

# Forum:[Hack Forums](#)

Thread: [\[TuT\] Introduction to Template Injection](#)

[BLANK] - April 26, 2017, 6:02 pm

## Template Injection Introduction

### Introduction

I decided to write this tutorial for a couple of reasons. First, Template Injection is not a well known exploit but it is used on many CTF and is an interesting avenue for exploiting a web server, this tutorial is mostly to educate new application testers and give them a new tool for their toolbox. Second, this tutorial will be my first major contribution to the Hacker Forums.

Template Injection is a method of taking advantage of an off the shelf commercial Template Engine used by web application developers to provide dynamic content. Examples include:

FreeMarker  
Velocity  
Smarty  
Twig  
Jade

Unsafely embedding user input in templates enables Server-Side Template Injection, a frequently critical vulnerability that is extremely easy to mistake for Cross-Site Scripting (XSS), or missed entirely. The following code is a basic example of a Template Injection:

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

As we can see from the code above, output is a dynamic variable that will take user input to generate a message using the first name of the user based on the email address provided by the user. A application tester may immediately think of using XSS against the above code, and that may work, but we want to test for a more subtle attack.

```
Code custom_email={{7 * 7}}
```

output = 49

In the above example we sent a server side code execution that was resolved from inside the sandbox. Some template engines will allow us to pop out of the sandbox and exploit arbitrary code.

### Detect Plain Text

Most template languages support a freeform 'text' context where you can directly input HTML. Manual testing of user inputs on dynamic pages is common when testing for XSS vulnerabilities. Often when a XSS vulnerability is found the application is also vulnerable to Template Injection. When testing for template injection there are some common payloads to attempt.

```
Code www.example.com/invalid{{7*7}}
```

The attack above appends the template injection on a resource that generates a dynamic error page. The error page should show the error message + '49', if it is vulnerable. Obviously seeing a

number does not excite an attacker, we can also use `{{ config.items() }}` to generate configuration information.

## Exploitation

Once you find an application that is vulnerable to Template Injection there are a few general steps to follow:

1. Read: find technical documentation on the template you are exploiting. All the template engines have some vastly different syntax so each attack payload will be different.
2. Explore: play around with the applicaiton and test for exploitable default objects and custom objects.
3. Attack: developpe a functional payload and own that application!

## Automate Detection

There are several automated tools that detect and exploit Template Injection including [TPL MAP](#)

## Further Reading

[Black Hat Paper](#)

<https://nvisium.com/blog/2016/03/09/exploring-ssti-in-flask-jinja2/>

**CyberStrike** - December 13, 2017, 5:08 pm

Very well written and informative, hope to see more from you in the future

**DarkWapOne** - August 18, 2018, 3:46 am

Very informative thank you for sharing :)