CVE检索优化

首先对`cvemap.json`进行筛选

通过查看`cvemap.json`可以发现一个特征，每个CVE号下都存在`vulnerable_cpe`以及`cve_description`字段

- `vulnerable_cpe`字段里明确列出了受影响的 CPE 模式（包括版本范围），例如：

```
"vulnerable_cpe": [
  "cpe:2.3:a:djangoproject:django:*:*:*:*:*:*:*:*",
  "cpe:2.3:a:djangoproject:django:5.1:*:*:*:*:*:*:*"
]
```

- `cve_description` 中的版本区间描述，这时要注意提取“before X.Y.Z”、“and X.Y.Z 之前”这样的关键词，并做严格的语义解析，避免漏判

筛选逻辑

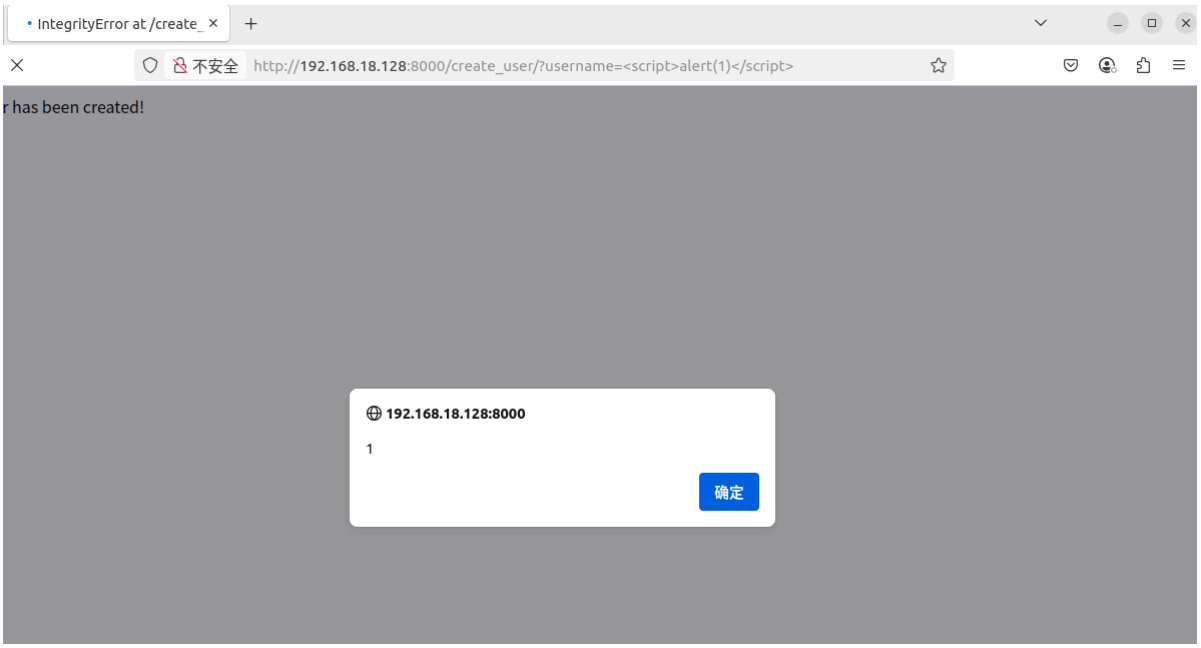
直接对比目标版本（如 `5.1.4`）与这些 CPE 模式，可以更精准地判断该 CVE 是否影响当前版本，而不必依赖描述文本的自然语言匹配。

如果某些 CVE 项没有提供明确的 CPE 列表，或者版本匹配逻辑出现歧义，再回退到解析`cve_description` 中的版本区间描述

补充检索

- 首次从 CVEMAP 获取：用 `cvemap_product(keyword, output_dir)` 拿到所有与 “django” 相关的 CVE。
- 版本匹配过滤：基于 CPE 字段和 / 或描述字段，筛掉那些未影响目标版本的 CVE，得到`filtered_cves`。
- 若 `filtered_cves` 为空或预期 CVE 不在其中：用 Google（或其他漏洞数据库 API，如 NVD、CIRCL）做初次检索：`"<keyword> <version> vuln"` 或 `"<keyword> <version> CVE"`，以获取遗漏的 CVE ID。
- 二次补充：把从 Google 搜索结果里提取到的新 CVE，重新做版本匹配并合并到主列表。

扫描阶段已执行漏洞利用


• IntegrityError at /create_
不安全
http://192.168.18.128:8000/create_user/?username=<script>alert(1)</script>
r has been created!
192.168.18.128:8000
1
确定