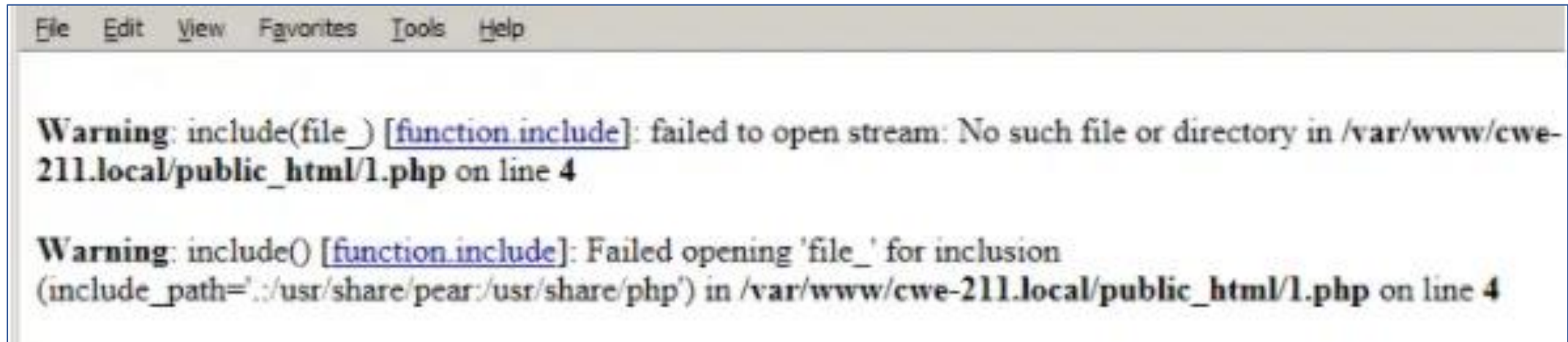


AZ BİLGİ

Hata Mesajları

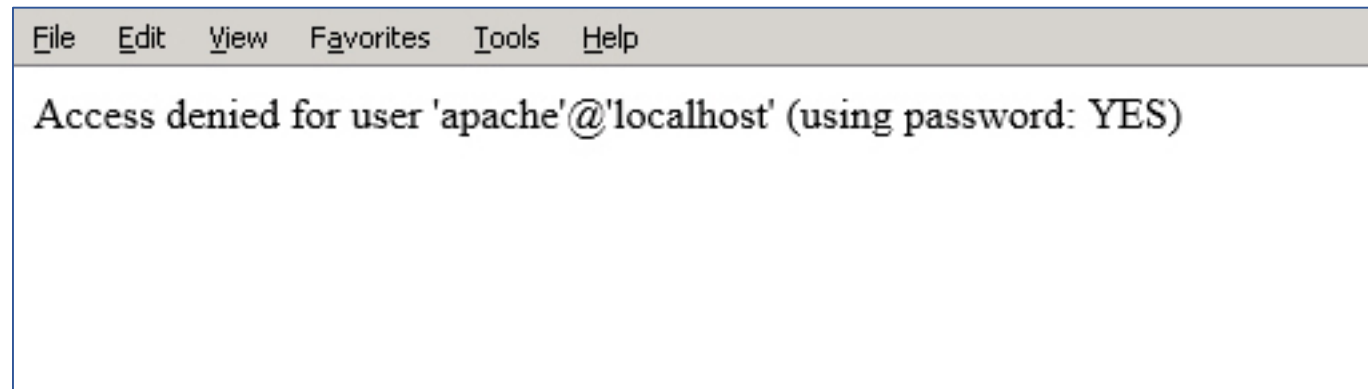


A screenshot of a web browser window with a menu bar containing 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The main content area displays two PHP warning messages. The first message states: 'Warning: include(file_) [function.include]: failed to open stream: No such file or directory in /var/www/cwe-211.local/public_html/1.php on line 4'. The second message states: 'Warning: include() [function.include]: Failed opening \'file_\' for inclusion (include_path=.:usr/share/pear:usr/share/php) in /var/www/cwe-211.local/public_html/1.php on line 4'.

```
File Edit View Favorites Tools Help

Warning: include(file_) [function.include]: failed to open stream: No such file or directory in /var/www/cwe-211.local/public_html/1.php on line 4

Warning: include() [function.include]: Failed opening 'file_' for inclusion
(include_path=.:usr/share/pear:usr/share/php) in /var/www/cwe-211.local/public_html/1.php on line 4
```

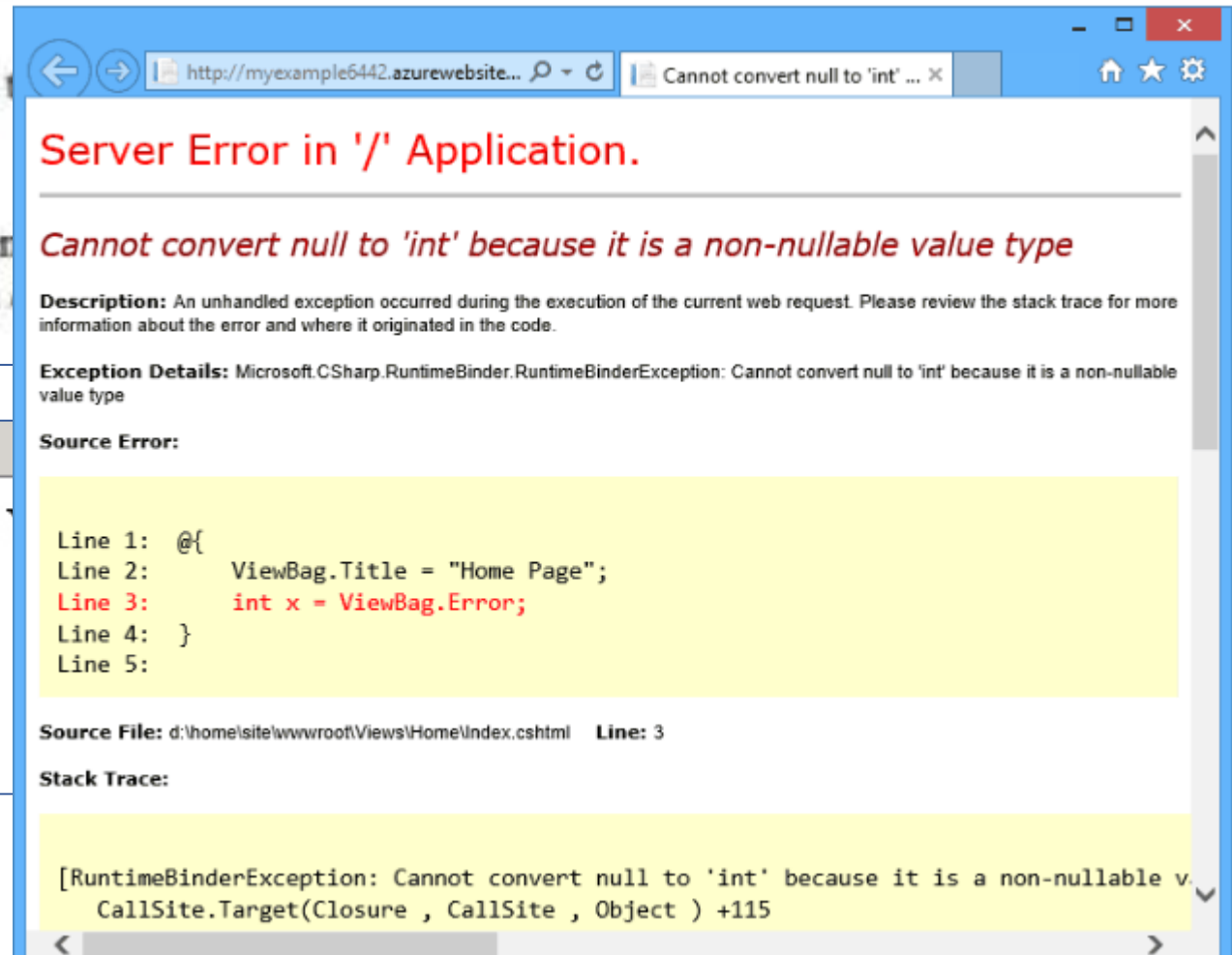
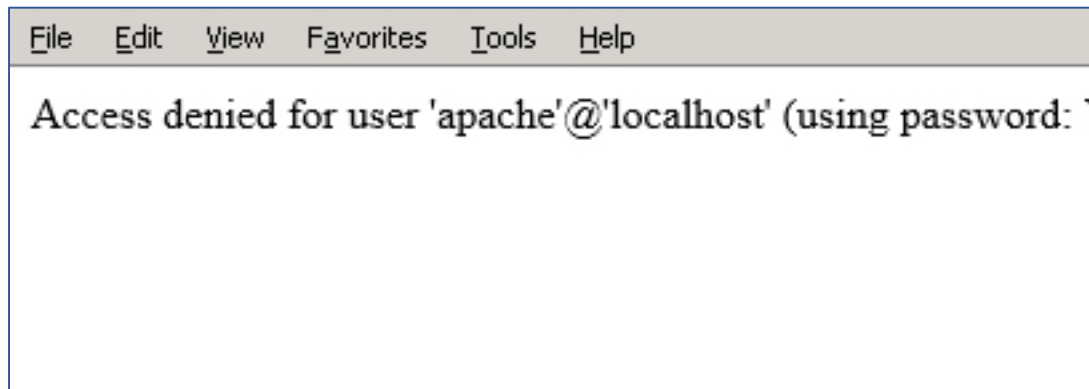
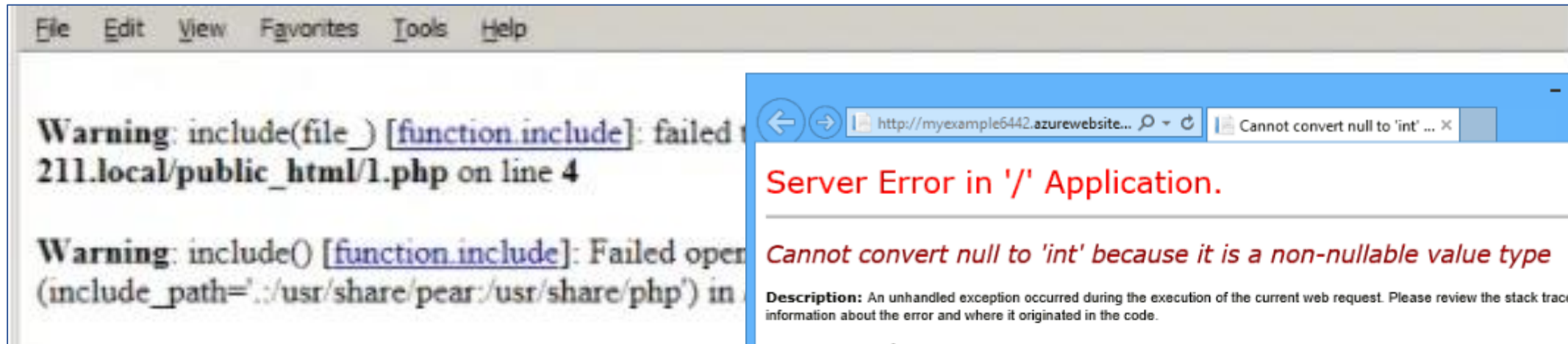


A screenshot of a web browser window with a menu bar containing 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The main content area displays a single message: 'Access denied for user 'apache'@'localhost' (using password: YES)'.


```
File Edit View Favorites Tools Help

Access denied for user 'apache'@'localhost' (using password: YES)
```

Hata Mesajları



Hata Mesajları

A screenshot of a web browser window with a menu bar (File, Edit, View, Favorites, Tools, Help). The main content area displays two PHP warning messages. The first message is: "Warning: include(file_) [function.include]: failed to open stream: No such file or directory in 211.local/public_html/1.php on line 4". The second message is: "Warning: include() [function.include]: Failed opening include_path='.:usr/share/pear:usr/share/php' for inclusion (include_path='.:usr/share/pear:usr/share/php') in 211.local/public_html/1.php on line 4".

```
Warning: include(file_) [function.include]: failed  
211.local/public_html/1.php on line 4  
  
Warning: include() [function.include]: Failed ope  
(include_path='.:usr/share/pear:usr/share/php') in
```

PHP

Set the option "display_errors=0" in php.ini file or use PHP function `error_reporting(0)` to disable output of error messages.

ASP

Use the following configuration in web.config.

```
<configuration>  
  <system.web>  
    <customErrors mode="On" defaultRedirect="error.htm" />  
  </system.web>  
</configuration>
```

Ruby

To disable error output run the server with the "-e production" option. Another way is to set "config.consider_all_requests_local=false" in application environment.

NodeJS

To disable verbose debugging messages set the NODE_ENV to production. This disallows stack traces from being returned upon error conditions.

```
set NODE_ENV=production
```

Hata Mesajları

1. General recommendations

- Never run applications in debug mode on production servers,
- Use least privilege approach and proper file system permissions,
- Run remotely accessible applications in chroot/sandbox environments,
- Never store log files with world-readable permissions;

2. Web site environment

- Disable directory listing to prevent exposure of web site structure and potentially sensitive files,
- Use custom error pages that prevent from displaying excessive system information,
- Disable error reporting output into the client's browser,
- Use IP-based approach to access sensitive directories or restrict access to directories otherwise.

Böyle
Bir Sayfa
Bulunmamaktadır.

Bunun Nedeni: Adres Satırına Yanlış Yazılmış
ya da
Yayınına Son Verilmiş Olabilir.

[Ana Sayfaya Dönmek için Tıklayınız. ►](#)



KOD

- Minification
 - Minify
 - Uglify
- Obfuscation

Comments, spaces, and other extra characters are removed so that the .js file can load and execute more quickly. JavaScript and CSS minification both contribute to a faster-loading website, and can result in better user engagement and increased organic traffic.

Minify

```
1 // This function takes in name as a parameter
2 // and logs a string which greets that name
3 // using the information passed
4 function sayHi(name) {
5     console.log("Hi" + name + ", nice to meet you.")
6 }
7
8 sayHi("Sam");
```

```
1 function sayHi(o){console.log ("Hi"+o+", nice to meet you.") }sayHi("Sam");
```

```
/* paragraph styling here */
```

```
p {
font-family: arial;
color: green;
background-color: white;
}
```

```
/* links */
```

```
a:link {
color: blue;
}
```

```
a:visited {
color: white;
}
```

```
p{font-family:arial;color:green;background-
color:white;}a:link{color:blue;}a:visited{color:white;}
```

Obfuscation

obscurity isn't security

```
1  (function (window) => {  
2    var canvas = window.document.getElementById('canvas');  
3    if (canvas.getContext) {  
4      var ctx = canvas.getContext('2d');  
5  
6      ctx.fillRect(25, 25, 100, 100);  
7      ctx.clearRect(45, 45, 60, 60);  
8      ctx.strokeRect(50, 50, 50, 50);  
9    }  
10 })(window)
```


Original

Obfuscated

<https://jscrambler.com/blog/javascript-obfuscation-the-definitive-guide>

Obfuscation

obscurity isn't security



```
function getRecommendations(products, numRecommendations) {  
  const weights = [];  
  for (let i = 0; i < products.length; i++) {  
    const product = products[i];  
    const weight = getWeight(product);  
    weights.push({_id: product.id, weight});  
  }  
  
  weights.sort((recommendation1, recommendation2) =>  
    recommendation2.weight - recommendation1.weight  
  );  
  
  return weights.slice(0, numRecommendations);  
}
```

e-definitive-guide

Obfuscation

obscurity isn't security

```
function getRecommendations(products, numRecommendations) {  
  const weights = [];  
  for (let i = 0; i < products.length; i++) {  
    const product = products[i];  
    const weight = getWeight(product);  
    weights.push({_id: product.id, weight});  
  }  
  
  weights.sort((recommendation1, recommendation2) =>  
    reco  
  );  
  
  return  
}
```

```
function getRecommendations(products, purchasedProducts, numRecommendations){var weights=  
[];for(var i=0;i < products.length;i++){var product=products[i];var  
weight=getWeight(product);weights.push({id:product.id,weight});}weights.sort((recommendation  
1,recommendation2)=>recommendation2.weight - recommendation1.weight);return  
weights.slice(0,Math.min(numRecommendations,weights.length));}
```

Obfuscation

```
function getRecommendations(products, numRecommendations) {
  const weights = [];
  for (let i = 0; i < products.length; i++) {
    const product = products[i];
    const weight = getWeight(product);
    weights.push({_id: product.id, weight});
  }

  weights.sort((recommendation1, recommendation2) =>
    recommendation2.weight - recommendation1.weight
  );

  return weights.slice(0, numRecommendations);
}
```

```
// 1371 more lines of code
function getRecommendations(02, F5) {
  var C7 = M7wd_;
  var s2 = [arguments];
  s2[6] = C7.U1()[0][5];
  C7.P1();
  for (; s2[6] !== C7.q8()[26][3];) {
    switch (s2[6]) {
      case C7.U1()[3][6]:
        s2[9][C7.F(4)]((function () {
          C7.P1();
          var z2 = [arguments];
          z2[6] = C7.q8()[25][8];
          for (; z2[6] !== C7.U1()[1][7];) {
            switch (z2[6]) {
              case C7.U1()[31][14]:
                z2[9] = {};
                z2[9][C7.F(3)] = s2[4][C7.F(6)];
                z2[9][C7.F(1)] = s2[3];
                return z2[9];
                break;
            }
          })[C7.F(0)](this, arguments));
        s2[6] = C7.U1()[24][32];
        break;
      case C7.U1()[15][6][26]:
        s2[6] = s2[5] < s2[0][0][C7.w(7)] ? C7.q8()[7][4] : C7.q8()[6][26];
        break;
      case C7.q8()[3][16]:
        s2[5] = 0;
        s2[6] = C7.q8()[20][11];
        break;
      case C7.U1()[22][8]:
        s2[5]++;
        s2[6] = C7.q8()[17][5][23];
        break;
      case C7.q8()[18][23]:
        s2[9] = [];
        s2[6] = C7.U1()[28][19];
        break;
      case C7.U1()[8][16]:
        s2[4] = s2[0][0][s2[5]];
        s2[3] = V0oXS$(s2[4]);
        s2[6] = C7.q8()[15][18];
        break;
      case C7.U1()[21][8]:}
    }
  }
}
```

Obfuscation

But concealing data is just one of several dimensions of JS obfuscation. Strong obfuscation will also **obfuscate the layout and program control flow**, as well as include several **optimization** techniques. Typically, it will target:

- Identifiers;
 - Booleans;
 - Functions;
 - Numbers;
 - Predicates;
 - Regular expressions;
 - Statements; and
 - Program control flow.
- hides constants and names;
 - makes it difficult to understand the order in which the code is executed;
 - makes it difficult to understand what the relevant code is;
 - increases overall program size and introduces new classes and methods;
 - introduces new predicates and rewrites the conditional and looping constructs;
 - increases long-range variable dependencies.