SSTI

BY UNCLE RAT

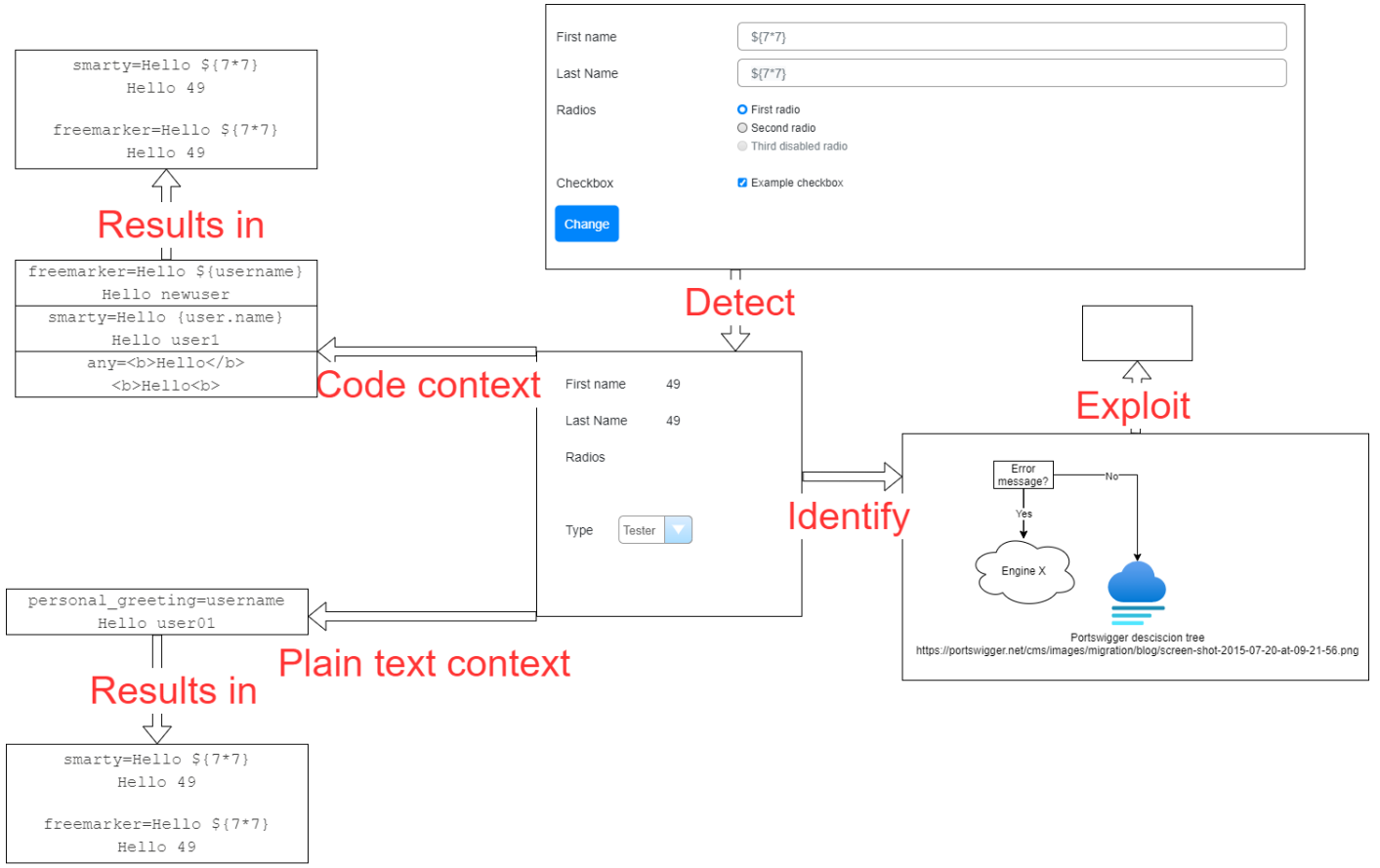
Agenda

- What is SSTI?
- Why SSTI?
- Different SSTI Contexts
- Attack strategy
  - Detection
  - Identification of the templating engine used
  - Exploiting SSTI

What is SSTI?

What is SSTI?

- Sometimes templating engines are being used
  - Twig - https://twig.symfony.com/
  - Tornado - https://ajinabraham.com/blog/server-side-template-injection-in-tornado
- Developers can sometimes feed unsanitised data to engines
- This allows for arbitrary code execution


smarty=Hello ${7*7}
Hello 49
freemarker=Hello ${7*7}
Hello 49
Results in
freemarker=Hello ${username}
Hello newuser
smarty=Hello {user.name}
Hello user1
any=Hello
Hello
Code context
First name
Last Name
Radios
First radio
Second radio
Third disabled radio
Checkbox
Example checkbox
Change
Detect
First name 49
Last Name 49
Radios
Type Tester
Identify
Exploit
Error message?
No
Yes
Engine X
Portswigger desciscion tree
https://portswigger.net/cms/images/migration/blog/screen-shot-2015-07-20-at-09-21-56.png
personal_greeting=username
Hello user01
Plain text context
Results in
smarty=Hello ${7*7}
Hello 49
freemarker=Hello ${7*7}
Hello 49

Why SSTI?

Why SSTI?

- SSTI is very hard to detect
  - Often mistaken for XSS
  - Often missed fully
- Can easily lead to Remote Code Execution

Identify the templating engine

Identify the templating engine

- Sometimes templating engines give an error when you fuzz them
- If the don't follow the flowchart layed out by albinowax over at
  - https://portswigger.net/research/server-side-template-injection

Attack strategy - Detection

Attack strategy - Detection

- XSS similarities
- Insert attack vector into every field possible
  - ${{7*7}}
- If it resolves we might have an SSTI or CSTI
  - Now we identify the templating engine

Exploiting SSTI

Exploiting SSTI

- READ THE MANUAL
- Key areas of interest are:
  - 'For Template Authors' sections covering basic syntax.
  - 'Security Considerations' - chances are whoever developed the app you're testing didn't read this, and it may contain some useful hints.
  - Lists of builtin methods, functions, filters, and variables.
  - Lists of extensions/plugins - some may be enabled by default.

Exploiting SSTI

- Smarty: `{php}echo `id`;{/php}`
  - `Very simple executing id command on target`
- `Mako:` <%
  import os
  x=os.popen('id').read()
  %>
  ${x}
- Do you research into other templating engines and what kind of PoC you need for those