукту нет вообще (а это встречается довольно часто) то стоит просмотреть любые технические обсуждения в интернете. К примеру это могут быть темы на форуме данного продукта связанные с написанием собственных модулей. В таких темах разработчики приложения часто сами дают описания каких-либо функций. Зная предназначение хотя бы самых часто-используемых функций Вы сможете значительно выиграть во времени.