6月12日

Google在线搜索的两个阶段

第一阶段：通过{keyword} CVE list在线检索相关CVE

访问Google返回的前十条链接，过滤掉链接内的一些无关内容后提取出正文部分，交由模型判别是否对渗透任务直接有用，若判断有用则保存反之丢弃。

对上述内容构建索引，结合RAG查询引擎回答相关的CVE

> ```
> Starting query: For each document, list out ALL CVE numbers, urls, keywords and their applicable version that are relevant to exploit the vulnerabilities of Django 1.11.4.
>    You must analyze each document to extract the CVE number. You must list out ALL the CVE numbers!
>    You must respond **ONLY** with a valid JSON string, without any additional text or markdown formatting. The urls should link to a webpage that contains exploit implementation details, not to a file path.
>    For each field, include at most 5 most relevant items. If there are more than 5 items, you can include the most relevant 5 items.
>    {
>        "CVE": {"CVE-2023-42442": "<app> from 3.0.0 to 3.5.5(excluding) and from 3.6.0 to 3.6.4(excluding)",
>                "CVE-2023-46123": "<app> up to 3.8.0(excluding)",
>        },
>        "link": {"https://github.com/jumpserver/jumpserver/security/advisories/GHSA-2vvr-vmvx-73ch": "<app> from 3.0.0 to 3.10.6",
>        },
>        "keyword": {"Unauthorized access to session replay vulnerability": "<app> from 3.0.0 to 3.5.5(excluding) and from 3.6.0 to 3.6.4(excluding)",
>        }
>    }
> ```
>
> ```
> query keywords: ['4', 'relevant', 'app', 'cve', 'excluding', '0', '6', '3', 'must', '5']
> Extracted keywords: ['4', 'cve', '0', '6', '3', '5']
> ```

关键词提取逻辑是 `SimpleKeywordTableIndex` 的底层实现

```
Google search raw response: {"CVE": {"CVE-2019-14233": "1.11.x before 1.11.23", "CVE-2018-7536": "1.11.x before 1.11.23", "CVE-2019-14234": "1.11.x before 1.11.23", "CVE-2020-7471": "1.11.x before 1.11.23", "CVE-2019-19844": "1.11.x before 1.11.23"}, "link": {"https://www.meterian.io/components/live/python/django/1.11.4": "1.11.x before 1.11.23", "https://ryu22e.org/en/posts/2019/12/25/django-cve-2019-19844/": "1.11.x before 1.11.23", "https://github.com/ryu22e/django_cve_2019_19844_poc": "1.11.x before 1.11.23", "http://www.securityfocus.com/bid/100643": "1.11.x before 1.11.5", "http://www.securitytracker.com/id/1039264": "1.11.x before 1.11.5"}, "keyword": {"Django strip_tags HTMLParser vulnerability": "1.11.x before 1.11.23", "Django urlize function catastrophic backtracking": "1.11.x before 1.11.23", "Django JSONField HStoreField SQL injection": "1.11.x before 1.11.23", "Django StringAgg SQL injection": "1.11.x before 1.11.23", "Django password reset token Unicode case transformation": "1.11.x before 1.11.23"}}
```

第二阶段：通过{cve} exploit PoC在线检索与目标CVE相关的可执行漏洞，其余过程与第一阶段无异。

目前模型过滤规则完全一致，但是两个在线检索阶段的侧重点不同，使用同一套过滤规则会导致某阶段的准确率不同；应设计两套侧重点不一致的过滤法则供不同阶段使用，如CVE检索阶段可以着重描述中的影响版本是否包含目标版本，PoCK检索阶段应注重内容中是否包含`code(s)`、`plain instruction(s)`、`command line operation(s)`等可以指导执行渗透的相关信息