面向网络资产漏洞评估的设备指纹搜索引擎构建方法

姚茗亮[1],鲁 宁[1,2],白撰彦[1],刘懿莹[1],史闻博[1]
(1. 东北大学计算机科学与工程学院,辽宁沈阳 110819;2. 西安电子科技大学计算机科学与技术学院, 陕西西安 710071)

摘 要: 网络资产漏洞评估技术对于梳理互联网资产、实现网络资产漏洞安全管理起到十分重要的作用. 已有方法因无法有效获取设备指纹信息而产生评估结果准确度低、功能单一等问题. 为此,本文提出一种面向网络资产漏洞评估的设备指纹搜索引擎构建方法,具有多接口、交互性强、减少网络冗余探测的优势. 本文通过统计和评估真实网络中的服务器类型、HTTPS 协议漏洞来验证方法的高效性.

关键词: 网络资产; 漏洞评估; 设备指纹搜索
中图分类号: TP399 **文献标识码:** A **文章编号:** 0372-2112 (2019)11-2354-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.11.017

Construction Method of Device Fingerprinting Search Engine for Network Asset Vulnerability Assessment

YAO Ming-liang[1], LU Ning[1,2], BAI Zhuan-yan[1], LIU Yi-ying[1], SHI Wen-bo[1]
(1. *School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning* 110819, *China*;
2. *School of Computer Science and Technology, Xidian University, Xi' an, Shaanxi* 710071, *China*)

Abstract: The network asset vulnerability assessment technology plays an important role in sorting out Internet assets and realizing the security management of network asset vulnerabilities. The existing methods have the issues such as low accuracy of evaluation results and single function due to the inability to effectively obtain the fingerprinting information of the device. Therefore, this paper proposes a construction method of device fingerprinting search engine for network asset vulnerability assessment, which has the advantages of multiple interfaces, strong interactivity, and reduced network redundancy detection. This paper verifies the efficiency of the method by counting and evaluating server types and HTTPS protocol vulnerabilities in real networks.

Key words: network assets; vulnerability assessment; device fingerprinting search

1 引言

2017 年爆发的软件勒索病毒 WannaCry 事件[1]表明:种类庞杂、更换频繁的信息系统使得网络管理者无法清晰、及时地掌握其漏洞,进而修复它们. 为此,如何持续性地跟踪和评估信息系统的漏洞就显得非常重要.

网络资产漏洞评估技术就是针对信息系统的资产管理混乱、漏洞跟踪不连续等问题而提出的[2]. 该技术作为预防网络攻击发生的重要措施,一经提出就受到研究者们广泛关注. 例如:Genge B 等人提出一种漏洞资产分析方法 ShoVAT[3],它利用已商业化的网络空间探测引擎 SHODAN 来充当设备指纹源,并将相关搜索结果与美国国家漏洞库相映射,确定相关设备可能存在的安全漏洞;Luis Alberto 等人在假设软件指纹已收集的前提下,设计了一种面向软件漏洞的资产管理方法 IVA[4],该方法通过静态分析检测软件存在的漏洞;Kai Simon 提出一种基于 Google 和 SHODAN 分析网络资产漏洞的方法[5],可以快速确定潜在的网络资产漏洞,并有效检测出零日攻击. 虽然上述方法在一定程度上实现了网络资产漏洞评估功能,但由于他们更侧重漏洞匹配而忽略了指纹探测,因此存在准确度低和功能单一等问题,基于此,本文重点关注网络资产漏洞评估过程中的设备指纹探测方法.

目前,可以通过网络空间探测引擎(如 SHODAN、

收稿日期:2018-10-11;修回日期:2019-04-18;责任编辑:李勇锋
基金项目:国家自然科学基金(No. 61601107);国家自然科学基金重点项目(No. U1708262);中央高校基本科研业务费(No. N172304023)

CENSYS 等)搜索设备指纹. 但是存在以下缺陷:1)其内部架构尚未公开;2)对相对敏感的原始数据经过处理;3)用户大量调用、获取数据需要权限. 这些原因使得通过该类搜索引擎分析数据的难度较大. 而现有收集设备指纹的工具如 ZMAP、NMAP、MASSCAN 等虽然具有强大的功能,但缺乏友好的交互方式,而且这类工具存在对同一网段重复探测等问题. 为此,本文设计了一种面向网络资产漏洞评估的设备指纹搜索引擎构建方法,该方法能够高效地搜集设备指纹信息,并通过 Web 搜索引擎和云数据库的方式供用户查询. 相比于其他发方法,利用本文所述方法具备如下优势. 第一,提供多接口查询数据. 本文所述方法不仅可以利用搜索引擎界面查找数据,而且所有数据都会上传到云数据库,并提供接口供研究人员访问. 第二,结果可交互. 采用良好的前后端工作分离模式,通过搜索引擎交互,可在 Web 页面展示多条探测结果. 第三,减少网络冗余探测. 通过建立指纹数据库,不断更新并保存已探测过网段的指纹信息,避免设备指纹信息利用率低下的问题.

为了验证本文所提方法对网络资产漏洞评估的作用和帮助,首先统计全球不同类型的服务器排名;然后对 HTTPS 协议的三个漏洞进行漏洞评估. 结果表明,本文所提方法不仅可收集指纹数据,还可监测漏洞在全网的影响.

2　相关工作

网络资产漏洞评估需经过网络资产探测和漏洞分析评估两步. 前者实现网络设备信息收集功能;后者分析漏洞和评估. Genge B 等人提出了自动漏洞识别工具 ShoVAT[3]. 该工具将漏洞评估功能嵌入到网络空间探测引擎 SHODAN 中,利用 SHODAN 搜索结果中的 server banner 字段,结合美国国家漏洞库检测全网设备可能存在的漏洞. 不过,ShoVAT 并不能保证 SHODAN 每次的查询结果都是正确的,缺乏验证,而且 SHODAN 的查询内容至多只能对一类网络资产进行评估,不能全面评估漏洞现状. Kai Simon 提出一种基于 Google 和 SHODAN 功能的组合来分析漏洞的方法[5]. 这种方法能够快速地确定潜在网络资产的漏洞,并有效检测零日攻击. 不过,该方法因太过依赖 SHODAN,使得后者的限制会影响它对网络资产漏洞分析的准确性;Luis Alberto 等人提出一种采用静态分析方法检测软件漏洞的漏洞管理系统 IVA[4]. IVA 需要使用专门的工具提前收集系统软件的指纹信息,同时结合美国国家漏洞库进行漏洞分析. 然而,它因指纹收集工具的局限性影响了漏洞评估的效果. Wang 等人为漏洞分析和漏洞管理提供了一个框架 OVM[6]. 它根据收集到的 IT 产品信息进行漏洞分析,可以准确描述外部威胁和内部漏洞. 但是,如果不能准确、有效的收集设备信息,那么漏洞分析结果也会缺乏权威性. 综上所述[3-6],不难发现大部分网络资产漏洞评估方法都需要先收集指纹信息,再利用某些方法进行漏洞评估. 所以,如何更有效地收集指纹信息成为制约网络资产漏洞评估方法好坏的关键.

设备指纹信息收集方法按照扫描方式可分为主动扫描和被动扫描. 前者指通过检查内部网络 IP 的有关服务和版本的信息来识别操作系统的类型,还可以检测对应 IP 是否含有某种漏洞,如:NESSUS[7]. 但由于该方法采取一定的攻击方式,存在窃取目标隐私信息的风险,因此不适合科学研究;后者主要通过发送和接收正常通信消息来收集有关设备的信息,本文所提出的方法属于后者. 目前,已有的网络空间探测引擎 SHODAN 和 CENSYS 等尚未公开架构、数据也未开源,这也是大量相关研究受到限制的主要原因. 而本文提出一种设备指纹搜索引擎的构建方法,既能解决传统收集设备指纹的工具不易交互的缺陷,又能为相关网络资产漏洞评估技术的研究提供全面的指纹信息.

3　设备指纹搜索引擎构建方法

本节将详细阐述面向网络资产漏洞评估的设备指纹搜索引擎构建方法的设计思想和实现机制.

3.1　基本思路

整体架构的设计思想为:设备指纹收集模块中的主机设备探测模块负责扫描不同端口的 IP,其中指纹信息获取模块将抓取每个 IP 对应的指纹信息并以 JSON 格式保存. 由于获取到的初始指纹数据格式混乱,包括畸形格式和无法响应的数据. 因此需要通过数据清洗模块过滤掉无关数据同时将格式良好的数据暂存到调度模块中. 调度模块使用本文提出的缓存审查法可以有效缓解导入指纹数据库时发生的堵塞,还能通过审查机制避免重复数据导入数据库. 数据存储模块将持续接收并存储来自调度模块的指纹数据,并定期将数据备份到云数据库中,供研究人员使用. 搜索交互模块作为与用户交互的主要接口,通过搜索引擎与用户交互. 设备指纹搜索引擎具体架构如图 1 所示.

3.2　设备指纹收集模块

网络扫描是信息收集的重要手段,通过扫描可以发现存活主机和开放端口,进而获取其设备指纹信息. 本文选择扫描工具进行设备探测.

3.2.1　主机设备探测

本文针对 ZMAP、NMAP、MASSCAN 三种工具的端口扫描、主机发现功能进行实验对比,实验结果如图 2、图 3 所示. 由图 2、图 3 可知,针对同一 C 段网络扫描到的存活 IP 数目基本相同,而 ZMAP 的扫描时间最短且更稳定. 综上所述,选取 ZMAP 作为本文的扫描工具.

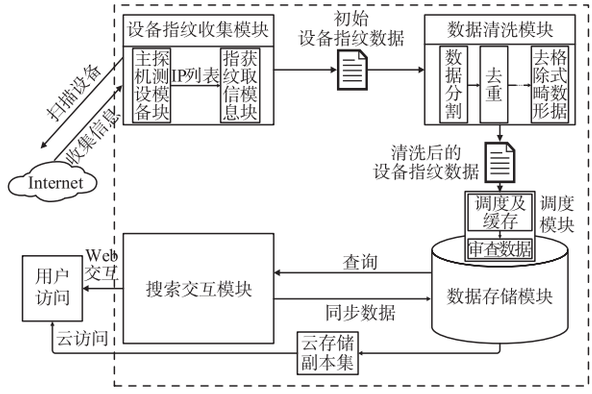

图1 设备指纹搜索引擎架构图

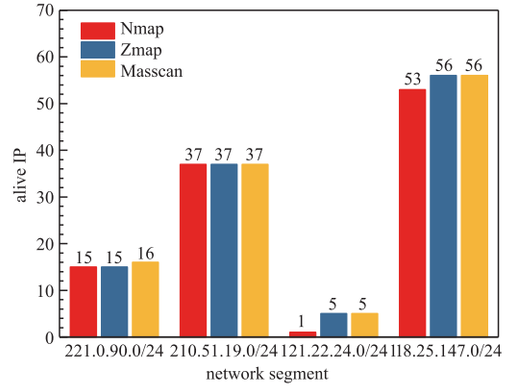

图2 三种工具针对80端口扫描存活IP数目

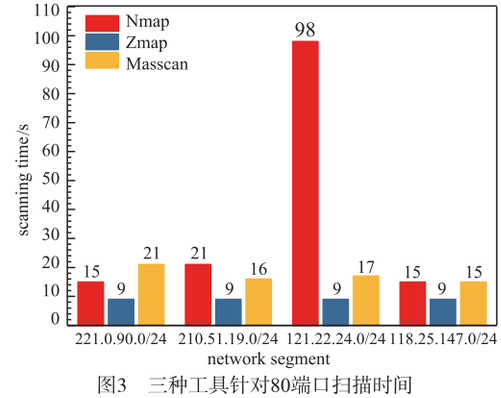

图3 三种工具针对80端口扫描时间

3.2.2 指纹信息获取

在收集到大量的 IP 数据后,需要获取每个 IP 相应的指纹信息. 本文通过 Zgrab 工具批量获取指纹信息. 指纹信息抓取的原理是先对主机端口发送数据请求信息,并保持请求内容基于相应协议的支持,便可得到主机的响应内容,其中包括但不限于响应头、服务器名称、国家名称等字段信息.

3.3 数据清洗模块

由于抓取到的指纹信息存在格式畸形等问题,所以需要数据清洗模块进行处理. 该模块包括以下两个步骤. (1)分割数据. 由于计算机无法一次性读取几百 G 指纹数据,因此必须要分割大块数据. 本文自行编写分割数据的脚本工具,采用按行分割的原理,能够保证被切割保数据的完整性. (2)去除畸形格式数据. 针对被切割后的数据,对每块包含畸形格式和未响应特征的数据进行清洗.

3.4 数据存储及调度模块

设备指纹搜索引擎每天都会扫描到海量数据,这些数据在经过数据清洗模块处理之后需要存储到数据存储模块中,本文选取适合分布存储的 Mongodb[8]. 因为数据是以日志结构合并树[9]的形式写入磁盘中,所以通过维护所有记录的全局排序来对描述单个主机的多个记录进行逻辑分组,并输出描述每个主机的结构化 JSON 文档,来高效地生成 IPv4 地址空间的每日快照.

为了解决将海量数据导入指纹数据库可能会导致拥塞、数据重复导入等问题,本文提出一种缓存审查方法解决导入数据时的数据拥塞和重复导入的问题. 该方法将处理后的结构化数据反序列化为缓冲数据,同时审查是否与之前导入过的指纹数据相同. 如果检测出指纹数据发生更新,则新的指纹数据将被导入数据存储模块. 如果检测出指纹数据没有变化,则只需将该条数据标记为在最近一次扫描中已审查并删除.

3.5 搜索交互模块

搜索交互模块采用 ElasticSearch[10]结合 Mongodb 实现. 为加快查找速度,本文利用同步工具 monstache[11]将数据存储模块中的数据同步到 Elasticsearch 中. 不仅可以在多个服务器上提供数据备份,还可以存储数据副本,提高数据的可用性,保证数据安全. 交互界面的后端采用了基于 PHP 的 Laravel 开源框架设计实现,通过搭建搜索引擎帮助用户进行数据交互,实现搜索指纹数据的功能.

4 实例分析

本文分别在 2017 年至 2019 年期间对不同端口执行多次扫描,主要端口包括 21、22、25、53、80、110、143、443、502、995 等,共收集约 1.39T 指纹信息作为全球样本数据. 下面将以 443 端口的指纹数据为例,从服务器类型和 HTTPS 协议漏洞两个角度评估全球安全态势.

4.1 服务器类型统计

全球不同国家和地区都在使用不同的服务器建设自己的网站,而占领全球市场份额越多的服务器越有可能成为黑客的攻击对象. 因为这些漏洞并不是代码上存在的漏洞,而是由于环境自身或环境配置不当造成. 全球服务器类型的统计结果如图 4 所示. 在可访问的 19910775 条有效数据中,AkamaiGHost 和 Apache 服

务器分别占据了全球 29.96% 和 20.89% 的份额,接近全球一半的市场份额. 而不可访问的服务器("unknown")约占样本数据的 15.53%,也就是说全球约有 85% 的服务器处于直接暴露在互联网空间的状态,没有较好的防护措施.

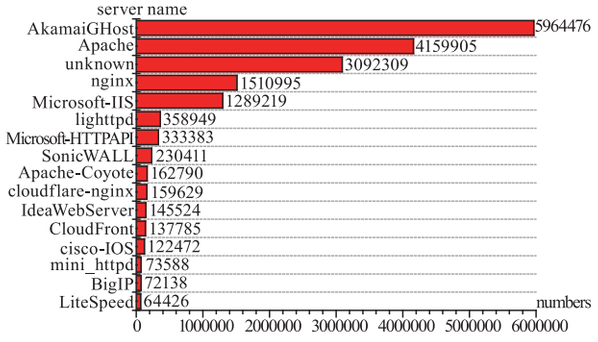

图4　全球服务器类型统计

4.2　HTTPS 协议漏洞评估

为了探究 HTTPS 协议漏洞在全球范围的情况,利用本文提出的方法,结合分布式 POC 验证和漏洞威胁特征识别技术,以"POODLE"漏洞、"DROWN"漏洞和"HeartBleed"漏洞为例,进行漏洞评估.

4.2.1　POODLE 漏洞

"POODLE"漏洞又被称为"贵宾犬攻击",攻击者针对 SSL3.0 版本窃取通信内容.

本文对 4212000 条样本数据检测其是否含有"POODLE"漏洞. 在可访问的 903307 条样本数据中,有 30664 条存在该漏洞,遍布全球 165 个国家和地区. 该漏洞在全球分布数量排名前十位的国家和地区具体状况如图 5 所示. 其中该漏洞在各国占有率排前三位的国家分别是中国(6.44%)、法国(5.68%)、德国(3.70%).

"POODLE"漏洞的分布,能够从侧面反映出一个国家或地区对 SSL3.0 版本的依赖情况. 实验结果表明,"POODLE"漏洞主要分布在北美洲、东亚、欧洲等地区. 这些地区对 SSL3.0 版本依赖程度较高,大部分网站都兼容 SSL3.0,而这也使得该漏洞在这些地区无法被有效消除. 从安全性角度考虑,网站建设者不应继续使用该版本建设网站,否则该漏洞无法从根本上得到解决.

4.2.2　DROWN 漏洞

"DROWN"漏洞又被称为"溺水攻击",针对 SSL 2.0 版本协议漏洞对 TLS 进行跨协议攻击.

本文对 4848000 条样本数据进行检测,在可响应的 60905 条样本数据中,有 37631 条数据显示依然存在该漏洞,遍布全球 156 个国家和地区. 该漏洞在全球分布数量排名前十位的国家和地区具体状况如图 6 所示. 其

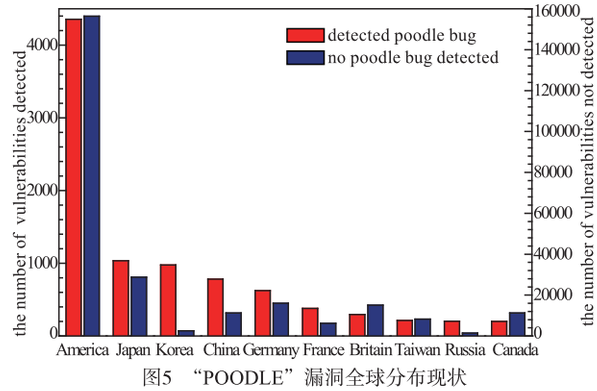

图5　"POODLE"漏洞全球分布现状

中,本文检测到以色列有 4710 个服务器含有该漏洞,漏洞占有率以 99.16% 高居第一,成为被该漏洞危害最严重的国家. 而中国台湾和乌克兰也分别以 94.69% 和 92.67% 的漏洞占有率排在二三位.

"DROWN"漏洞与"POODLE"漏洞相比,在全球的分布更为严重,尤其是东亚、中东等地区. "DROWN"漏洞本身是由于 SSL 2.0 版本协议较老而存在漏洞,如果能够及时更新,则该漏洞就能够得到有效预防. 不过,实验结果表明,该漏洞在全球大部分国家和地区并未得到有效控制.

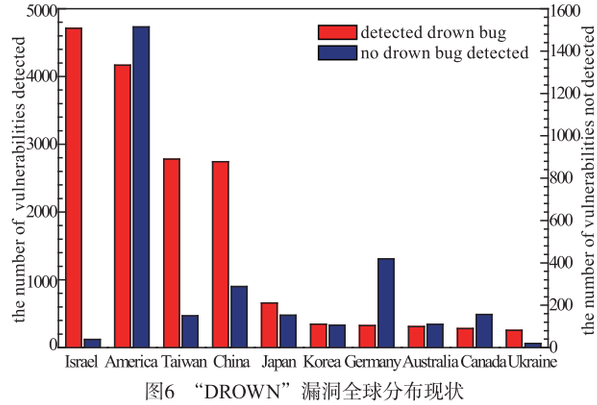

图6　"DROWN"漏洞全球分布现状

4.2.3　HeartBleed 漏洞

"HeartBleed"漏洞又被称为"心脏滴血漏洞",是一个出现在加密程序库 OpenSSL 的安全漏洞.

通过对 53324611 条样本数据的检测分析,在可访问的 117736 条有效数据中,有 256 条数据显示依然含有该漏洞. 全球存在心脏滴血漏洞数量排名前 10 位的国家和地区的情况具体如图 7 所示. 其中,"心脏滴血"漏洞占有率排名前三位的国家分别是美国(1.76%)、中国(0.67%)、澳大利亚(0.63%).

实验结果表明,随着技术的发展和漏洞补丁的普及,大部分"心脏滴血"漏洞得到了有效修复,所以该漏洞对应的危险级别也可以相对调低.

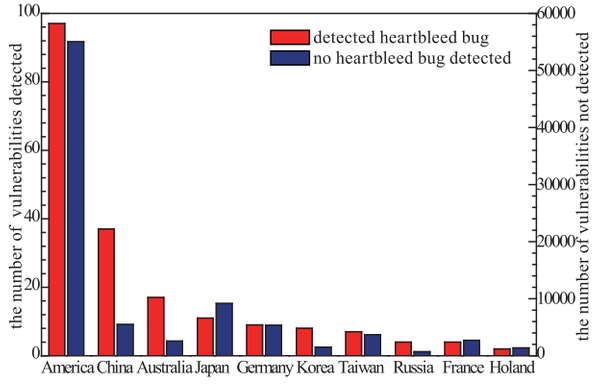

图7 “心脏滴血”漏洞全球分布现状

5 结论及工作展望

针对现有的网络资产漏洞评估方法中存在的问题,本文提出一种面向网络资产漏洞评估的设备指纹搜索引擎构建方法. 该方法将原本独立的设备指纹收集、清洗、存储、分析等功能集成到一个架构中,并通过搜索引擎和云数据库接口进行交互.

未来的研究工作主要包括:1)通过融合 IDS(intrusion detection system)和海量日志文件等多源信息,提出一种高精度的网络资产漏洞评估方法;2)将漏洞检测与弱口令爆破技术相结合,提出高效的预防网络攻击的方法.

参考文献

[1] Mattei T A. Privacy, confidentiality, and security of health care information: lessons from the recent WannaCry cyber-attack[J]. World Neurosurgery, 2017, 104: 972 – 974.

[2] Füssel H M, Klein R J T. Climate change vulnerability assessments: an evolution of conceptual thinking[J]. Climatic Change, 2006, 75(3): 301 – 329.

[3] Genge B, Enăchescu C. ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services[J]. Security and Communication Networks, 2016, 9(15): 2696 – 2714.

[4] Luis Alberto, Benthin Sanguino, Rafael Uetz. Software Vulnerability Analysis Using CPE and CVE[DB/OL]. http://cn. arxiv. org/pdf/1705. 05347, 2017.

[5] Kai S. Vulnerability Analysis Using Google and Shodan [M]. Kuala Lumpur: Cryptology and Network Security, 2016. 725 – 730.

[6] Wang J A, Guo M. OVM: an ontology for vulnerability management[A]. Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies[C]. ACM, 2009. 34.

[7] Beale J, Deraison R, Meer H, et al. Nessus Network Auditing[M]. Syngress Publishing, 2004.

[8] Banker K. MongoDB inAction[M]. Manning Publications Co, 2011.

[9] O’Neil P, Cheng E, Gawlick D, et al. The log-structured merge-tree (LSM-tree)[J]. Acta Informatica, 1996, 33(4): 351 – 385.

[10] S Banon. Elasticsearch[EB/OL]. https://www. elastic. co, 2013.

[11] Ryan Wynn, Ahmed Magdy. Monstache[EB/OL]. https://github. com/rwynn/monstache, 2016.

作者简介

姚茗亮 男,1994 年 1 月出生于辽宁朝阳,2017 年于东北大学攻读硕士学位,现为硕士研究生. 主要研究方向为网络漏洞评估.
E-mail: rye_learnmore@163. com

鲁 宁 男,1984 年 9 月出生于内蒙古包头,博士,东北大学副教授,硕士生导师. 主要研究领域为网络安全.
E-mail: snowting915@126. com