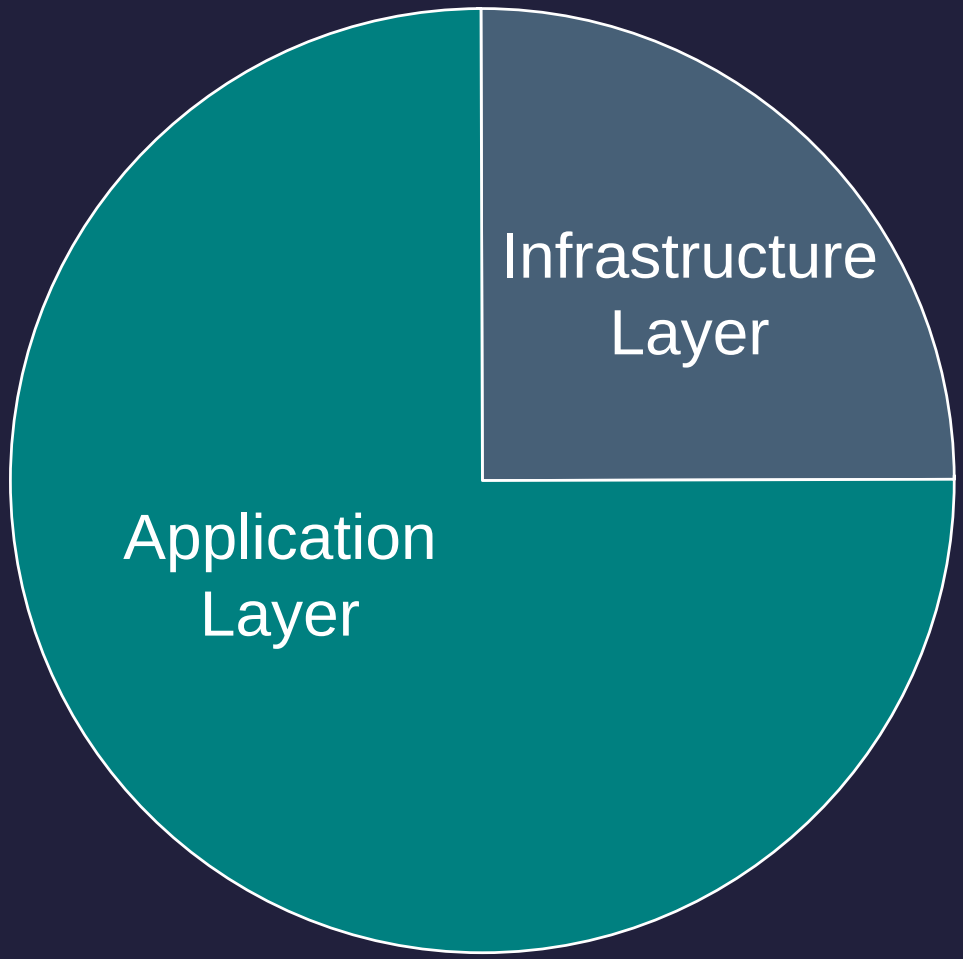
Refactoring for Security

Toan Huynh
Department of Electrical and Computer Engineering
University of Alberta
huynh@ece.ualberta.ca

Cyberattacks on Web-based Systems



The Problem

```
...
if (input contains valid chars) then
  process input
end if
...
```

The Problem (cont.)

...

process input

...

The Problem (cont.)

- Secure coding practices are not being used which leads to
  - Buffer overflow exploits
  - SQL injection
  - Cross-site scripting
  - ...

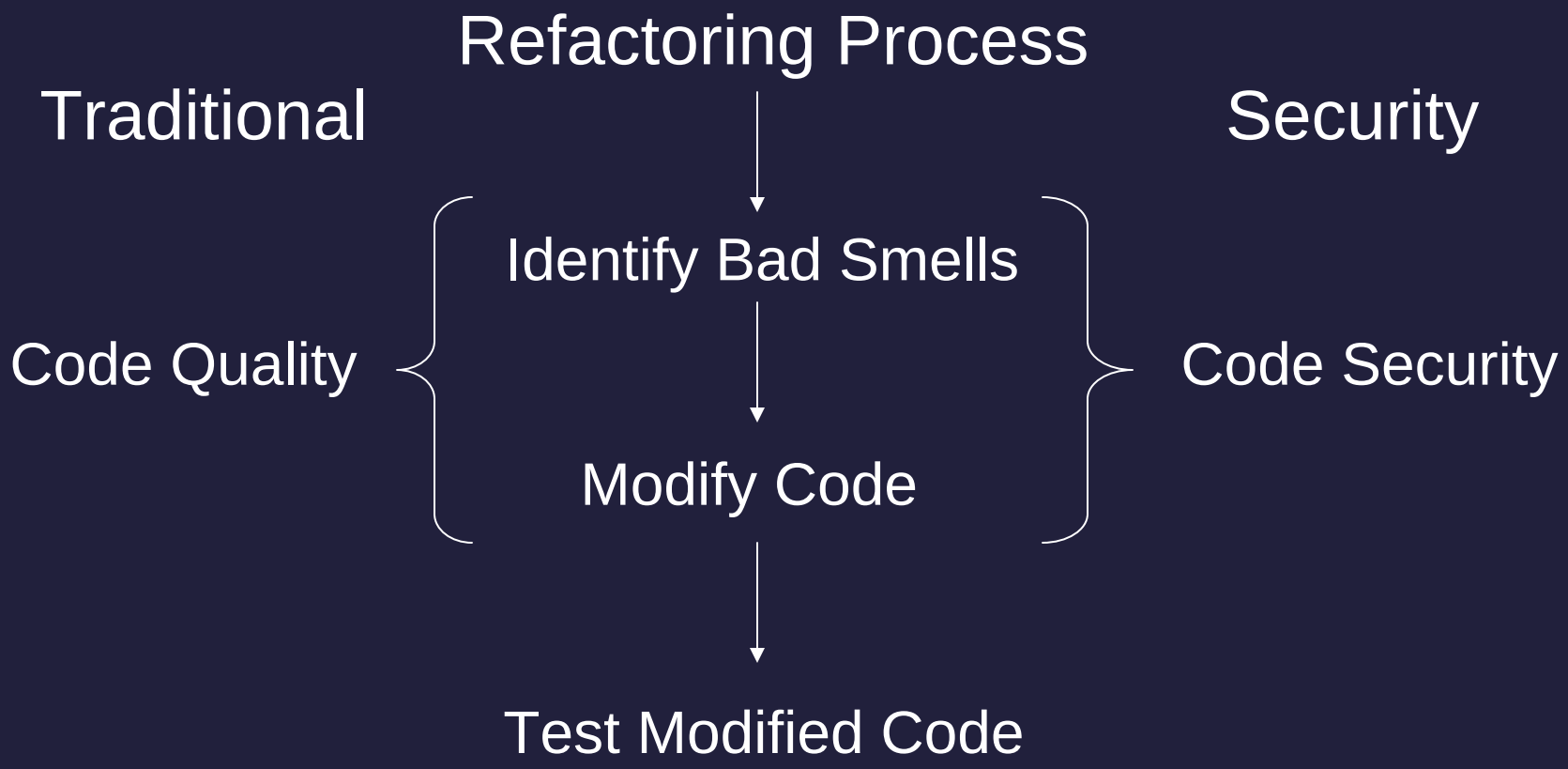
2-Step Solution

Coding

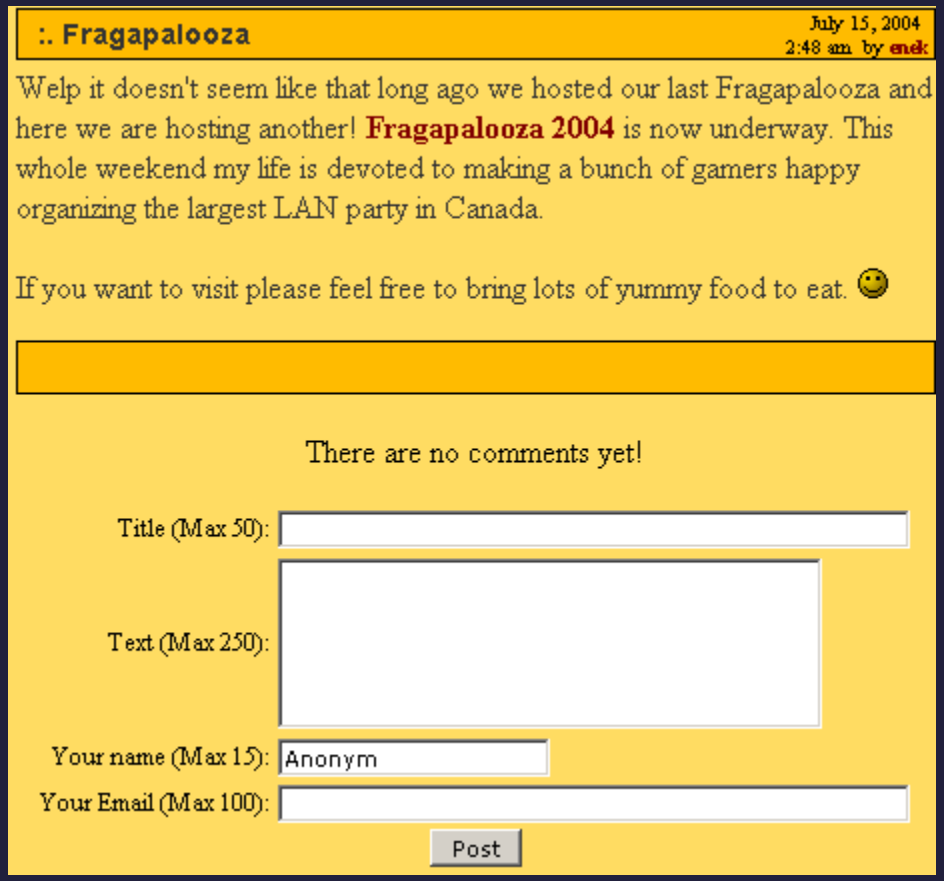
Programming

Refactor for Security

Refactoring: Traditional vs Security

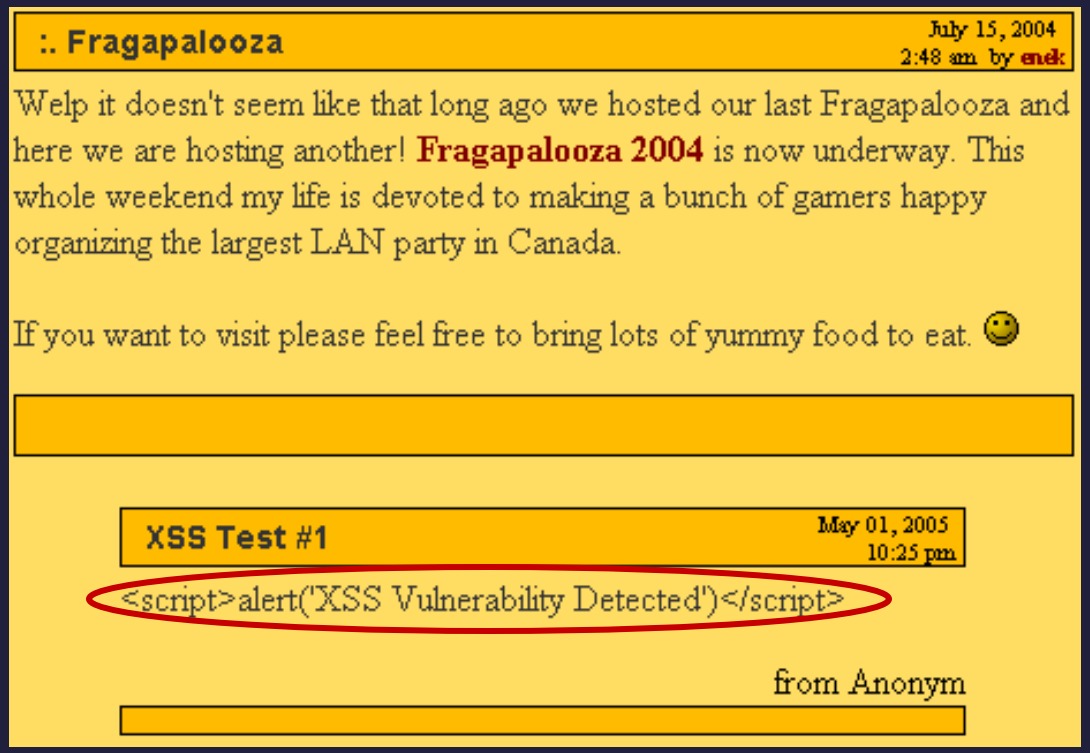


Example : Cross-site Vulnerability

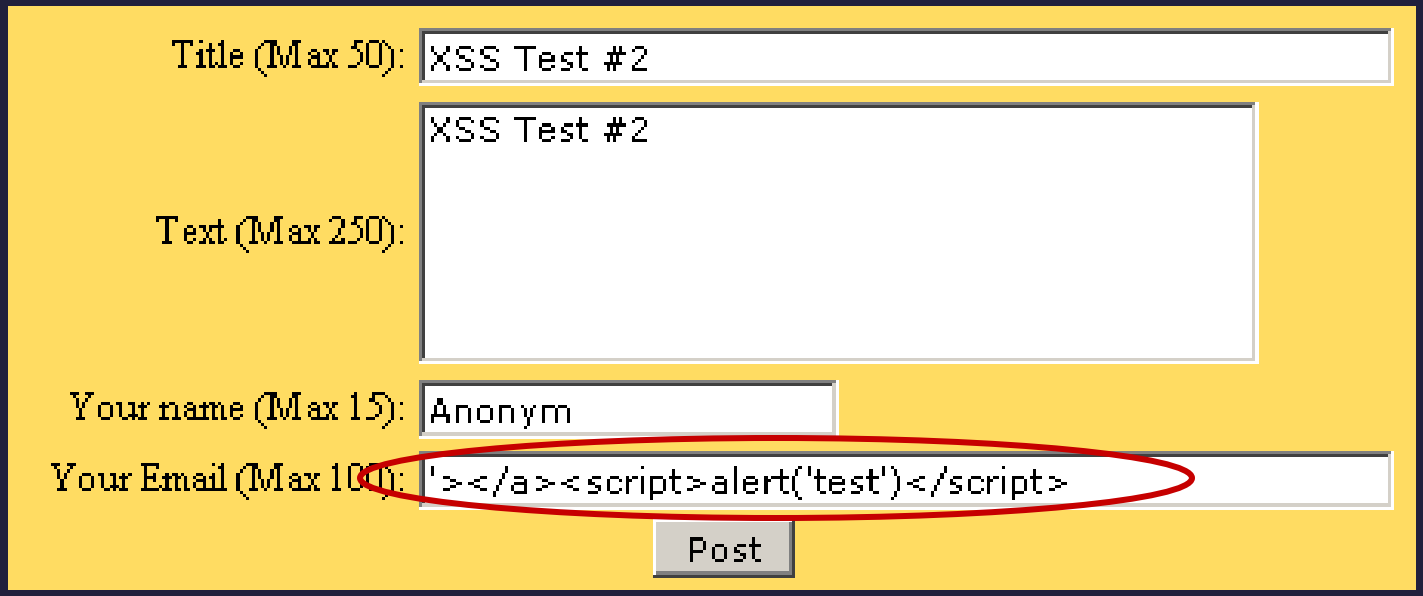


Example (cont.)

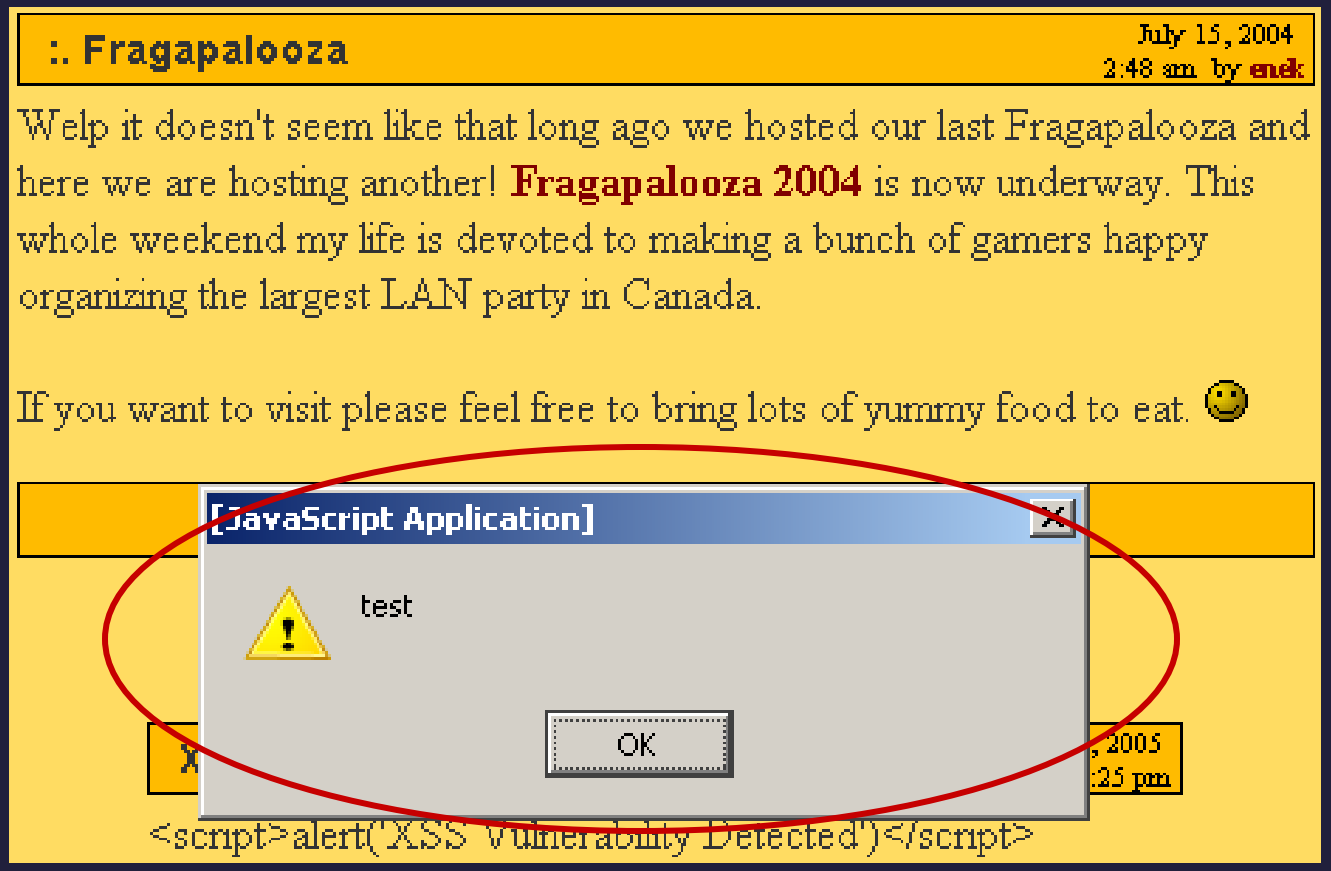
Example (cont.)



Example (cont.)



Example (cont.)



Filter All Outputs

- All outputs should be validated before they are to be displayed.

Filter All Outputs (cont.)

```
<?php
    …
    $message = htmlspecialchars($row[‘message’]);
    print($message’);
    …
    print($row[‘email’]);
    …
?>
```

Filter All Outputs (cont.)

```
<?php
    …
    $message = htmlspecialchars($row[‘message’]);
    print($message’);
    …
    $email = htmlspecialchars($row[‘email’]);
    print($email);
    …
?>
```

Summary

- Two step process will allow more secure applications to be developed
  - More security refactors will be introduced
  - A tool is under development