Lab 2.3 - Command Injection Vulnerabilities - Grepper

💀 Command injection allows a user to enter, prepend or append an executable command to unsanitized input. It happens all the time.

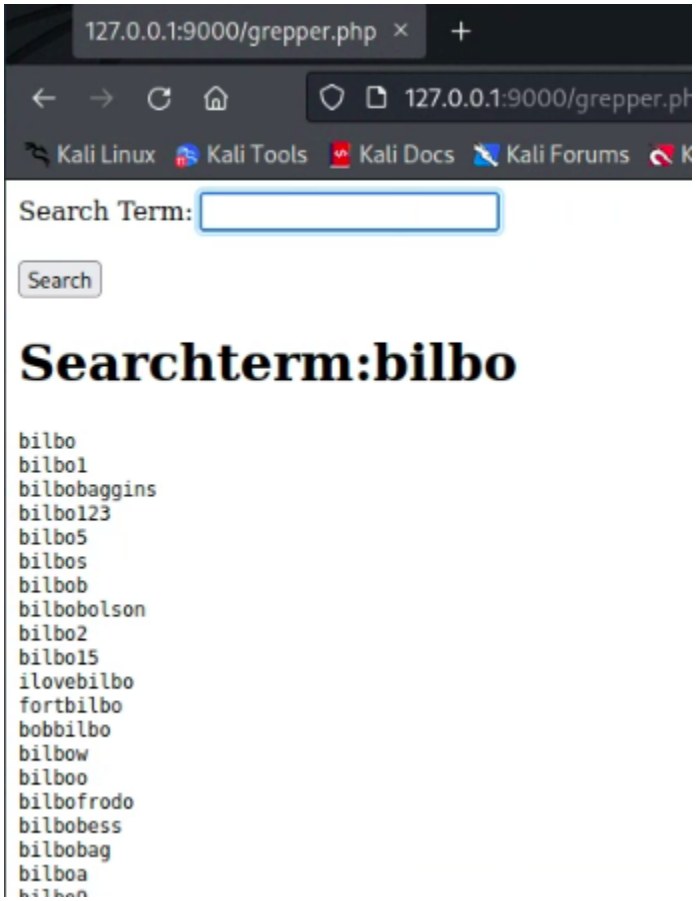
grepper.php

Create a new php file called grepper.php with the following contents. Adjust the source code as necessary to point to your version of rockyou.txt. This application will allow you to search it for dictionary items.

```
<form id="logform" method="post">
<div>Search Term: <input type="text" name="search"><div>
</select>
<div class="full-width"></br>
  <button type="submit">Search</button>
</div>
</form>
<?php
if(isset($_POST['search'])) {
  $searchterm=$_POST['search'];
  echo "<div>";
  echo "<h1>Searchterm:" . $searchterm . "</h1>";
  echo "</div>";

  echo "<pre>";
  passthru("cat /usr/share/wordlists/rockyou.txt | grep " . $searchterm);
  echo "</pre>";
}
?>
```

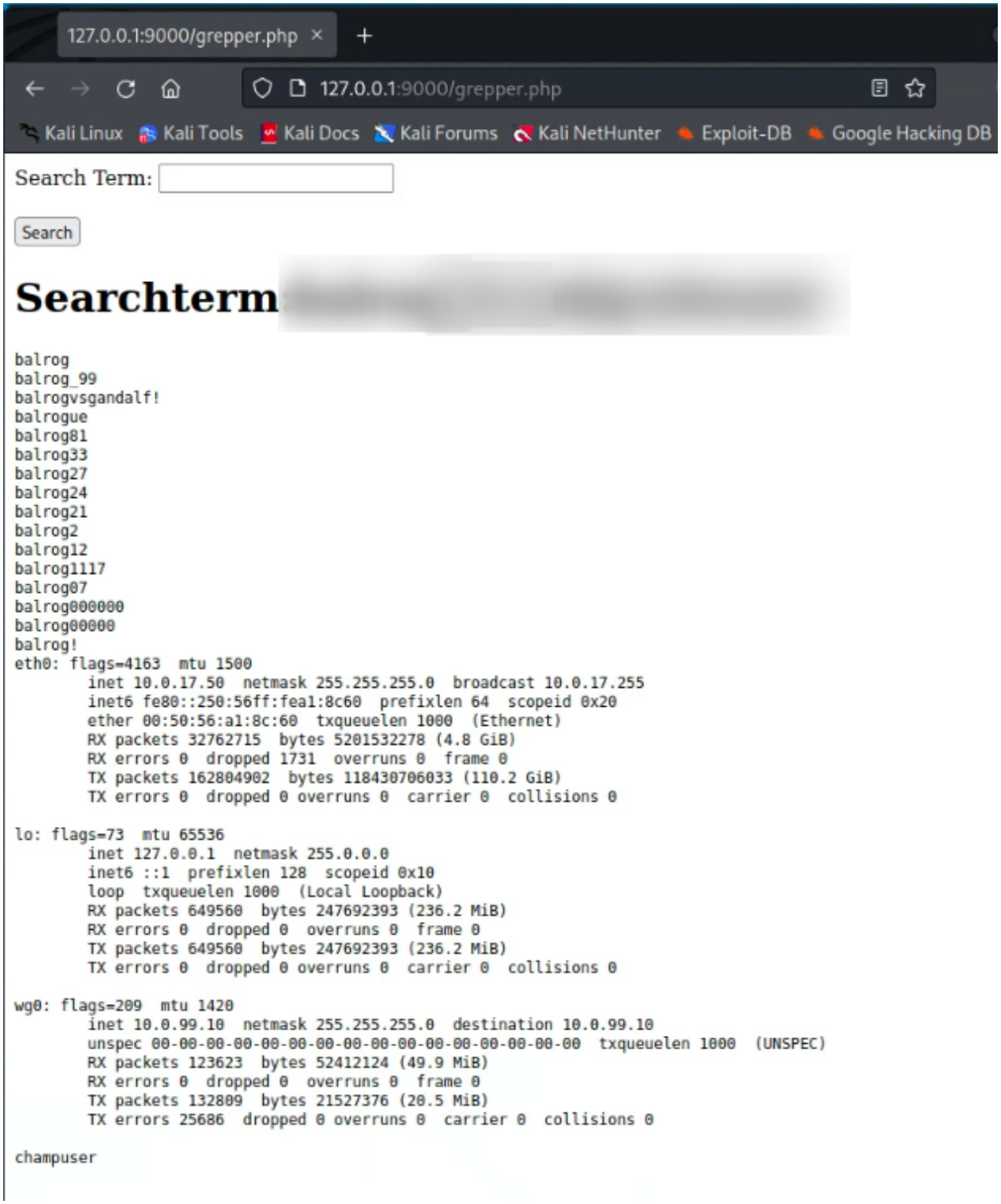
`Deliverable 1.  Try the application out and search for a string of interest.`



Note that the http parameters are not visible in the url, this is because we are using POST and form data to make this happen.

💡Though semicolon usage in prose may be underappreciated and perhaps baffling to you, it is critical in command injection. The ";" is in essence a command separator. By including it in the right place in an vulnerable application you can inject a series of your own commands.

`Deliverable 2.  Figure out how to run commands of your choosing. Provide a screenshot similar to the one below that shows your application output as well as commands you've snuck in.`



Challenge, see if you can use this technique to invoke a reverse shell, you can catch it on another local port, alternatively you can work with a partner to exploit their version of grepper

and invoke a shell on the remote system.

Deliverable 3. Write a technical article on command injection. If you remember sec335, shellshock was an example of this type of vulnerability.