```
}
}
?>
/ ------------------------------General Method-----------------------------------------
private function validator_breed($value)
{
$breed_file = simplexml:load_file("breeds.xml");
$xmlText = $breed_file->asXML();

if(stristr($xmlText, $value) === FALSE)
{
return FALSE;
}
else
{
return TRUE;
}
}
}
?>
```

You can now make a slight change to the code line that checks the validity of the `dog_breed` value (see the **bold** highlighted line in Example 4-9).

```
((ctype_alpha($value)) && ($this->validator_breed($value) === TRUE) && strlen($value) <= 35)
? $this->dog_breed = $value : $error_message = FALSE;
```

The statement passes `$value` into `validator_breed ($this->validator_breed($value))`. If a `TRUE` is returned then that part of the `if` statement would be true. Otherwise it would be false. If the complete `if` statement is `TRUE` then the `dog_breed` property is set to the breed found in the XML file (`$this->dog_breed = $value`). If it is false, then `FALSE` is passed into the `$error_message` property (`$error_message = FALSE`).

> *Programming note—When you call a function in the same object, you must use the `$this` pointer (`$this->validator_breed($value) === TRUE`).*

You have now completed a program that much more secure. Users cannot enter any invalid information. The only field they can attempt to do so is in the `dog_name` field. However, even if they try to enter program code, or other code, the server-side program will strip out the special characters so the code becomes harmless. If a packet sniffing program tries to change the data before it is received by the server-side program, the validation and/or filtering methods will either cause the data to be rejected or will, again, make the data harmless.

This program is efficient as it tries to validate the information before it is sent to the server. It reduces the amount of communication back and forth between the server and the user. The completed program has two tiers (interface, business rules) and can be expanded to include a third tier (data) without major changes to the current code.

> *Programming recommendation—For small web applications that are used infrequently it is not necessary to break the program apart, as is done in this example. However, as mentioned, if it is an application that might expand in the future, or might gain significant users in the future, the program should initially be created with this in mind, by breaking the program into tiers.*