文章编号:1673-0062(2016)01-0066-05

高性能操作系统检测方法研究

陈 军,万亚平,陈 虹,饶 婕*

(南华大学 计算机科学与技术学院,湖南 衡阳 421001)

摘 要:随着网络技术的发展,网络信息系统逐渐深入到各行各业,网络安全成为人们关注的话题.入侵检测是一种用于检测计算机网络系统中入侵行为的网络信息安全技术,通常黑客们利用操作系统指纹识别工具检测计算机系统,并以此分析漏洞和各种安全隐患,大肆攻击入侵网络服务器与私人计算机.因此,本文全方位地介绍 P0F 及 NMAP 两种操作系统识别工具,深入研究分析他们的工作原理,通过对比分析 P0F 与 NMAP 的功能及性能,利用 P0F 的高性能、准确性,对 P0F 源代码进行研究修改,设计出高性能、高准确性的操作系统检测工具,以满足大数据环境下准确获取操作系统信息的需求.

关键词:入侵检测;操作系统检测;系统指纹识别;P0F;NMAP;socket 编程

中图分类号:TP393.08 **文献标识码**:B

Research on High-Performance Operating System Detection Method

CHEN Jun, WAN Ya-ping, CHEN Hong, RAO Jie*

(School of Computer Science and Technology, University of South China, Hengyang, Hunan 421001, China)

Abstract: Currently, network security and information security problems have become increasingly serious. Intrusion detection is a network of information security technology to detect computer network system for the intrusion, hackers generally use the operating system fingerprinting tools to detect computer system, and thus analyze vulnerabilities and security risks, attacking the network server and personal computer intrusion. Therefore, this article introduces comprehensively two operating system identification tools of P0F and NMAP, researches and analyses their works in depth through comparative analysis of the functions and performance of P0F and NMAP, uses P0F performance and accuracy to modify the source code of P0F, design high-performance, high-accuracy operating system detection

收稿日期:2015-11-15

基金项目:湖南省研究生科研创新基金资助项目(CX2015B404);湖南省学位与研究生教育教学改革研究课题基金资助项目(JG2010B037);南华大学教改课题基金资助项目(2014XJG24)

作者简介:陈 军(1990-),男,湖南新田人,南华大学计算机科学与技术学院硕士研究生.主要研究方向:网络安全、信息安全、嵌入式系统. * 通讯作者.

tool, in order to satisfy the needs of accurate obtaining the operating system information in large data environments.

key words: intrusion detection; operating system detection; fingerprint identification systems; P0F; Nmap; socket programming

0　引　言

随着网络技术的迅猛发展,今天的 Internet 已经从各个方面改变着人们的工作与生活方式,它已经成为我们生活中必不可少的一部分[1].与此同时,黑客们利用网络漏洞大肆攻击入侵网络服务器与私人计算机,使网络安全与信息安全问题日益严峻.因此,为了更好的保护网络不受黑客的攻击,就必须对黑客的攻击方法、攻击原理及过程有详细而深入的了解[2].然而,网络扫描是黑客攻击的第一步,他们利用一些网络探测工具对远程计算机进行扫描,其目的就是精确地判别出远程目标主机操作系统的版本及类型、查出正在监听的端口等等,从而有针对性地对其实施攻击[3].

本文深入研究操作系统指纹识别工具 P0F 与 NMAP,对比两种不同操作系统指纹识别工具在性能、准确性及网络资源使用情况;通过对结果进行分析,利用 P0F 的高性能、高准确性设计出更高性能、适用于大数据环境下的操作系统指纹识别工具.

1　Nmap:主动扫描

NMAP 是最流行、最精确的网络映射及操作系统检测工具之一[4].它的工作原理是将特制的数据包发送至目标主机,然后根据响应数据包判定远程主机操作系统信息,NMAP 被称为是一个准确的 OSD(Operating System Detection)工具,它具有区分操作系统版本间较小差异的能力,用它来扫描网络需要每个主机响应数据包,根据响应数据包判断远程主机的信息,但扫描一个大的网络需要耗费很长时间[5].使用 NMAP 或任何其它活动扫描产生的额外流量还占用带宽,这将会导致网络拥塞情况,如此频繁的网络扫描将非常不方便.

虽然 NMAP 是准确的,但是有些种情况下,NMAP 无法检测机器的操作系统.NMAP 至少需要一个开放端口,以准确地执行操作系统扫描,当一台机器在网络上不接受传入连接的情况下,NMAP 将不能够确定机器的操作系统.网络地址转换(NAT)设备能防止 NMAP 的正常工作,如果没有办法来解决一个 NAT 内部的设备,NMAP 就不能扫描装置.NMAP 的主动性也意味着,防火墙和 IDS(Intrusion Detection Systems)能够检测 NMAP 发送的数据包,并阻止它[6].

图 1.NMAP 扫描主机测试是根据 192.168.2.0/24 网络(254 个 IP 地址,在线 84 台主机)进行 NMAP 扫描测试结果所绘,图中横轴为 NMAP 扫描一定数量 IP 地址所花时间(单位:s)描的 IP 地址个数(菱形点曲线表示扫描的范围,即 IP 地址的个数,方形点曲线表示扫描到的在线主机数,即在所扫描的范围内在线的主机总数).根据测试结果可以发现,随着扫描主机范围增加,所耗费的时间增加,主机扫描速率大概为 150 个 IP 地址每秒,对于大数据环境下,处理数据的速度远远达不到要求.

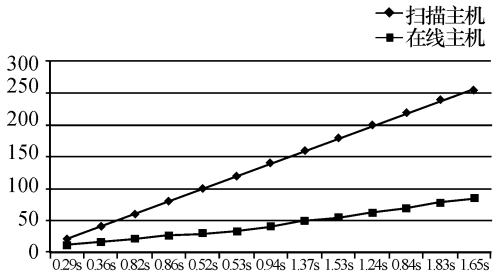

图 1　NMAP 扫描主机测试

Fig.1　NMAP scan host test

2　P0F:被动分析

被动检测工具解决了大部分主动检测工具所不能解决的问题[7].被动检测工具最大的优点是在主动检测工具检测失败的情况下能够正常工作,这是由于它是根据对远程主机发送过来数据包进行解析,从而判断远程主机的操作系统,不会主动发送数据包给远程主机,因此不会耗费额外流量导致网络堵塞,也不会由于防火墙、端口关闭或是在 NAT 环境下失败.被动检测工具对于所有的网络通信都是可检测的,它不必依赖于远程主机必须回应相应的数据包而得到正确的信息[8].

P0F 是一个被动检测工具,它能够检测出远程主机使用 WEB 浏览器类型、网络适配器类型、远程主机是否处于 NAT 环境等等[9].P0F 的运行机制是通过解析网络连接中的 TCP 握手数据包分析远程

主机的信息,在正常运行 P0F 的环境中,远程主机发起 TCP 连接,P0F 将根据连接数据包获取远程主机的操作系统信息.像其它被动检测工具一样,P0F 能够识别出远程主机是否处于 NAT 或防火墙后,但前提是必须要是 TCP 数据包才能进行识别.然而,如果远程主机未使用 TCP 或者传输层数据进行了加密,P0F 也是无法工作的[3].

P0F 能够识别远程主机的 OS 类型,是由于不同的 OS,进行 TCP 连接发送 SYN 或 SYN+ACK 包时,虽然 TCP 包的头部信息结构一致,但是 TCP 选项字段的信息不是固定的,这些数据会由于 OS 的不同而有所变化.P0F 获取 9 个字段(IP 版本,生存时间,IP 选项字段长度, 最大段长度, 窗口大小, 窗口因子,TCP 选项的顺序, quirks,是否为 TCP Payload data 等)信息,这 9 个字段足够区分远程操作系统的类型[10].

2.1 P0F 抓取数据包测试情况

在局域网 200 多台在线主机的环境中,通过在连接外网网络的主机中运行 P0F 操作系统指纹识别工具一天时间(2014/12/03 19:29:28 到 2014/12/04 19:39:53),抓取所有经过测试主机的数据包(总共 4504684 条),通过分析 P0F 识别结果进行统计得到如表 1.

表 1 P0F 数据包百分比

Table 1 P0F packet percentage

数据包类型	指纹数据条数	所占百分比/%
SYN	1 983 875	44.040 3
SYN+ACK	137 849	3.060 1
MTU	2 121 724	47.100 4
UPTIME	115 288	2.559 3
HOST CHANGE	122 992	2.730 3
IP Sharing	19 200	0.426 2
HTTP Request	1 886	0.041 9
HTTP Response	1 870	0.041 5

通过表 1 数据绘制图 2.根据实际的情况进行对比分析,其中涉及 241 个 IP 地址(内网段地址 211 个,IPv6 地址 25 个,其它移动设备 IP 地址 5 个),针对数据进行分析排除未识别的操作系统后,经过对比发现有 10 台 IP 地址发生变化识别有误外,其它都能成功识别,识别率达(231/241)95.85%,P0F 的操作系统识别率很高.

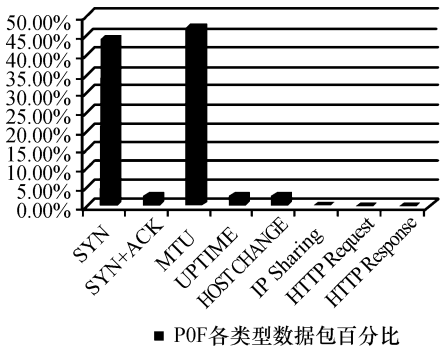

图 2 P0F 数据包百分比

Fig.2 P0F packet percentage

2.2 P0F 性能测试

针对 P0F 高识别率,对 P0F 进行空白的性能测试,通过模拟发送数据包进行性能测试.根据测试结果,绘制成图 3 所示.

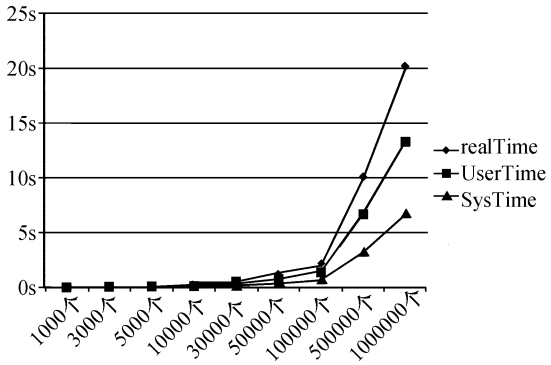

图 3 单实例 P0F 性能测试

Fig.3 Single instance of P0F performance test

图 3 单实例 P0F 性能测试中横轴表示模拟发送数据包的个数,纵轴表示 P0F 识别指定数目数据包所耗费时间(单位:s),不同的曲线表示不同的时间(realTime、UserTime、SysTime 分别应用系统实时时间、用户时间、系统时间).由上可以看出,P0F 每秒处理 50000 个数据包,一个包大小假设为 64 字节,得到 50000 * 64 * 8 / (1024 * 1024) = 24.4 Mbps,对于千兆网卡来说如果网络上都是 sync 包,理论上最多起 40 个 P0F 实例就可以处理完成,这种情况仅仅存在泛洪攻击的条件下,而实际的网络中,10 个程序处理千兆网卡的 sync 包数据绰绰有余,根本不会存在有堆积的情况.

3 NMAP 与 P0F 对比分析

根据以上 NMAP 与 P0F 的测试,通过对比两种操作系统指纹识别工具在性能、准确性、网络带

宽资源方面的情况,在处理数据方面 P0F 占据优势,对于应用被动检测操作系统的情况下,P0F 绝对是不二选择.然而,P0F 处理数据包的速度为每秒 5000 个数据包,在大数据环境下,使用 P0F 处理数据包也会产生大量数据包堆积.现有的工具及解决方案,无法解决大数据环境下实时获取操作系统信息的要求,因此,可利用 P0F 的高性能、准确性加以改进,设计出高效的操作系统检测工具.

4　Multiple-P0F 设计

根据测试发现,NMAP 实时处理数据能力有限,特别是 NMAP 为主动扫描工具,需要发送及接收数据,会占用网络中的大量带宽资源,可能会导致整个网络阻塞.然而,P0F 能够实时处理大量的数据,准确性及效率高,且不会占用网络中的带宽资源.即便如此,处于大数据环境下,P0F 将会导致大量数据堆积,无法实时响应数据包;为解决此场景,通过分析、研究,设计出以下方案来解决大数据处理问题,如图 4.

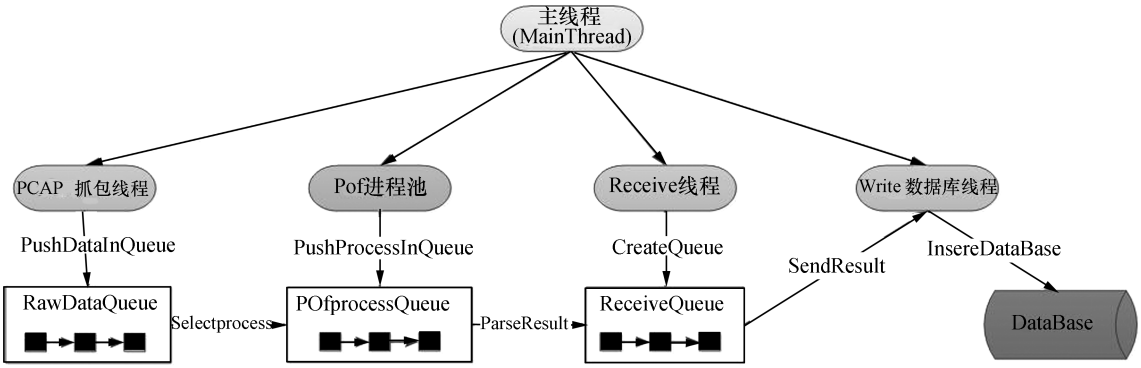

图 4　Multiple-P0F 总体设计

Fig.4　Multiple-P0F overall design

根据图 4 Multiple-P0F 总体设计,通过对 P0F 源代码深入研究,并根据实际情况进行改进,仅用于处理 PCAP 抓取的数据包.Multiple-P0F 在主线程启动后,创建三个线程(PCAP 抓包线程,Receive 线程,Write 数据库线程)及 P0F 进程池(创建多个 P0F 处理数据包实例,可根据具体情况进行调整),然后 PCAP 抓包线程专门用于抓取网络数据包,并将抓取到的原始数据包(数据链接层数据包),发送给 P0F 进程池中的其中一个实例处理,P0F 将解析完得到的结果数据包发送至 Receive 线程,Receive 线程经过一定处理后,发送最终结果给数据库线程,数据库线程负责将得到的结果数据写入数据库,以供统计及结果分析.

其中,抓包线程与 P0F 进程池之间通过本地 Socket(UDP 方式)的方式进行通信;PCAP 抓包线程抓取到一个原始数据包后,生成一个 36 位的随机序列号(此序列号 Receive 接收线程用于处理原始数据包与 P0F 解析后得到结果进行匹配),然后通过随机选取 P0F 进程池中的一个实例进行处理.P0F 进程实例处理完之后,同样通过本地 Socket 的方式将结果(包括 36 位的序列号)发送给 Receive 接收线程,Receive 接收线程根据序列号,找到原始数据包,将原始数据包与解析后的结果发送给 Write 数据库线程,Write 数据库线程将结果写入数据库,完成数据的保存及后续用途;需要注意的是,为了处理的方便,PCAP 线程、P0F 进程池、Receive 线程都使用队列进行顺序处理.

4.1　测试结果

根据 Multiple-P0F 的设计实现其功能,并将 Multiple-P0F 与 P0F 的测试结果绘制得到图 5.其中横轴是 P0F 与 Multiple-P0F 抓取到数据包的个数,纵轴是处理完一定数量数据包所耗费时间,realTime、UserTime、SysTime 分别为 P0F 处理数据包耗费的实时时间、用户时间、系统时间曲线;而加-M 选项是与 P0F 对应的 Multiple-P0F 的曲线(在同时开启 5 个 P0F 进程的情况下,得到的测试结果).

从图 5 Multiple-P0F 性能测试可以看出,通过 Multiple-P0F 处理数据包明显比 P0F 效率高,在

Multiple-P0F 开启 5 个 P0F 进程情况下,效率提高 4 倍左右,并且都是以相同方式处理,准确性与 P0F 一致.因此,根据图 5 数据包个数与耗费时间的对应关系计算(根据 2.2 节 P0F 的空白性能测试可知,每秒处理 5000 个数据包,一个包大小假设为 64 字节,50000 * 64 * 8 / (1024 * 1024) = 24.4 Mbps),在大数据环境下开启 10 个 P0F 进程,可以实时获取所处环境的所有主机操作系统信息,不会产生数据堆积的情况.

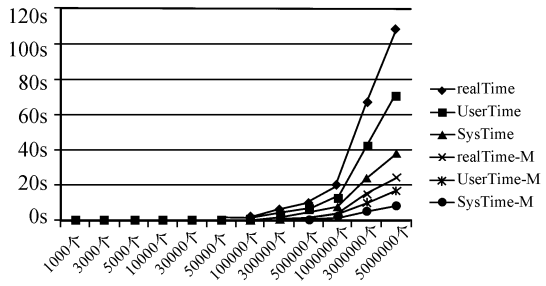

图 5 Multiple-P0F 性能测试

Fig.5 Multiple-P0F performance test

4.2 结论及不足

本文通过深入研究 P0F 与 NMAP 两种操作系统指纹识别工具,把两者在性能、准确性及网络拥塞等情况进行对比分析,总结出在网络数据流较大的情况下,P0F 在性能及准确性方面比 NMAP 有很大优势,且 P0F 属于被动检测工具,不会造成网络拥塞,因此在处理大规模数据时,P0F 是首选.

然而,在处理超大规模数据量的情况下,通过测试及源代码分析可知,P0F 也是力不从心.因此,利用 P0F 高性能、高准确性的优势,被动检测不会导致网络拥塞的特点,改进并设计出处理大数据环境下高性能的检测操作系统方案,通过测试在正常的大数据规模情况下,不会出现数据包来不及处理而堆积的情况,完全满足实时获取远程主机操作系统信息.

Multiple-P0F 的设计方案在一定程度上解决了大数据环境下实时获取远程主机 OS 信息,但有一定的局限性.首先远程主机必须使用 TCP 进行网络通信且未对传输层数据进行加密;其次,在运行 Multiple-P0F 的主机,必须能够获取远程主机通过 TCP 通信的网络数据包,能够对数据包进行处理,二者缺一不可.

参考文献:

[1] 李慧慧.一种基于多线程机制的端口扫描器的设计与实现[D].太原:太原理工大学,2010.

[2] 赵旭.Snort 网络入侵检测系统的研究与改进[D].西安:西安电子科技大学,2007.

[3] 邬书跃.基于支持向量机和贝叶斯分析技术的入侵检测方法研究[D].长沙:中南大学,2012.

[4] Gagnon F, Esfandiari B. A hybrid approach to operating system discovery based on diagnosis theory[J]. IEEE Network Operations and Management Symposium, 2012, 131(5):860-865.

[5] Shu G, Lee D. A formal methodology for network protocol fingerprinting[J]. Parallel & Distributed Systems IEEE Transactions on, 2011, 22(11):1813-1825.

[6] 杨理文.基于渗透测试的网络安全评估技术研究[D].长沙:国防科学技术大学,2011.

[7] 王忠.基于 Snort 的入侵检测应用系统设计与实施[D].成都:电子科技大学,2008.

[8] Beverly R. A robust classifier for passive tcp/ip fingerprinting[J]. Springer Berlin Herdelberg, 2014, 3015: 158-167.

[9] Szewczyk P, Valli C. Personal firewalls-testing robustness[C]. Valli C. Proceedings of Australian Digital Forensics Conference. Perth, Western Australlia: Edith Cowan University Research Online, 2012:105-112.

[10] Paxson V, Floyd S. Wide area traffic: the failure of poisson modeling[J]. IEEE/ACM Transactions on Networking (ToN), 1995, 3(3):226-244.