SSTI Identification

We can detect SSTI vulnerabilities by injecting different tags in the inputs we control to see if they are evaluated in the response. We don't necessarily need to see the injected data reflected in the response we receive. Sometimes it is just evaluated on different pages (blind).

The easiest way to detect injections is to supply mathematical expressions in curly brackets, for example:

Code: html

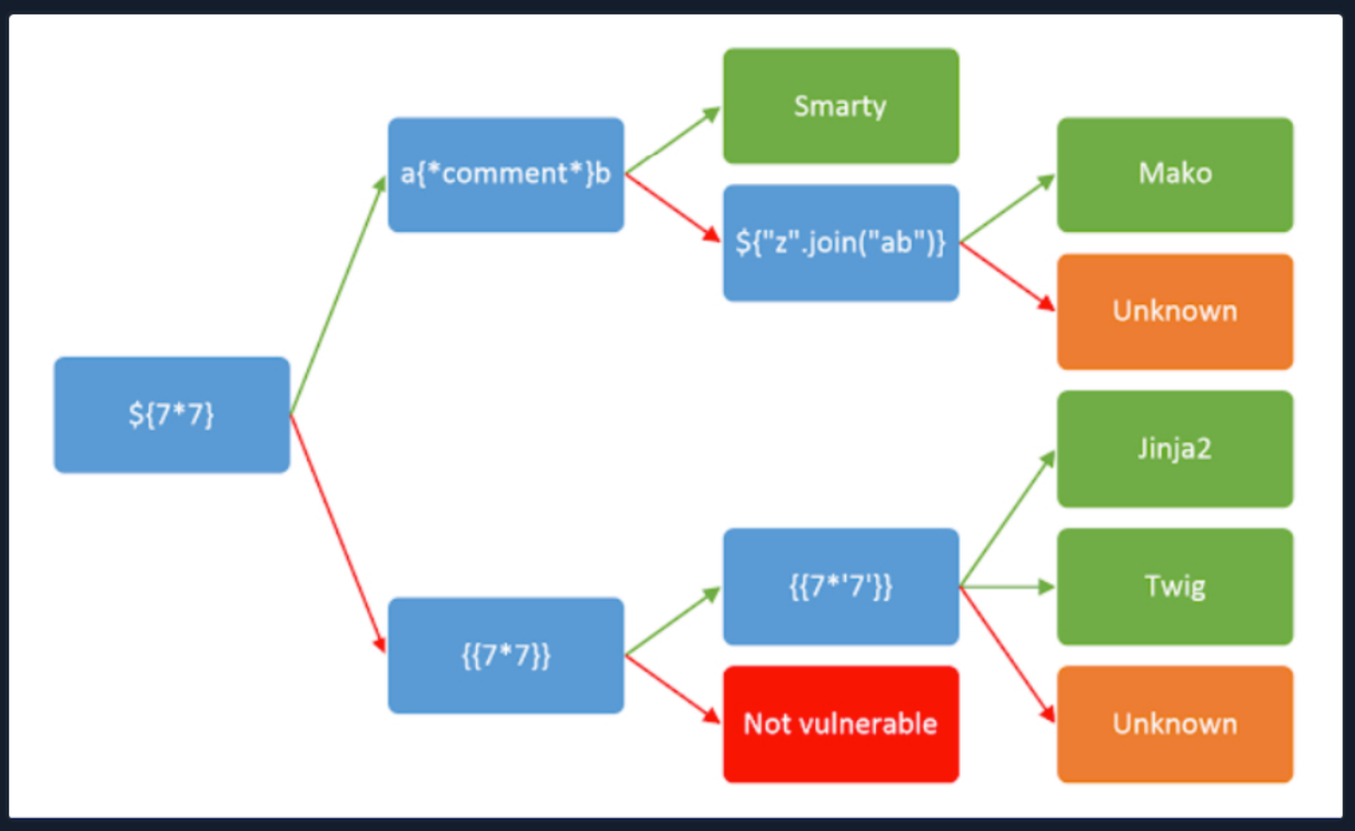
```
{7*7}
${7*7}
#{7*7}
%{7*7}
{{7*7}}
...
```

We will look for "49" in the response when injecting these payloads to identify that server-side evaluation occurred.

The most difficult way to identify SSTI is to fuzz the template by injecting combinations of special characters used in template expressions. These characters include `${{<%[%'"}}%\`. If an exception is caused, this means that we have some control over what the server interprets in terms of template expressions.

We can use tools such as Tplmap or J2EE Scan (Burp Pro) to automatically test for SSTI vulnerabilities or create a payload list to use with Burp Intruder or ZAP.

The diagram below from PortsSwigger can help us identify if we are dealing with an SSTI vulnerability and also identify the underlying template engine.



In addition to the above diagram, we can try the following approaches to recognize the technology we are dealing with:

- Check verbose errors for technology names. Sometimes just copying the error in Google search can provide us with a straight answer regarding the underlying technology used
- Check for extensions. For example, .jsp extensions are associated with Java. When dealing with Java, we may be facing an expression language/OGNL injection vulnerability instead of traditional SSTI
- Send expressions with unclosed curly brackets to see if verbose errors are generated. Do not try this approach on production systems, as you may crash the webserver.

Table of Contents

Introduction to Server-Side Attacks

Abusing Intermediary Applications

AJP Proxy

Nginx Reverse Proxy & AJP

Apache Reverse Proxy & AJP

Server-Side Request Forgery (SSRF)

Server-Side Request Forgery (SSRF) Overview

SSRF Exploitation Example

Blind SSRF

Blind SSRF Exploitation Example

Time-Based SSRF

Server-Side Includes (SSI) Injection

Server-Side Includes Overview

SSI Injection Exploitation Example

Edge-Side Includes (ESI) Injection

Edge-Side Includes (ESI)

Server-Side Template Injections

Introduction to Template Engines

SSTI Identification

SSTI Exploitation Example 1

SSTI Exploitation Example 2

SSTI Exploitation Example 3

Extensible Stylesheet Language Transformations Server-Side Injections

Attacking XSLT

Skills Assessment

Server-Side Attacks - Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left