基于 go 语言的自动化情报搜集框架的实现

张友明 陈 科

成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】目前搜索引擎 hack 依旧是对安全极具威胁的攻击方式，通过搜索引擎能直接从网络中获取到受害者的网站后台或是网站的源代码，甚至是用户的个人隐私信息。为了尽可能的降低这种安全威胁，提高维护网络安全的效率，提出了基于 go 语言的自动化情报搜集框架，主要通过 bing 和 google 搜索引擎进行情报搜集。该框架利用了 go 语言的并发特性，从而极大的提高爬虫的性能，加快情报收集的速率。

【关键词】 爬虫技术；go 语言；情报收集；网络安全；搜索引擎

搜索引擎无论在何时都是一把强大的双刃剑。黑客攻击就是主要的威胁之一[1]。在不受限制的情况下，通过特殊的语法结构[2]就能精准的搜索到很多暴露在网络上的隐私信息。大多数时候，黑客总能通过这些不起眼的信息从而突破防线，攻击到内部并造成巨大的破坏。但是对于维护安全的人来说，一条一条的查找这些小情报并维护是一件工作量巨大的事情，所以了解和学习坏人的做法，才能更加高效的解决问题[3]。基于 go 语言的自动化情报搜集框架在一定程度上能够提高安全人员的工作效率以及降低情报搜集的工作量。

1 简介

1.1 功能框架

本文提出的基于 go 语言的自动化情报搜集框架包含的主要搜索引擎语法搜集情报、网页 JS 脚本爬取并分析、网页屏幕截屏、自定义搜索引擎语法，具体功能结构图如图 1 所示。

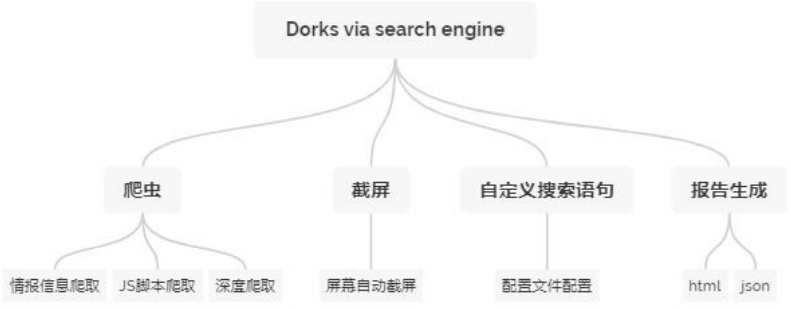

图 1 框架功能图

1.2 框架功能介绍

命令行界面：通过命令行进行功能选择和参数初始化，想比于漂亮的界面，命令行减少了不必要的界面渲染，可以直接在 shell 中运行，所以更加的方便，运行效率更加的高。

自定义搜索语句规则：通过命令行指定文件路径，框架自动对用户自定义的规则语句对程序进行初始化。考虑到用户输入的不可控性，所以采用了固定文件内容模板的方式，方便程序对进行文件内容解析。同时框架会存在默认的搜索引擎语法，方便一般的用户能够随时直接使用，而无需考虑搜索引擎语法的问题，在框架初始化的同时会从指定的地址拉去最新的搜索语法规则到本地完成语法的信息配置。

爬虫获取情报信息：获取用户指定的目标域或站点，调用该框架的搜索引擎爬虫接口进行信息检索，搜索引擎会对检索信息进行第一层的信息过滤和处理，该框架根据检索到的信息会再对检索到的信息进行第二次的处理和分析，最终提供给用户一个简洁的信息报告。

JS 脚本爬取：该框架默认情况下只爬取搜索引擎检索的到信息，若用户指定深入爬取，该框架会对检索的站点进行进一步探测，若目标为一个站点，由于 JS 常常会暴露很多关于站点的私密信息，所以程序会根据配置信息进行 JS 脚本爬取以方便供其他 JS 脚本分析程序进行分析。

网页屏幕截屏：通常单独的打开网页检索信息的效果并不高，所以在爬取搜索引擎检索的信息时会同时生成网页的截屏，方便直接查看检索信息的图片形式，也能提高查找和发现攻击面的效率，采用了 headless-chrome 技术结合 chromedp 框架进行网页的截图操作。

报告生成：根据框架检索到的信息和用户执行程序时选择的命令参数自动生成 JSON 或 HTML 文件格式报告，同时也提供不生成报告文件，直接将结果显示在屏幕上的功能。

1.3 核心技术简介

1.3.1 搜索引擎语法

本文中提及到的信息检索功能实现都是基于搜索引擎语法，主要使用的是搜索引擎的 bool 逻辑检索语法、特殊检索语法以及高级搜索语法，例如特殊检索语法的 intile 和 allintile，site，inurl，intext，高级搜索语法的 filetype，info 等常用检索语法[4]。通过将以上

的搜索引擎语法进行不同的组合，从而达到信息筛选的功能。例如利用 site 语法指定目标范围，可以是一个域名或者一个精准的站点，然后联合高级语法 initle 对被检索的信息的网页标题信息进行查询，inurl 对网页的链接类别进行查询，filetype 指定检索文件类型过滤，例如 .pdf、.sql、.xls、.doc 等可能存在重要信息的文件，经过搜索引擎语法的层层过滤筛选，获取到精准的目标情报信息。在本自动化情报搜集框架中默认的搜索引擎规则中也是使用了这样的搜索引擎语法框架，从而能够精准的获取目标的相关泄露信息。

1.3.2 网页屏幕截屏功能

本文中提到的网页屏幕截屏功能采用了现在开发中流行的无界面浏览器 headless-chrome，相比于界面浏览器它的加载速度更快。在框架实际的实现过程中使用了 chromedp 框架。该框架是一个更快、更简单的 Golang 库用于调用支持 Chrome DevTools 协议的浏览器。通过它的 Tasks 结构体设定任务，page 接口的 CaptureScreenshot() 函数进行屏幕截屏。通过两者的结合不仅大大降低了功能开发的难度，并且提高了截图功能的效率。

1.3.3 爬虫技术

在本框架中爬虫部分使用到的核心模块是 go 语言原生的 net/http 库，goquery 库等相关爬虫技术模块。其中 net/http 主要是用于模拟浏览器发送请求并接受响应数据的模块，goquery 库用来解析 net/http 库响应返回的 html 数据，通过以上两个库的协调工作完成基本的爬虫结果处理，程序再根据用户输入的命令调用两个库进行深层的 JS 脚本爬取，最后将结果保存在本地方便后续调用和处理。在爬虫技术模块中还是使用了 go 语言并发特性，让所有的爬取线程都能并发执行，尽可能的提高 CPU 的利用率。

1.3.4 报告自动生成技术

本框架最终生成报告为 JSON 和 HTML 两种形式，JSON 格式采用了 go 语言原生的 encoding/json 库，HTML 采用了文件读写配合 gohtml 库的方式，其中 gohtml 库能够通过 go 语言编程自动生成 html 代码。JSON 格式方便其他程序能够快速解析或调用，HTML 方便用户能够直观的查看情报搜集结果并进行数据分析。

2 设计思路及实现

2.1 功能流程

本框架是以命令行界面的形式存在，用户通过命令行输入参数和自定义的配置文件，框架将自动解析配置文件和用户输入的命令，然后根据命令通过搜索引擎进行情报搜集，无论最终的搜集情况如何，都会生成一个最终报告。其总体的功能运行流程如图 2 所示。

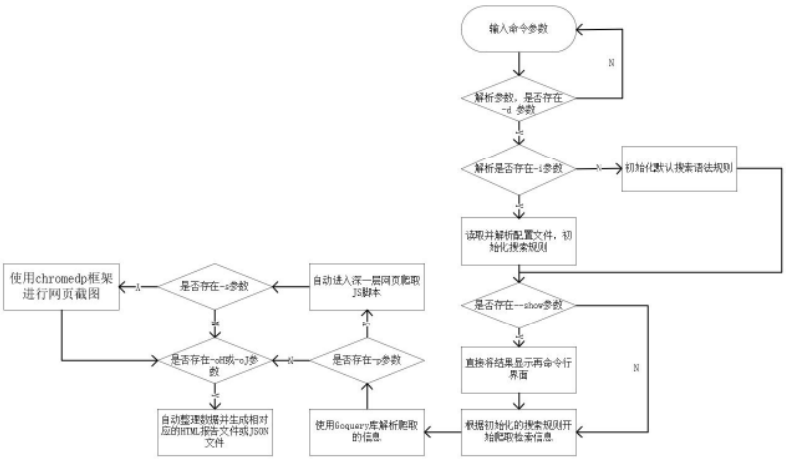

图 2 框架功能流程图

2.2 实现步骤

2.2.1 命令行界面及解析参数

该框架的命令行使用 cli 命令行框架实现，用户通过不同的参数完成操作。本框架中 -d 参数，即目标域名或网址为必选项，其中搜索引擎语法规则，用户可以使用默认的语法，也能通过参数 -i 指定自定义的语法规则，通过 go 语言的 yaml 解析库解析配置文件。它们的值的组合将会为后续操作提供基础参考。命令行功能主要实现步骤：

第一步：通过初始化 cli 命令行库中 cli 结构体中的 APP 结构体来声明 Flag 和 Command 两种命令行操作方式，Flag 即为全局操作指令，Command 为命令操作。在 APP 结构体中，Command 属性接受一个 []*cli.Command，Flag 接受 []cli.Flag，从而实现不同的命令行操作。

第二步：将 Flag 的值映射到 Options 结构体中，Options 为自定义的结构体，用于存储用户输入的命令参数值。

第三步：将这个初始化过程封装成 ParseOptions() 函数并以 *Options 作为返回参数，方便框架的主程序包调用。

2.2.2 情报信息检索及数据解析

本框架的情报检索过程会根据命令行解析的参数和用户提供的自定义语法规则或是程序默认的规则进行情报信息检索，并对信息进行处理。框架的信息检索步骤如下：

第一步：通过 net/http 库，模拟发送请求，报头主要包括搜索语法规则和用户指定的域名或站点拼接的 URL、User-Agent，header 等。

第二步：接受请求的响应数据。通过 goquery 库的 NewDocumentFromReader() 函数解析响应返回的 HTML 数据，并利用其 Find() 函数指定检索语法规则，过滤出不同的信息并归类。

第三步：如用户制定了 -p 即深层次爬取信息，程序则会自动进入第二步解析出来的 URL，并再次利用 goquery 库。通过 .js 后缀、http://、https://、<script> 标签组合检索规则并通过 goquery 库的 Find() 函数过滤出 JS 脚本的 URL 地址。

第四步：若用户还指定了 -s 参数即网页截图，通过 chromedp 框架的 Tasks 结构体配置任务；Navigate() 函数指定目标 URL，参数值来自初次爬虫解析的 URL；GetLayoutMetrics() 函数获取目标网页的高和宽；CaptureScreenshot() 函数获取被检索目标网页截图。

2.2.3 情报报告生成

该框架的报告主要以方便查看和方便其他程序调用解析为目的，所以采取了 HTML 格式和 JSON 格式。程序最终会根据用户的选择生成不同类型的报告。报告内容主要以情报信息为主，包括爬取到的 URL，JS 脚本 URL，网页截图。HTML 报告的生成由 gohtml 库动态生成，通过 gohtml 库的 Tag() 函数和过滤出来的数据结合动态生成 HTML 代码，再通过文件读写将生成的 html 代码写入文件中，最终生成完整的 HTML 报告。JSON 报告主要利用了 encoding/json 库，通过 go 语言的结构体定义 JSON 节点的结构，在通过 josn 库中的 Marshal() 函数进行编码，动态的将结果保存为 JSON 格式并自动生成 JSON 报告。

3 结束语

搜索引擎 HACK 在当下时代依旧是一个严重的安全问题，在搜索引擎的高效检索的帮助下，黑客能够在极短的时间获取到精准的情报信息。本文中提出的基于 go 语言的自动化情报搜集框架就是模仿了搜索引擎 HACK 的攻击方式，能自动化的进行情报检索，对于提升安全有一定的帮助，使用者能通过本框架进行自我检索和发现并即时采取安全措施进行补救。当然，该框架还有一定的不足之处，例如情报信息的内容识别、类型区分等还存在一定的缺陷。在未来的优化中会着重于解决信息的处理，增强情报信息的识别效率和准确率，提高该框架的信息爬取效率和信息的搜集准确率。

【参考文献】

[1] 董颖 . 大数据时代的网络黑客攻击与防范治理 [J]. 网络安全技术与应用 ,2021(05):68-70..

[2] 荆济学 , 张伟 . 浅谈如何利用 Google 高效搜索 [J]. 电脑知识与技术 ,2010,6(19):5186-5188.

[3] 陈海红 , 闫利华 . 黑客攻击技术研究 [J]. 赤峰学院学报 (自然科学版),2010,26(01):28-30.

[4] 董佳 . 常用搜索引擎搜索语法简析与比较 [J]. 科技风 ,2014(19):102.