信息收集与漏洞扫描

——构建高效网络安全检测体系

干文轩　刘彬
攀枝花学院　四川攀枝花　617000

摘　要：本文深入探讨了网络安全漏洞扫描系统中扫描核心功能模块的构成及其工作原理。文章详细分析了信息收集模块、漏洞模块以及报告生成模块的设计细节。系统采用模块化设计理念，确保了高度的灵活性和可扩展性，为网络安全检测提供了坚实的技术保障。研究揭示了扫描核心各模块间的协同机制，其成果对于增强网络安全防护能力具有显著价值，为网络安全领域的理论研究与实践应用提供了宝贵的参考依据。

关键词：网络安全；漏洞扫描系统；扫描核心功能模块；模块化设计

1. 引言

随着信息技术的飞速发展，网络安全问题日益凸显，黑客和病毒通常是通过系统中存在的漏洞对目标主机进行系统攻击的。漏洞扫描技术就是对系统中存在的漏洞和安全隐患进行检测，使用户了解系统的漏洞所在，进而采取相应的措施。漏洞扫描系统作为网络安全防护的重要组成部分，通过对目标系统进行全面的信息收集与漏洞检测，能够及时发现并修复潜在的安全隐患，从而有效防范网络攻击。本文重点研究网络安全漏洞扫描系统中的扫描核心功能模块的设计。通过深入分析信息收集模块、漏洞模块以及报告生成模块的设计细节，为网络安全检测提供有力的技术支撑。

2. 漏洞扫描核心设计

网络安全漏洞扫描系统的扫描核心模块由信息收集模块、漏洞模块（由漏洞扫描与漏洞验证组成）以及报告生成模块组成 (如图 1 所示)。信息收集模块负责搜集目标域名、端口、中间件、旁站、C 段、WHOIS、目录接口、泄露信息、系统指纹及资产测绘等数据，为后续分析打点。漏洞扫描模块作为核心，全面检测 Web 漏洞、弱口令、SQL 注入及中间件漏洞等，确保安全无死角。报告生成模块则系统化整理扫描结果，采用统一模板，提炼关键信息，利用可视化技术呈现，并支持多种格式导出，全面满足用户对漏洞扫描报告的需求。

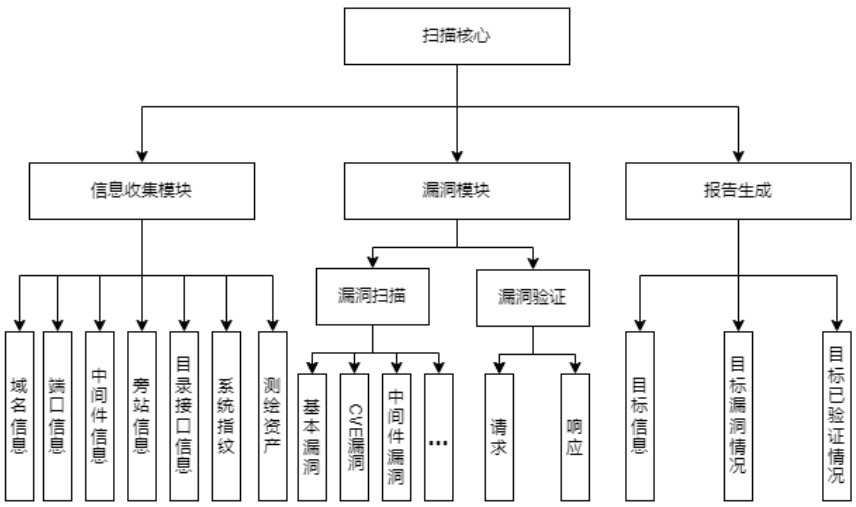

图 1 扫描核心模块设计图

3. 信息收集模块

3.1 域名信息

在域名信息收集方面，本研究涵盖了域名注册信息、记录信息、子域名信息、解析历史、DNS 信息、SSL 证书信息及威胁情报。注册信息通过 WHOIS 方法获取，包含注册人详情、注册及到期时间等。DNS 查询则用于收集 A、MX、TXT 等记录及子域名信息，后者亦可通过脚本枚举获取。域名解析历史利用专业网站查询分析，而 DNS 信息虽漏洞少见，但可识别解析服务器的配置问题。SSL 证书信息通过资产测绘或自定义指纹识别收集，包括发行者、有效期及域名范围。威胁情报则来自微步威胁情报社区。本研究实现了子域名信息爆破功能，通过规整域名格式、加载子域名列表及循环请求判断，最终返回存在的子域名列表。功能实现部分代码如下：

```
def SubDomainDig(self,domain):
    domain = self.urlmethod.Domain(domain)
    subdomainlist = self.domainload.GetSubDomainList()
    found_subdomains = []
    for subdomain in subdomainlist:
        url = f’http://{subdomain}.{domain}’
        try:
            requests.packages.urllib3.disable_warnings()
            response = requests.get(url, timeout=5,verify=False)
            if response.status_code == 200:
            else:
        except Exception as e:
    return found_subdomains
```

3.2 端口信息

在获取目标开放端口信息方面，本研究采用了两种方法。首先，利用 Socket 技术，通过 PortScanner 类的 run 方法扫描指定 IP 地址的端口。该方法循环从 _portqueue 队列中获取端口号，尝试连接并捕获服务信息，存储于全局变量 ports 中，并更新 openPorts 字典。异常情况下，会捕获并打印异常信息。其次，采用 Nmap 工具结合其 Python 库进行端口扫描。以 NmapTCP 扫描为例，TCPScan 方法通过 Nmap 对指定 IP 进行 TCP 连接扫描，提取主机状态、开放端口及服务名称，并格式化为字符串输出。扫描过程中若发生异常，则捕获异常并返回错误信息。两种方法共同实现了对端口信息的全面获取。

3.3 中间件信息

中间件信息可以从多个方面获取，可以从 HTTP 响应头、端口指纹、应用层协议、错误消息、Web 技术指纹、流量特征分析、测绘信息等方面获得。下面将简单举例服务中间件指纹识别。该方法通过 get_server_by_html 获得页面的关键信息进行中间件与采用技术、框架进行判断。实现代码如下：

```
def get_server_by_html(html):
    soup = BeautifulSoup(html, ‘html.parser’)
    server = soup.find(‘meta’, attrs={‘name’: ‘Server’})
    if server:
        print(‘Server:’, server[‘content’])
    scripts = soup.find_all(‘script’)
    for script in scripts:
        if ‘wordpress’ in script.text:
            print(‘WordPress found’)
        if ‘jquery’ in script.text:
            print(‘jQuery found’)
        if ‘bootstrap’ in script.text:
            print(‘Bootstrap found’)
        if ‘fontawesome’ in script.text:
            print(‘FontAwesome found’)
        if ‘jquery-ui’ in script.text:
            print(‘jQuery UI found’)
                ……
```

3.4 旁站信息

旁站针对对目标网站的相邻 IP 地址或同一服务器上的其他站点进行信息收集。旁站信息于现阶段安全可以用来探测目标网站 C 段云服务器的信息，包括云供应商，对于云安全有已经可能存在的漏洞，可以通过这些漏洞来接管目标服务器。旁站信息收集可以首先使用列表推导式生成 IP 地址段列表，接着针对每个目标进行，从头进行 IP 信息收集。

3.5 目录与接口信息

扫描通过枚举目标服务器的目录结构，寻找潜在的敏感文件或目录。同时也通过枚举的方式，爆破目标服务器的接口信息。该部分通过定义了一个 dirScan 类用于目录扫描。__init__ 方法初始化类实例，创建 URLMethod 实例并生

成随机的 User-Agent 设置为请求头。getDirs 方法读取指定文件中的路径，去除每行末尾的换行符并返回路径列表。Scan 方法调用 getDirs 获取路径列表，初始化结果字典，标准化输入的 URL，遍历路径列表，拼接 URL 和路径，发送 HTTP GET 请求，检查响应状态码，如果不是 404 则记录并打印结果，处理异常并打印错误信息，最终返回结果字典。实现效果如图 2 所示。

图 2 目录接口爆破效果图

4. 漏洞模块

4.1 漏洞扫描

4.1.1 基本漏洞扫描

基本漏洞包括注入漏洞（SQL 注入、NoSQL 注入、LDAP 注入、代码注入等），失效的身份验证（弱密码、密码重用、会话管理不当等问题）、敏感数据泄露、XXE 外部实体漏洞、失效的访问控制、跨站脚本、反序列化漏洞、CSRF、SSRF 等基本漏洞。

4.1.2 CVE 漏洞扫描

CVE 漏洞扫描的两大重点就是 CVE 漏洞库的采集与更新，以及 CVE 漏洞扫描结果的标准化，CVE 漏洞库的采集与更新部分，可通过系统的情报系统功能对 GitHub 进行情报监控，获得 CVE 的利用脚本（Exploit）最新信息，并及时更新漏洞库。下面将通过 CVE-2022-22965 详细介绍 CVE 漏洞检测的过程，CVE-2022-22965 通过扫描方法，首先根据 ProxyStute 参数决定是否使用代理，然后构造 payload 并发送多个 HTTP 请求（两个 POST 请求和一个 GET 请求）。接着访问 tomcatwar.jsp 文件，检查响应状态码和内容，如果状态码为 200 且响应内容包含特定字符串，则认为存在漏洞并返回相关信息；否则返回漏洞不存在或已被利用的信息。整个过程中还包含了异常处理，捕获并返回任何可能的异常。

4.2 漏洞验证

漏洞验证通过请求与响应实现，几乎所有的验证均按照这个模式进行。上述我们完成了 CVE-2022-22965 漏洞存在的判断，紧接着我们可以对目标进行进一步利用。该部分首先从配置文件中读取漏洞利用类型 expType，然后根据 expType 构建不同的 Shell URL。接着根据 ProxyStute 参数决定是否使用代理，然后发送 HTTP GET 请求并获取响应。如果请求成功，返回响应内容；如果发生网络请求异常，捕获并返回相应的错误信息。

漏洞验证利用部分的代码如下：

5. 报告生成模块

报告模块，需要提前定义报告内容，该部分初始化 GenReport 类，根据项目 ID 从数据库中获取项目信息、端口信息和漏洞信息，生成相应的表格并填充到 Word 模板文档中，最后保存生成的报告并返回文件名。整个过程中，InfoSet 方法负责从数据库获取数据并生成表格，InfoUpdate 方法负责更新文档中的占位符为实际数据。

6. 结束语

项目阐述了网络安全漏洞扫描系统的扫描核心功能模块的设计方法，通过实践验证，该系统具有高度的灵活性和可扩展性，能够准确快速地发现潜在的安全隐患，为网络安全防护提供了有力的技术支撑。展望未来，我们将继续优化和完善漏洞扫描系统的功能和性能，提升其检测精度和效率。同时，我们也将积极探索新的网络安全技术和方法，以应对日益复杂多变的网络威胁。

参考文献：

[1] 张伟 , 李明 . 网络安全漏洞扫描技术研究综述 [J]. 计算机科学 , 2020, 47(11): 220-226.

[2] 王强 , 赵雷 . 基于 Nmap 的端口扫描技术研究与应用 [J]. 信息网络安全 , 2019, (5): 64-70.

[3] 刘晓霞 , 陈俊 . Web 应用漏洞扫描系统的设计与实现 [J]. 计算机应用与软件 , 2018, 35(9): 182-187.

项目信息：

刘彬 -2024- 基于鲲鹏架构的国产化网络安全漏洞扫描系统（国家级创新创业项目）；刘彬 -2023- 创新性网络安全漏洞 POC/EXP 管理巡检系统设计（2023YB08）