网络空间终端设备识别框架①

曹来成[1], 赵建军[1,2], 崔 翔[2], 李 可[2,3]

[1](兰州理工大学 计算机与通信学院, 兰州 730050)
[2](中国科学院 信息工程研究所, 北京 100093)
[3](北京邮电大学 计算机学院, 北京 100876)

摘 要: 针对传统 Web 服务识别方法对服务器软件实现差异依赖性强、不利于扩展到其他终端设备等缺陷, 提出了一种网络空间终端设备双重因素主动识别框架. 该框架从服务标识(Banner)和 Web 指纹两个角度分析终端设备网络指纹特征. 首先, 从 HTTP 数据包头部的“Server”字段、“WWW-Authenticate”字段及 HTML 源代码中深入挖掘终端设备 Web 服务中的 Banner 信息; 其次, 从图片文件和文本文件中提取终端设备具有的 Web 指纹特征; 同时, 结合具体案例分析各识别方法的有效性及优缺点. 结果表明, 该框架解决了传统识别方法中 Banner 信息获取途径单一、依赖服务器软件差异特征的问题, 成功将 Web 服务识别方法扩展到了网络空间终端设备上. 最后, 总结已有成果并探讨该框架未来改进方向.

关键词: 终端设备; 主动识别; 服务标志; Web 指纹; 网络指纹特征

Cyberspace Terminal Device Identification Framework

CAO Lai-Cheng[1], ZHAO Jian-Jun[1,2], CUI Xiang[2], LI Ke[2,3]

[1](School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China)
[2](Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China)
[3](School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Since the traditional web service identification methods are highly depended on the differences of server software implementation, unfavorable to expand to other terminal devices, a kind of proactive identification framework with double factors for cyberspace terminal device is proposed. The framework analyzed the network fingerprint characteristics of terminal device from perspectives of banner and web fingerprint. Firstly, banner information of web service in terminal device is mined from “Server” field and “WWW-Authenticate” field and HTML source code in HTTP packet header; secondly, web fingerprint of terminal device wasis extracted from image files and text files; meanwhile, it analyzes the effectiveness as well as advantages and disadvantages according to the specific cases. The results show that the framework solves the problems of extracting banner information simply and being limited by the differences of server software implementation in traditional identification methods. Finally, it summarizes research results and discusses future improvements of the framework.

Key words: terminal device; proactively identification; banner; web fingerprint; network fingerprint characteristic

随着物联网技术及各设备间通信协议的发展, 网络空间中的终端设备已越来越多. 无线路由器、数字视频摄像机、网络打印机等终端设备都具有了公网 IP 地址, 这些终端设备和传统的主机、服务器及路由器共同构成了当前的网络环境. 当前网络空间中终端设备的规模十分庞大、类型十分复杂, 同时带来的安全隐患也是前所未有的.

网络空间终端设备往往具有各种公开或未公开的安全漏洞, 将大量用户置于持久性的安全威胁之中. 攻击发生后, 用户甚至不能发现自己已被攻击, 或者

① 基金项目:国家自然科学基金(61562059,61461027)
收稿时间:2015-12-30;收到修改稿时间:2016-03-07 [doi:10.15888/j.cnki.csa.005310]

不能确定被攻击的脆弱点. 与以往的攻击方式不同, 当今的黑客攻击可能针对一个无线路由器以重置路由器信息、获得管理员权限, 或者攻击大型工业控制系统扰乱工业生产等. 例如, 海康威视网络数字视频录像机(DVR)在 2014 年被公开存在一个安全隐患[1], 原因是该设备使用的出厂默认口令为弱口令. 设备管理员若不对出厂默认用户名和密码进行修改, 攻击者就可以使用管理员用户名“admin”和密码“12345”等弱口令登陆后台管理界面. 又如 2013 年 10 月, 中国国家互联网应急响应中心发布的安全报告显示多款 D-LINK 路由器产品固件存在后门漏洞[2]. 攻击者只需要将浏览器的“用户代理(User-Agent)”标志修改为“xmlset_roodkcableoj28840ybtide”, 再访问路由器 IP 地址, 则无需身份验证即可访问路由器的 Web 管理界面, 查看或者更改受影响设备的配置信息.

如今黑客攻击的方式更加多样, 针对的设备更加广泛, 利用的漏洞更加复杂, 造成的威胁更加严重, 进一步凸显出了全面统计网络空间终端设备的分布情况以及各设备脆弱点的重要性和紧迫性. 特定的安全漏洞往往威胁某一类终端设备, 或是某种终端设备的某个特定型号, 因此, 对网络空间终端设备进行安全评估和安全预警就要求能够对终端设备的类型及其型号进行准确的识别. 以往的识别技术和防护技术都只是针对传统的 Web 服务器软件, 并不能完全适用于新的网络环境. 本文在此方向上探索和研究新的终端设备识别方法, 并为大规模网络空间终端设备安全评估和安全预警提供思路.

1 相关工作

生物学上的指纹识别是指通过光电扫描和图像处理技术, 对人体指纹进行采集、分析和对比, 能精确地鉴别出个人身份的技术[3]. 将此概念引申至计算机网络学科中, 则指通过获取主机或终端设备在网络通信中产生的流量, 运用某种匹配算法对流量中的网络特征进行识别, 从而获取主机或终端设备某些信息的技术[4].

Fyodor 在文献[5]中率先提出了 TCP/IP 栈指纹的概念. 由于各类操作系统 TCP/IP 协议栈的实现不同, 因而对于不同的探测报文, 不同的操作系统或相同操作系统的不同版本会产生不同的响应, 从而可以根据 TCP/IP 栈指纹的差异来对操作系统进行识别.

为了进一步识别远程主机 Web 端口上运行的 Web 服务, 服务标识(Banner)采集和 Web 指纹识别的概念也被提出[6]. Banner 信息是某些服务器软件通过网络向发出请求的用户反馈的自身软件名称及版本等标志性基本信息. 对运行着 Web 服务的主机进行对应的连接, 缺省情况下将会得到远程主机返回的 Banner 信息. 例如, 对“www.apache.org”进行 HTTP 连接, 在返回的 HTTP 数据包中就会包含“Server:Apache/2.4.7 (Ubuntu)”字样. 通过对这些 Banner 信息进行解析, 可以识别出目标主机所安装的服务器软件及对应版本.

但是基于 Banner 信息的 Web 服务识别方法并不能做到百分之百准确, 原因是 Web 服务器返回的 Banner 信息能够被人为的修改、伪装和模糊[7]. 通常的做法可以分为两种: 一是修改服务器软件的源代码或相关二进制文件; 二是使用商业软件或服务器插件. 对于像 Apache、Nginx 这样的开源 Web 服务器软件, 通过修改其源代码中的 Banner 信息部分, 便可以做到对 Banner 信息的伪装; 对于非开源 Web 服务器软件, 比如 IIS、Netscape 等, 通过修改存放 Banner 信息的动态链接库(DLL)文件, 也能实现隐藏真实 Banner 的目的. 使用商业软件或服务器插件同样能够做到隐藏 Banner 信息, 这些商业软件或服务器插件可以提供自定义的 HTTP 应答信息, 例如商业软件 ServerMask[8]就具有这样的功能. ServerMask 是 IIS 服务器软件的一个插件, 它不仅能够隐藏 Banner 信息, 而且能够对 HTTP 响应包头部中各字段的序列进行重新组合, 从而来模仿 Apache 这样的服务器软件. ServerMask 甚至有能力伪装成任何一个 Web 服务器软件来处理每一个请求.

对于进行过修改、伪装和模糊的 Web 服务器软件, 利用 Banner 信息识别服务器软件的方法已经失效. 为此, Dustin 等人的 HMAP[9]采用了一种不依赖 Banner 信息的 Web 服务识别技术. 即通过不同 Web 服务器软件之间反馈信息的差异来识别, 例如对于“404”类型错误, 某些服务器软件反馈的原因短语为“Object Not Found”, 而有些服务器软件反馈的原因短语为“Not Found”. 同时, HMAP 也利用了 Web 服务器软件对超长 URL 的处理方式差异来识别.

杨可新等人认为, HMAP 所采用的方法从根本上还是依赖于服务器软件返回的文字或语法的差异, 而这种差异仍然能够通过修改源代码或二进制文件以及使用商业软件或插件来进行修改; 同时, 利用超长

URL 来对服务器进行探测则会增加 Web 服务器的处理负担, 甚至导致缓冲区溢出, 形成拒绝服务, 因此在使用上具有相当高的安全风险[10]. 为此, 杨可新等人提出一种类似 TCP/IP 栈指纹识别的 Web 指纹识别方法. 该方法不依赖于 Web 服务器软件返回的文字或语法差异, 而是根据不同的 Web 服务器软件对畸形 HTTP 请求的处理方式不同来进行识别. 这种方法的核心原理是, 不同的服务器软件内部实现不同, 因而对于没有明确规定的非正常请求, 不同的服务器软件会按不同的方式处理.

但是这种方法仍然存在一个缺陷. 不同服务器软件实现上的差异是有限的, 因而当服务器软件的种类和版本逐渐增多时, 通过这种方法来识别将会出现重复, 导致误判[11]. 而且此种方法只能适用于主版本的识别. 对于主版本相同的服务器软件, 各次版本内部的改动较小, 处理方式与其他次版本基本一致, 因此该方法对服务器软件具体次版本的识别能力较差. 如今的网络空间中, 终端设备的类型十分繁多. 路由器、无线接入点、网络打印机、数字视频摄像机、智能家用设备等等这些终端设备 Web 服务的实现均各不相同, 并且每种终端设备又有许多型号和版本, 因此之前的识别方法已不能适用于新的网络环境.

2　网络指纹一般模型

唐彰国等人在分析了何聚厚[4]的文章后提出一种网络指纹模型[12]. 他指出, 能够作为网络指纹的网络特征应具有以下属性: (1)唯一性; (2)稳定性; (3)可区分性; (4)可测性. 将此网络指纹模型运用到网络空间终端设备上, 本文的解读如下:

1) 唯一性: 对于某一网络空间终端设备个体而言, 相同的探测得到的结果唯一;

2) 稳定性: 对于某一网络空间终端设备个体而言, 探测结果与探测发生的时间、地理位置、IP 地址和网络拓扑结构无关;

3) 可区分性: 对于网络空间中不同的终端设备而言, 相同的探测得到的探测结果是不同的;

4) 可测性: 网络空间终端设备的网络特征值应该是可测量的, 并且是可分辨的.

该网络指纹模型进一步说明了利用服务器软件实现差异进行识别的局限性. 在当前终端设备大规模化、多样性化的网络环境下, 使用之前的识别方法, 可能会出现探测不同终端设备得到相同探测结果的情况, 从而违背了网络指纹模型中的“可区分性”.

依据上述网络指纹一般模型, 隋新提出一种利用“Banner+Fingerprint”双重因素进行被动识别的方法[13]. 在分析了文献[13]后发现, 对于没有进行过修改、伪装和模糊的服务器软件, 利用 Banner 信息在识别 Web 服务器软件次版本时精确度较高; 同时, 多重因素识别技术也弥补了单一因素识别技术中的一些缺点和不足. 当某种因素识别失败或不准确时, 依靠其他因素缩小识别范围, 降低识别难度, 从而最终能够达到较好识别效果, 这一点为网络指纹识别技术提供了新的思路.

但是对于网络空间中的所有终端设备而言, 被动识别技术并不适用. 虽然被动识别对网络性能的影响较小, 但仅仅依靠网络嗅探或流量监听是不能满足对数量繁多的终端设备进行大规模识别的, 从而难以实现全面统计网络空间终端设备的分布情况以及各设备脆弱点的最终目的. 因此本文认为, 针对网络指纹模型中的“可测性”, 该方法仍存在需要优化的地方.

3　终端设备“Banner+Web指纹”双重因素主动识别框架

在比较各种传统远程主机识别方法优缺点并分析网络指纹一般模型之后, 本文提出一种“Banner+Web指纹”双重因素主动识别框架. 该框架共从两个角度、五个方面发现、获取和识别终端设备的网络指纹, 并能够保证探测所得网络指纹均符合网络指纹一般模型中的各项属性. 同时, 针对一个终端设备的所有探测均只使用正常的 HTTP 请求, 避免了探测请求被安全防御设备拦截及引发异常流量报警. 该框架的具体框图如图 1 所示.

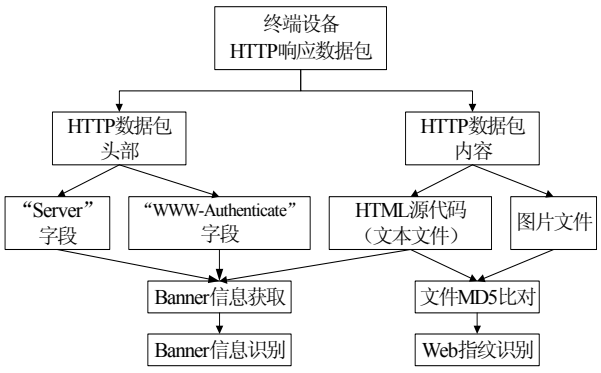

图 1　双重因素主动识别框架

3.1 终端设备 Banner 信息识别

Banner 信息识别是一种快捷、有效的识别方法, 并且能够较精确的识别服务器软件的具体版本信息. 但是如果服务器软件进行了某种安全加固, 对 Banner 信息进行了修改、伪装和模糊, 利用 Banner 信息进行识别的方法便会失效. 但是对于网络空间终端设备而言, 终端设备一经出厂, 其中的 Banner 信息便存储在终端设备的固件或硬件当中, 没有生产厂家特定的工艺流程, 无法完成对 Banner 信息的修改; 此外, 以往能够修改Banner信息的商业软件和服务器插件都仅仅适用于传统的 IIS 等 Web 服务器软件, 目前尚不存在能够修改终端设备 Banner 信息的应用软件. 因此, 本文认为通过Banner信息进行终端设备识别依然是可行且有效的, 同时探究了 Banner 信息识别的新方法, 对终端设备存在的Banner信息从网络指纹各部分进行了深入挖掘.

3.1.1 HTTP 数据包头部“Server”字段中的 Banner 信息

HTTP 数据包头部的“Server”字段是 Web 服务器用来表明自己是什么软件及版本等信息的字段. 文献[14]中第 14.38 小节规定了“Server 字段”的具体含义, 一个“Server”字段所包含的信息可能是如下类型: “Server:Apache/2.4.7 (Ubuntu)”. 表明该 Web 服务器搭建在 Ubuntu 操作系统上, 服务器软件是 Apache, 版本是 2.4.7. 文献[6]中已经对利用“Server”字段中的Banner信息进行Web服务器软件识别的方法做了阐述, 本文将这种方法进行了改进与扩展, 从针对传统 Web 服务器软件(Apache、IIS、Nginx 等)的识别扩展到了对终端设备 Web 服务的识别, 从而通过终端设备 Web 服务进一步识别终端设备的类型、版本等.

对于网络空间终端设备而言, 在用户与设备的通信过程中, 某些终端设备会将其特有的相关信息作为 HTTP 数据包头部“Server”字段的内容返回给用户. 例如, 在浏览器中直接访问一个已知的海康威视数字视频录像机 Web 服务的 IP 地址, 并通过 Wireshark[15]等抓包软件对 HTTP 数据包进行截获, 可知该海康威视网络数字视频录像机 Web 服务返回的 HTTP 数据包头部中“Server”字段的内容为“DVRDVS-Webs”, 如图 2 所示. 在采集大量样本并进行实验后发现, 海康威视网络数字视频录像机 Web 服务返回的 HTTP 数据包头部中“Server”字段内容均为“DVRDVS-Webs”或“Hikvision-Webs”. “DVRDVS-Webs”和“Hikvision-Webs”充分反映了设备的类型和名称, 因此可以作为海康威视数字视频录像机的 Banner. 在对大量 IP 地址进行主动识别时, 只需校验 HTTP 返回包的“Server”字段是否为“DVRDVS-Webs”或“Hikvision-Webs”便可实现对海康威视网络数字视频录像机的识别.

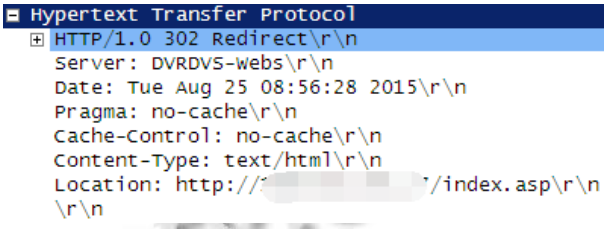

图 2 “Server”字段中的 Banner 信息

3.1.2 HTTP 数据包头部“WWW-Authenticate”字段中的 Banner 信息

当访问某些终端设备的 Web 服务时, 终端设备期望在收到的 HTTP 请求包头部中带有用户的身份认证信息, 这个认证信息就是 HTTP 请求包头部中“Authorization”字段的值. 若终端设备收到的 HTTP 请求包头部中不含有“Authorization”字段, 则终端设备便会发送一个头部中含有“WWW-Authenticate”字段的 HTTP 响应包, 以告知用户需要认证信息. 文献[14]中第 14.47 小节规定了“WWW-Authenticate”字段的具体含义, 一个“WWW-Authenticate”字段值的形式可能如下: WWW-Authenticate: Basic realm="****". “****”这部分内容可能含有对终端设备信息的详细描述.

例如, 在浏览器直接访问一个 TP-LINK 路由器 Web 服务的 IP 地址, 并通过 Wireshark 等抓包软件对 HTTP 数据包进行截获, 可知该 TP-LINK 路由器的 Web 服务返回的 HTTP 数据包头部中“WWW-Authenticate”字段的内容为“Basic realm=‘TP-LINK Wireless N Router WR941ND’”, 如图 3 所示. “TP-LINK Wireless N Router WR941ND”可以明确作为“TP-LINK TL-WR941ND 型 450 兆无线路由器”的Banner信息. 该Banner信息中包含的终端设备信息更加充分, 包括 TP-LINK 路由器的类型(Wireless N Router)及具体型号(WR941ND).

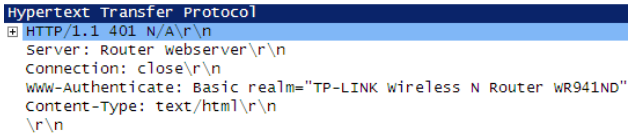

图 3 “WWW-Authenticate”字段中的 Banner 信息

3.1.3 HTML 源代码中的 Banner 信息

某些终端设备 Web 服务返回的 HTML 源代码中直接包含了大量的 Banner 信息, 这些 Banner 信息通常以某种关键词组合的型式出现. 通过分析发现, 关键词组合一般可分为“终端设备类型”与“型号”两部分. 例如, 一个 D-LINK 高速以太网打印机服务器 Web 服务返回的 HTML 页面源代码中包含了“Server Name”、“Model”、“Firmware Version”、“MAC Address”、“IP Address”、“Up Time”、“Printer Name”、“Printer Status”、“DP-300U”等关键词. 其中, “Printer Name”与“Printer Status”表明了终端设备的类型为打印机服务器, “DP-300U”则为打印机服务器的具体型号, 如图 4 所示.

```
31    <tr><td>Model</td><td>: DP-300U</td></tr>
50    <tr><td></td><td>Printer Name</td><td>: PS2P2</td></tr>
51    <tr><td></td><td>Printer Status</td><td>: Off line</td></tr>
```

图 4　HTML 页面中的 Banner 信息

在对大量 IP 地址进行主动识别时, 只需通过正则匹配的方式依次校验返回的 HTML 源代码中是否包含某种终端设备的“设备类型关键词”和“型号关键词”便可实现对终端设备的识别. 利用 HTML 源代码中的 Banner 信息进行终端设备识别时, 关键词的个数会影响识别准确率. 当关键词个数过少时, 可能会造成误判; 而当关键词个数过多时, 则有可能造成遗漏.

3.2 终端设备 Web 指纹识别

Web 指纹, 又指 HTTP 指纹, 即为一个 HTTP 报文中所包含的能够间接反映出 Web 服务器特有信息的内容. 主要包括 HTTP 数据包大小分布、HTTP 数据包时序间隔以及 HTML 页面的 MD5 值及内容关键词等. 利用 HTTP 数据包时序和 Web 服务器软件实现差异的识别方法因差异特征受限、识别对象单一而不适用于终端设备数量繁多的网络环境, 因此, 本文主要从终端设备 Web 服务中“文件”的角度探索识别方法.

3.2.1 图片文件比对

某些终端设备为了提升 Web 界面的美观程度而加入了图片元素, 图片元素通常用作设备厂商的 LOGO、Web 界面背景等. 本文将终端设备 Web 界面中的图片元素所包含的信息归纳为四个属性, 即为 “图片名称”、“图片格式”、“图片路径”和“图片内容”. 不同图片元素所对应的各种属性均不相同, 因此, 根据终端设备 Web 服务中图片文件的差异能够识别出不同的终端设备.

例如, D-Link DFL-800 和 DFL-860 系列防火墙的 Web 界面中都加入了一个“锁”的图片元素, 如图 5 所示. 该图片元素的图片名称为“PadLock”, 图片格式为“可移植网络图形格式(Portable Network Graphic Format, PNG)”, 图片路径为“/images/misc/PadLock.png”. 在进行大规模主动识别时, 请求 IP 地址+图片路径, 并通过响应状态是否为“200 OK”来验证图片元素的“图片名称”、“图片格式”和“图片路径”属性; 通过计算响应内容的 MD5 哈希值来验证“图片内容”属性. 当所有属性均与指纹库中某终端设备指纹吻合时, 则可确定终端设备识别成功.

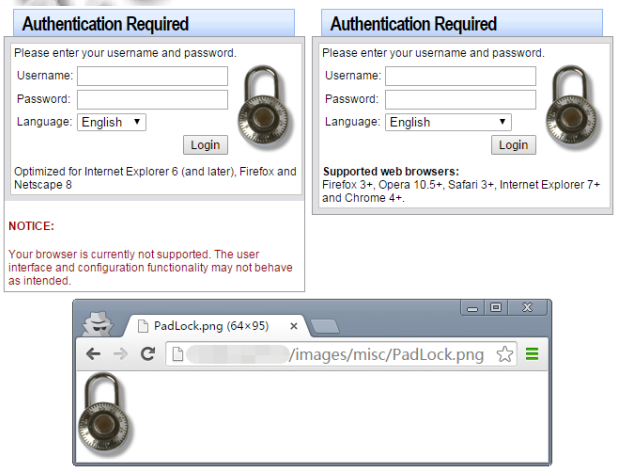

图 5　DFL-800 和 DFL-860 系列防火墙的“锁”图片元素

3.2.2 文本文件比对

对于 Web 界面不含有图片元素的终端设备, 则可用文本文件比对的识别方法. 客户端接收到 Web 服务返回的 HTML 源代码中, 包含了终端设备 Web 服务界面全部的布局、样式和功能等信息. 对于型号完全相同的终端设备, 其 Web 服务界面的布局、样式和功能完全一致, 即客户端所得到的 HTML 源代码完全一致. 因此通过 HTML 文件比对同样能够对终端设备进行识别.

本文对 D-link DFL 系列 6 种不同防火墙设备进行了探究, 分别对其发送正常的 HTTP 请求, 通过分析得到的 HTML 源代码发现, 这 6 种防火墙设备可分为 3 组, 分别为英文版(3 个)、英文版+俄文版(2 个)、英文版+中文版(1 个). 由于加入了不同的语言支持, 得到的 HTML 源代码亦不相同. 进一步研究发现, 对于上述同一组设备, HTML 源代码中仅存在一些细微的

差异, 例如有的HTML源代码中会多出一些空白符(空格符、制表符、换行符)等. 空白符不影响 HTML 源代码功能的实现, 这些细微差异是由于设备的次版本或出场批次不同而造成的. 在去除了空白符后, 同一组防火墙设备返回的 HTML 源代码则完全一致.

分别对上述 6 种防火墙设备返回的完整 HTML 源代码进行 MD5 哈希运算, 结果如表 1 所示; 同时对去除空白符后的 HTML 源代码再次进行 MD5 哈希运算, 结果如表 2 所示. 由表 1 可知, 6 种防火墙设备返回的完整 HTML 源代码的 MD5 运算结果均不相同, 虽然同一组设备的差异非常细微, 但是由于MD5哈希运算的“雪崩效应”, 导致运算结果完全不相同; 由表2可知, 在对 HTML 源代码进行了去除空白操作后, 同一组设备得到的 MD5 哈希运算结果相同. 因此, 通过上述研究和分析, 本文认为通过文本文件比对的方法能够有效的识别出网络空间终端设备, 并且通过对完整HTML源代码和去除空白符后的HTML源代码进行两次MD5运算, 能够有效的区分设备的主版本和次版本以及生产批次.

表 1 完整 HTML 源代码 MD5

设备	MD5
英文版 1	a1667901437c68312ec4817b5d9cfb3b
英文版 2	777c44f3ca2afe9bf9360142e7afd4a4
英文版 3	877b43596a7048b069df1c337ce9225c
英文版+俄文版 1	0d362bb167ca86f467f4f43f4254071e
英文版+俄文版 2	974133c46c4c44b01ea2e523b1ffca87
英文版+中文版	be051d0c4a26efa083ef88fb417b8db7

表 2 无空白符 HTML 源代码 MD5

设备	MD5(去除空白符)
英文版 1	0b680f5d5e8c25dc1f6df1325e59b92c
英文版 2	0b680f5d5e8c25dc1f6df1325e59b92c
英文版 3	0b680f5d5e8c25dc1f6df1325e59b92c
英文版+俄文版 1	6cb5322134b80993b494385d7915f04b
英文版+俄文版 2	6cb5322134b80993b494385d7915f04b
英文版+中文版	e9beb3a98625465641c23687fd34084d

4 讨论

本文提出的识别框架, 在识别网络空间终端设备时具有一定创新性. 与传统的识别体系相比, 本文采用了双重因素的识别体系, 增强了识别框架的可靠性. 相比文献[6]仅从“Server”字段获取 Banner 信息, 本文提出的框架能够从 HTTP 头部各字段和 HTML 源代码中识别终端设备的 Banner 信息, 识别范围覆盖了一次完整的HTTP响应过程. 文献[10]中提出的识别思想为: 对于一种畸形请求, 不同的 Web 服务器可能会按照几种不同的方式处理, 由此可以识别出几种类型的 Web 服务器; 通过构造多种不同的畸形请求, 则能够识别出多种类型的 Web 服务器, 理论值为不同处理方式数量的乘积. 但是这种处理方式差异是有限的, 想要达到较好的识别效果, 就需要加入较多的特征项目, 因此就需要更多的识别时间. 当不同的终端设备对畸形请求作出相同的响应时, 此种识别方法就会出现误判. 本文提出的识别方法利用文件特征作为 Web 指纹, 保证了不同的终端设备不会作出相同的响应, 极大的降低了误报的可能性, 同时文件特征可以扩展至 JS 文件、CSS 文件等, 增强了识别框架的扩展性.

本文提出的识别框架能够完全适用于开放Web服务的终端设备: 首先, Banner 信息的获取范围覆盖了完整的 HTTP 响应, 包括 HTTP 头部和内容. 其次, 图片文件 Web 指纹适用于存在图片元素的终端设备, 且四种图片属性保证了指纹的可靠性; 文本文件为 Web 服务的基本要素, 几乎所有的 Web 服务都含有文本文件(HTML 文件、JS 文件、CSS 文件等), 因此文本文件 Web 指纹的适用性更为广泛. 对于不含图片元素的终端设备, 文本文件 Web 指纹也能够达到较好的识别效果. 同时, 该框架使用的所有探测均使用正常的 HTTP 请求, 不使用畸形请求、超长 URL 等探测方式, 确保了探测报文不被拦截, 不会引发探测对象出现异常.

5 结语

网络空间中终端设备的数量一直呈不断增长的趋势, 因此, 终端设备识别技术将会是一种非常实用而且必需的技术. 该技术对于全面统计网络空间中的设备分布情况以及预警设备潜在威胁均具有很强的实用意义. 基于网络指纹一般模型和双重因素识别技术的理论支持, 本文提出的网络空间终端设备“Banner+Web 指纹”双重因素主动识别框架, 充分考虑了终端设备类型复杂、数量繁多的特点, 通过从“Server”字段、“WWW-Authenticate”字段以及 HTML 源代码中深入挖掘Banner信息的办法解决了识别对象固定、识别精度受限等传统识别方法所具有的局限性. 同时, 双重因素识别弥补了单一因素识别技术中的缺

点, 当终端设备 Web 服务没有暴露出任何 Banner 信息时, 以图片文件和文本文件的 MD5 运算结果等作为 Web 指纹, 同样能够达到较好的识别效果. 当然, 该框架还存在一些不足和待优化之处, 比如目前所针对的识别对象均为开放 Web 服务的终端设备, 而对于只能通过 Telnet、SSH 等其他方式交互的终端设备, 还需要进一步发现适用的网络指纹; 对于如何将“双重因素”优化为“多重因素”, 进一步提高终端设备识别效率, 还需要进行大量的研究, 这也是本文接下来的研究重点.

参考文献

1 海康威视针对“设备安全”的说明. http://www.hikvision.com/cn/news_detail_63_i1273.html. 2015.

2 关于多款 D–LINK 路由器产品存在后门漏洞的情况通报. 2013. http://www.cert.org.cn/publish/main/9/2013/20131025152943288740930/20131025152943288740930_.html.

3 韩智,刘昌平.基于多种特征融合的指纹识别方法.计算机科学,2010,37(7):255–259.

4 何聚厚.网络伪装模型研究.计算机工程与应用,2008,44 (3): 10–13.

5 Fyodor. Remote OS detection via TCP/IP stack fingerprinting. Phrack Magazine, 1998, 17(3): 1–10.

6 Shah S. An introduction to HTTP fingerprinting. 2004. http://net-square.com/httprint_paper.html.

7 Abdulla AAA.击败 HTTP 指纹识别技术[硕士学位论文].长春:吉林大学,2012.

8 ServerMask Web Server Anonymization. http://port80software.com/products/servermask. 2015.

9 Lee D, Rowe J, Ko C, et al. Detecting and defending against Web-server fingerprinting. CSAC 2002: 2002 Computer Security Applications Conference. United States: IEEE Computer Society, 2002: 321–330.

10 杨可新,鞠九滨.利用 Web 指纹进行服务映射.计算机工程与应用,2004,40(4):7–9.

11 王永杰,鲜明,王国玉,肖顺平.基于指纹分析的 Web 服务探测技术.计算机工程,2005,31(17):26–28.

12 唐彰国,李焕洲,钟明全,张健.远程主机网络指纹模型研究.计算机工程与设计,2011,32(8):2592–2595.

13 隋新.主机特征信息被动识别的研究与实现.科学技术与工程,2013,13(3):652–658.

14 Gettys J, Mogul J, Frystyk H, et al. Hypertext Transfer Protocol(HTTP). https://tools.ietf.org/html/rfc2616. 1999.

15 罗青林,徐克付,臧文羽,刘金刚.Wireshark 环境下的网络协议解析与验证方法.计算机工程与设计, 2011,32(3):770–773.