Journal of Computer Applications ISSN 1001-9081 2016-03-10
计算机应用,2016,36(3):661 -664,702 CODEN JYIIDU http://www.joca.cn

文章编号:1001-9081(2016)03-661-04 DOI:10.11772/j.issn.1001-9081.2016.03.661

基于网络欺骗的操作系统抗识别模型

曹 旭*,费金龙,祝跃飞
(数学工程与先进计算国家重点实验室(信息工程大学),郑州 450002)
(* 通信作者电子邮箱 xu20101002@126.com)

摘 要:针对传统主机操作系统抗识别技术整体防御能力不足的问题,提出一种基于网络欺骗的操作系统抗识别模型(NDAF)。首先,介绍模型的基本工作原理,由网络内的欺骗服务器制定欺骗指纹模板,各主机根据欺骗模板动态改变自己的协议栈指纹特征,实现对攻击者操作系统识别过程的欺骗;其次,给出一种信任管理机制,依据威胁大小不同,有选择地对外部主机开展欺骗。实验测试表明,NDAF 会给其网络通信带来一定的影响,但所产生的额外开销相对稳定,约为 11% ~15%,与典型的操作系统抗识别工具 OSfuscate 和 IPmorph 相比,NDAF 操作系统抗识别能力较强。所提模型通过网络的一体化、欺骗性防御,能够有效提高目标网络防御水平。

关键词:操作系统识别;网络欺骗;主动防御;欺骗防御
中图分类号:TP393 **文献标志码**:A

Anti-fingerprinting model of operation system based on network deception

CAO Xu*, FEI Jinlong, ZHU Yuefei
(*State Key Laboratory of Mathematic Engineering and Advanced Computing* (*Information Engineering University*), *Zhengzhou Henan* 450002, *China*)

Abstract: Since traditional host operating system anti-fingerprinting technologies is lack of the ability of integration defense, a Network Deception based operating system Anti-Fingerprinting model (NDAF) was proposed. Firstly, basic working principle was introduced. The deception server made the fingerprint deception template. Each host dynamically changed the protocol stack fingerprint according to the fingerprint deception template, therefore the process of operating system fingerprinting by attacker was misguided. Secondly, a trust management mechanism was proposed to improve the system efficiency. Based on the different degree of threat, different deception strategies were carried out. Experiments show that NDAF makes certain influence on network efficiency, about 11% to 15%. Comparing experiments show that the anti-fingerprinting ability of NDAF is better than typical operating system anti-fingerprinting tools (OSfuscatge and IPmorph). NDAF can effectively increase the security of target network by integration defense and deception defense.

Key words: operating system fingerprinting; network deception; proactive defense; deception defense

网络技术的发展,让人们充分享受着各种便利的同时,也使得人们的信息安全面临着严峻的挑战。近年来,网络安全事件不断升级,各种网络攻击技术层出不穷[1]。网络的安全防护得到了人们的广泛关注。Atighetchi 等的研究[2]表明,网络攻击过程中 95% 的时间花费于前期侦察,具体实施网络攻击的时间只占到 5%。因此,破坏或干扰攻击者的前期信息收集,从而阻断或误导后续攻击,是网络防御的有效途径。主机操作系统类型和版本的识别是信息收集阶段的重要环节。从防御者角度来看,干扰攻击者信息收集阶段的相关活动,降低攻击者对己方主机操作系统探测的准确率是一种提高网络防护能力的有效手段。然而,传统的针对主机的操作系统抗识别技术,在当前 APT(Advanced Persistent Threat)攻击泛滥的背景下,容易因所在网络的其他主机防护不严密而被渗透[3]。

对此,本文提出一种基于网络欺骗的操作系统抗识别模型(Network Deception based operating system Anti-Fingerprinting model, NDAF),由网络内的 NDAF 服务器制定欺骗模板,然后各主机根据欺骗模板动态改变自己的协议栈指纹特征,实现对外部攻击者操作系统识别过程的干扰,增加攻击者信息收集分析的难度,提高网络防护水平。其主要优点表现为:1)网络内主机进行统一的欺骗管理,欺骗效果好;2)提出了信任管理机制,依据威胁大小不同,有选择地对外部主机开展欺骗,系统运行效率高;3)考虑到 IPv6 的部署已经全面展开,同时支持 IPv6 和 IPv4 两种 IP 协议栈。

1 相关工作

目前,国内外对操作系统真实信息保护的研究,主要集中在如何防止对主机操作系统类型进行探测方面。从原理上可分为以下两类:

1)数据包过滤类。

此类工具通过过滤某些特定的数据包实现对操作系统探测攻击的阻断。Trefiro 等[4]提出了 Stealth,通过过滤掉一些固定的数据包,例如 TCP 报头的 SYN 和 FIN 同时置位,或 FIN、PSH 和 URG 同时置位等,干扰攻击者对操作系统类型判断的准确性。Blackhole[5]是基于 BSD 平台的过滤器,它能够阻止远程主机对 TCP 或 UDP 关闭端口的探测,由此实现对探测工具影响。官方发行的 OpenBSD 具有特有的 packet filter

收稿日期:2015-08-18;修回日期:2015-10-31。
作者简介:曹旭(1983 -),男,江苏扬州人,博士研究生,主要研究方向:云计算、网络信息安全; 费金龙(1981 -),男,河南开封人,讲师,博士,主要研究方向:网络信息安全; 祝跃飞(1964 -),男,浙江杭州人,教授,博士生导师,博士,主要研究方向:密码学、网络信息安全。

防火墙，允许(通过配置文件 pf. conf)设定系统所发送报文的部分信息(例如 TTL、MSS、IPID 策略等)[8]。

2) TCP/IP 协议栈伪装类。

此类工具通过修改协议栈的某些参数使得对某些数据包的回应数据包携带与真实指纹信息不同的特征，从而欺骗攻击者。Roualland 等[7]提出了 Ip Personality，首先分析 Nmap 探测情况得到各种操作系统的特征，然后以指定的操作系统的指纹特征为模板来修改当前操作系统 TCP/IP 协议栈的网络特征值，进而阻止攻击者探测到真正操作系统的类型。Fingerprint Fucker[9]是一款基于 Linux 内核 2.2 实现的类似工具，与 Ip Personality 原理相同，也是通过分析 Nmap 指纹实现对协议栈参数的更改。Crenshaw 针对 Windows 系统设计了 OSfuscate[9]，通过修改注册表中的键值，实现对 TCP/IP 参数的更改，进而影响 TCP/IP 的回应。马君亮等[10]针对探测工具 Xprobe2 提出了针对性的操作系统伪装工具，通过对主机数据报进行伪装，来干扰 Xprobe2 操作系统指纹探测。Prigent 等提出了 IpMorph[11]，通过设计一个用户模式下的 TCP/IP 协议栈，实现对会话的监视和对数据包的重写。

综上可知，已有方法都是仅从主机防护出发，针对如何防止主机自身遭受攻击开展研究。然而，随着攻击技术的进步和攻击过程的复杂化、持久化，攻击者往往会从目标主机所在的整个网络出发，寻找突破点，进而攻克重要目标主机，因此简单的单机操作系统伪装很有可能无法达到欺骗攻击者的目的。

2 基于网络欺骗的操作系统抗识别模型

本文所针对的场景是：远程攻击者从网络外部发起，利用 IP、ICMP、TCP 和 UDP 协议，对目标网络内的主机进行操作系统探测，以掌握网络内主机的基本信息，为发起后续攻击提供基础。

对此本文提出了基于网络欺骗的操作系统抗识别模型(NDAF)，基本原理如图 1 所示。在目标网络内，部署 NDAF 服务器，由其负责制定欺骗策略，网络内部主机在其动态的、统一的控制下，改变出站数据流的某些参数，由此实现对攻击者前期探测的干扰，提高网络防护水平。

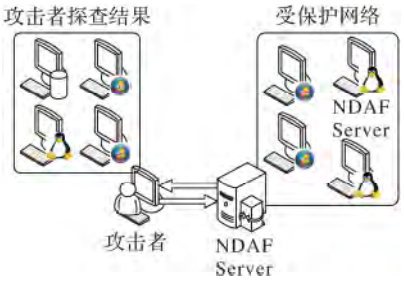

图1 NDAF 基本原理示意图

首先定义 NDAF 中的几个重要概念：

定义 1 欺骗指纹模板(Fingerprint Deception Template，FDT)。包含了两部分内容。首先是一个时间段内各主机对应的需要实施的指纹变换类型。其次，还包括需要客户端直接设定的信任状态表项，例如对己方网络内任何主机实施过攻击的 IP 将被直接设为“不信任”。

定义 2 欺骗指纹库(Fingerprint Deception Database，FDD)。存储常用的操作系统类型欺骗时需要修改的特征。因为远程攻击者可以所使用的识别工具并非单一种，因此欺骗指纹库综合已有的主流的操作系统识别工具来生成。NDAF 在实现时，具体基于 Nmap、SinFP 和 Xprobe 三种主流操作系统识别工具的指纹库来构建己方的欺骗指纹库。

定义 3 NDAF 服务器(NDAF Server)。是 NDAF 的核心。负责根据算法和信任表制定并维护指纹欺骗模板，以及将欺骗模板交付客户端。

定义 4 NDAF 客户端(NDAF Client)。负责向服务器请求指纹欺骗模板，并按照模板修改出站数据流，实现对外部主机的欺骗。

定义 5 NDAF 配置周期(NDAF Configure Period，NCP)。是指 NDAF 客户端重新请求指纹欺骗模板的周期，当配置有效期到期，需重新请求欺骗模板，并按照模板设定重新配置本节点属性等相关信息。

2.1 基本工作流程

系统基本工作流程如图 2 所示。

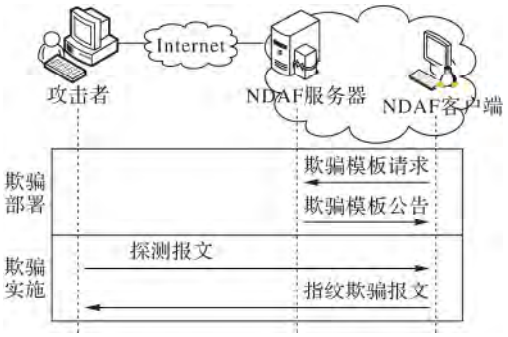

图2 NDAF 基本工作流程示意图

1) 欺骗部署流程。

①以 NDAF Configure Period 为周期，NDAF 客户端周期性地从 NDAF 服务器获取指纹欺骗模板。

②NDAF 客户端根据收到的指纹欺骗模板，设置下一 NDAF Configure Period 内用于进行指纹欺骗的欺骗模板，并将其中包含的高危 IP 加入自己的信任状态表中，设置相应的信任状态表项。

2) 欺骗实施流程。

①外部节点(可能是攻击者)向部署了 NDAF 客户端的内部主机发送了可能是探测报文的数据包。

②NDAF 客户端查找该节点 IP 对应的信任状态，如果是可信则正常回应；如果是“不可信”则根据欺骗模板中自己对应的欺骗配置进行欺骗式回应；如果是“部分可信”则根据其对应的 P 值按概率进行欺骗式回应。

2.2 信任管理机制

NDAF 采用一套独特的信任管理机制来对外部访问受保护网络的 IP 地址进行评价。

1) 信任状态。

NDAF 在相应的模块保存有一个信任状态表。每一个访问受保护网络的外部主机所对应的 IP 地址都会被赋予一个信任状态。信任状态表项为 $\langle IP_A, Tru_A \rangle$。其中 Tru_A 为信任状态，包括以下三种：

①可信(Trust)。说明该地址是普通的、无恶意的外部地址。

②不可信(Untrust)。说明该地址是攻击者的地址，需要对其进行欺骗。

③部分可信(Partial Trust)。说明该地址在一定的时间内无攻击行为，可以在一定程度上认定其无恶意。

2) 信任状态转换。

当外部主机需要与主机进行连接时,如果该连接是其对主机的回应,则将该 IP 初始化为“可信”地址; 如果该连接是主动发起的,则将该 IP 初始化为“不可信”地址。

外部主机 IP 地址的不可信的状态会保持一定的时间,然后随着时间的变化而发生改变。当经过一定的时间之后,如果该主机无任何针对目标网络内任何主机的恶意行为,则该地址状态由“不可信”转换为“部分可信”,然后为其分配一个会随着时间变化的欺骗概率 P。由概率可信地址发起的连接将有 P 的可能进行欺骗回应,以 $1-P$ 的概率直接放行。信任概率 P 并非固定不变的,随着时间的变化,IP 地址的状态会逐渐变化,针对该主机的欺骗概率 P 也会相应地发生改变。

NDAF 客户端会对来自“部分可信”主机的数据包按照概率 P 进行欺骗式回应。

下图解释了信任状态转换情况。

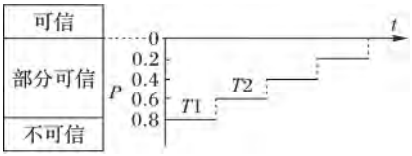

图 3　NDAF 欺骗概率变化示意图

由图可知,外部主机在 $T1$ 时间内,若无恶意行为则被分配的欺骗概率会从 1 逐渐变小,然后再经过 $T2$ 时间,若仍然无任何恶意行为则欺骗概率会继续变小,由此类推。

2.3　NDAF 的欺骗实现

NDAF 客户端核心是对外部探测主机的欺骗。具体实现上是部署于主机之上的一个 daemon 程序,如图 4 所示。其运行原理类似于一个工作于真实网卡和主机网络接口之间的虚拟网卡,以方便 NDAF 在不改变主机协议栈的情况下,能够及时分析和干预任何进出受保护主机的 IP 数据包,且易于设计实现。

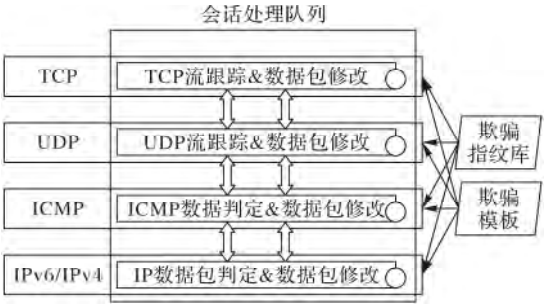

图 4　NDAF 欺骗机制示意图

对于每条数据流,NDAF 会跟踪数据流的状态,维护一条对应的流状态表项,以方便确定数据包归属,以及后续 IP 数据包的分析和处理。NDAF 跟踪的数据流会话,不仅包括 TCP 的流,同时对于 UDP,当双方通信超过 60 s 也认为其存在一个流。

NDAF 在收到数据包后,首先检查 IP 地址,若属于某个已记录数据流,则按照相应的规则直接处理; 若不属于任何已记录流,则检查信任状态表,确立处理规则。在进行修改时,将根据指纹欺骗模板分别按照 IP 报头各字段(IPv6 包含扩展报头) 、TCP/UDP 报头字段、上层载荷 banner 信息的顺序逐层处理。

3　模型安全性分析

本节形式化攻击者对受保护网络进行的操作系统探测过程如下:

受保护网络内部有主机 N 个。NDAF 保护失败的主机数为 M,攻击者一旦发现此种主机则可顺利的开展后续攻击; NDAF 能顺利完成欺骗的主机数为($N-M$) 个。

另假设攻击者已知目标网络的可用地址空间 N,并且连续扫描 n 个主机,$n < N$。

在给定的 n 次机会中,如果攻击者能够找到 1 台 NDAF 保护失败的主机,则认为目标网络防御失败,攻击者成功利用该主机攻陷整个网络。

然后进一步地,将该问题抽象为一个小球投盒问题。

设存在一个放有 N 个小球的盒子,小球分为两种: M 个绿色小球,代表 NDAF 保护失败的主机; 剩下的是红色小球,代表 NDAF 成功保护的主机。攻击者每次从盒子中取出一个小球,共取 n 次。当攻击者取到至少 1 个绿色小球,认为其攻击成功。此外,本文假设攻击者在探测完成前,先不展开攻击。这也符合网络攻击中的一般探测过程。也就是说,即使取到了红色小球依然继续取球,直到取完 n 个小球为止。

在此情况下,攻击者攻击目标网络成功的概率可以看作是一个超几何分布。

进一步地,可得到其分布公式如下:

$$\Pr(X=k)=(C_M^k C_{N-M}^{n-k})/C_N^n;\quad k=0,1,2,\cdots,m$$

其中: $m=\min\{M,n\}$,且有 $n \leqslant N, M \leqslant N(n,M,N \in \mathbf{N}^*)$。

则攻击者成功的概率如图 5 所示。

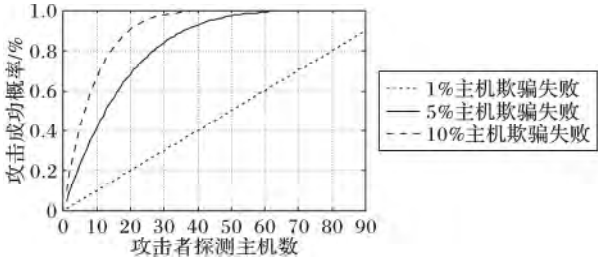

图 5　攻击者不同探测强度下的攻击成功率

由图 5 可知,随着攻击者探测的主机数目增多,攻击者的攻击成功率随之上升。此外,攻击者攻击成功率与欺骗能力也有关系,操作系统指纹欺骗成功率越高,攻击者的攻击成功率相对越低。

4　实验分析

本文设计并实现了 NDAF 的原型系统,其中 NDAF 客户端在 Windows 7 和 Ubuntu 12.04 两种操作系统上进行了开发和部署,NDAF 服务器程序在 Linux Ubuntu 12.04 系统上进行了开发和部署。

为了验证 NDAF 的实际性能,构建测试环境进行实验分析。测试环境如图 6 所示,主要由同一路由器连接的两个子网(201024 :: /64 和 2013 :: /64) 组成。其中: 2013 :: /64 为使用 NDAF 进行安全防护的网络,2010 :: /64 为外部网络,其中部署有 1 台攻击者主机。系统的 NDAF 配置周期设定为 30 min。

在实际部署时,受保护网络中的节点有可能是向外提供 Web、FTP 等服务的服务器,因此 NDAF 的部署必须考虑对节点网络通信效率的影响。因此,本节首先分析 NDAF 给节点通信带来的额外时间开销。节点 A 已知晓攻击者 X 的身份,然后在 NDAF 保护下,与攻击者主机 X 进行通信,节点 A 处

记录传输时间开销，然后停止 NDAF 保护，传输同样大小的测试文件，记录传输时间进行对比，结果如表 1 所示。

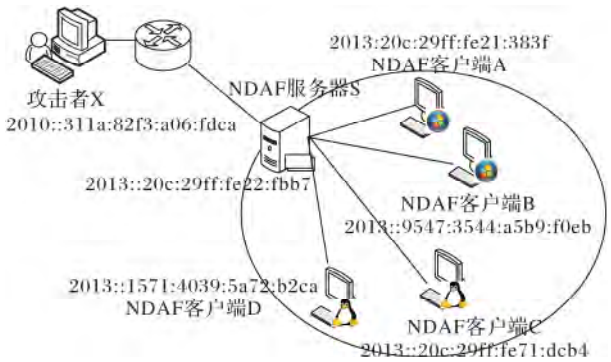

图 6 NDAF 实验环境拓扑结构示意图

表 1 NDAF 对节点通信时间的影响 s

测试文件情况	未启用 NDAF 保护	启用 NDAF 保护
HTTP(0.5 MB)	0.13	0.17
HTTP(5 MB)	0.47	0.58
FTP(0.5 MB)	0.21	0.29
FTP(5 MB)	0.64	0.77

由实验结果可知，网络内节点启用 NDAF 会给其网络通信带来了一定的影响，但所产生的额外开销相对稳定，约为 11% ~15%。这是由于在启用 NDAF 后，为修改网络数据流含有的操作系统特征，需要对输出的网络数据包进行修改，因此会产生一定的额外时间开销。

然后，将 NDAF 与目前典型的操作系统抗探测工具 OSfuscate(适用于 Windows 平台)和 IPMorph(适用于 Linux 平台)进行对比测试。首先，在主机 B 上部署 Windows 系统的虚拟机，接着随机选定目标操作系统，分别用每种抗探测工具保护，然后用 NMAP、Xprobe2 和 SinFP 进行探测，共测试 20 次，记录测试结果。接着，采用同样的过程，针对部署于主机 B 上的 Linux 系统进行测试，对比 NDAF 和 IPMorph 的抗识别能力。两次对比测试的结果如表 2 所示。

表 2 与典型操作系统抗识别工具的对比测试结果

测试序号	抗识别工具	NMAP	Xprobe2	SinFP
1	OSfuscate	15	17	17
	NDAF	19	20	20
2	IPMorph	18	19	18
	NDAF	19	20	19

由表 2 可知，NDAF 与典型的操作系统抗识别工具 OSfuscate 和 IPMorph 相比，抗识别能力较强。而且，即使是 NDAF 的几次欺骗失败中，操作系统识别工具 NMAP 和 SinFP 虽然无法将其识别为欺骗目标类型，但也未能将其识别为真实的操作系统。而 OSfuscate 和 IPmorph 则出现了完全欺骗失败(操作系统识别工具识别出真实操作系统)的情况。

最后，测试 NDAF 的在启用信任管理机制后的操作系统欺骗效果。首先不采用 NDAF 保护，由 nampSI4 结合 NMAP6.40 对受保护的网络进行探测，探测所得结果如图 7 所示。

由图 7 可以看到，在未启用 NDAF 对主机进行保护时，攻击者通过 nmap 所获得目标网络内主机的操作系统信息与真实情况相符。

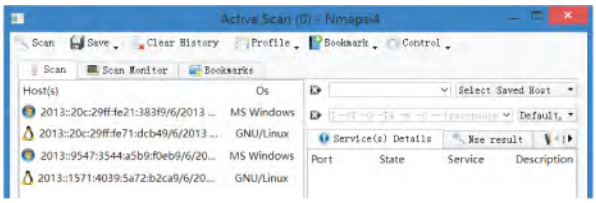

图 7 启用 NDAF 前攻击者操作系统探测结果

接着主机启用 NDAF 对自己进行保护，由攻击者 X 再次使用 NMAP 对目标网络进行扫描，探测所得结果如图 8 所示。

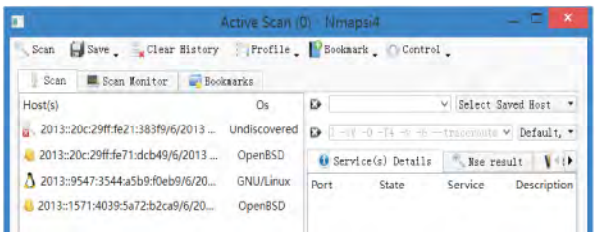

图 8 启用 NDAF 后攻击者操作系统探测结果

由图 8 可知，当启用 NDAF 对目标网络进行保护后，攻击者通过探测，仅能得到 NDAF 展示的虚假信息。也就是说，NDAF 能够有效的欺骗攻击者，从而提高目标网络的网络防御水平。

接着，关闭 NDAF 的 IDS 日志接收功能(即停止收集恶意行为统计情况)，测试 NDAF 的信任管理机制。在相隔 30 min 和 1 h 后，再一次使用 NMAP 进行探测，结果如图 9 和图 10 所示。

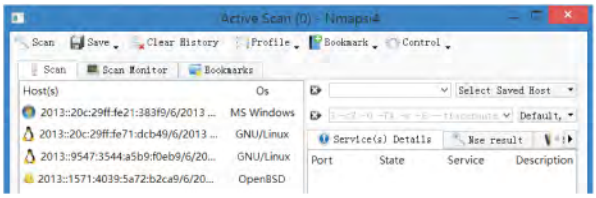

图 9 启用 NDAF 30 min 后攻击者操作系统探测结果

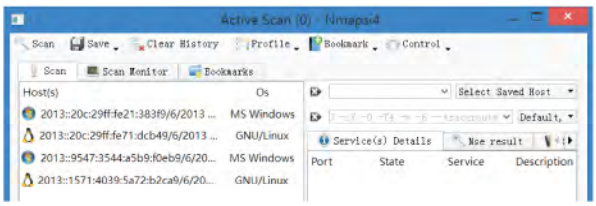

图 10 启用 NDAF 1 h 后攻击者操作系统探测结果

由图 9 和图 10 可知，30 min 后，NDAF 只是进行一定程度的欺骗，结果中有两台主机的操作系统类型被正确探明。1 h 后，已全部停止对节点 X 进行欺骗。也就是说，当启用 NDAF 对目标网络进行保护后，在外部主机无恶意行为的情况下，NDAF 对其实施欺骗的强度逐渐降低。由于实施欺骗需要对数据流进行修改，影响网络通信效率，因此动态地调整欺骗强度将有利于保障系统的整体运行效率。

5 结语

预先防御、主动防御是当前网络防御的重要研究方向。破坏或干扰攻击者的前期信息收集，从而阻断或误导后续攻击，是实现网络预先防御、主动防御的有效途径。本文针对攻击者前期信息收集过程中的操作系统探测，提出一种基于网络欺骗的操作系统抗识别模型的对抗方法，通过目标网络的一体化、欺骗性防御，误导攻击者的探测过程，消耗其攻击资源。

(下转第 702 页)

nications with Will Zou. learning optimization Greedy layer-wise training of deep networks [C] // Proceedings of the 20th Annual Conference on Neural Information Processing System. Cambridge, MA: MIT Press, 2006: 153 - 160.

[4] BENGIO Y. Learning deep architectures for AI [J]. Foundations & Trends ® in Machine Learning, 2009, 2(1): 1 - 127.

[5] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders [C] // Proceedings of the 2008 25th International Conference on Machine Learning. New York: ACM, 2008: 1096 - 1103.

[6] CHEN M, WEINBERGER K, SHA F, et al. Marginalized denoising auto-encoders for nonlinear representations [C] // Proceedings of the 2014 31th International Conference on Machine Learning. New York: ACM, 2014: 1476 - 1484.

[7] VINCENT P, LAROCHELLE H, LAJOIE I, et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion [J]. Journal of Machine Learning Research, 2010, 11(6): 3371 - 3408.

[8] LeCUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278 - 2324.

[9] FARABET C, COUPRIE C, NAJMAN L, et al. Learning hierarchical features for scene labeling [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1915 - 1929.

[10] MOHAMED A, DAHL G E, HINTON G. Acoustic modeling using deep belief networks [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(1): 14 - 22.

[11] LeCUN Y, BOTTOU L, ORR G B, et al. Efficient BackProp [M] // ORR G B, MÜLLER K-R. Neural Networks: Tricks of the Trade, LNCS 1524. Berlin: Springer, 1998: 9 - 50.

[12] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation, 2006, 18(7): 1527 - 1554.

[13] JAITLY N, HINTON G E. Using an autoencoder with deformable templates to discover features for automated speech recognition [EB/OL]. [2015-04-07]. http://www.cs.toronto.edu/~ndjaitly/jaitly-interspeech13.pdf.

[14] TSURUOKA Y, TSUJII J, ANANIADOU S. Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty [EB/OL]. [2015-04-07]. http://aye.comp.nus.edu.sg/~antho/P/P09/P09-1054.pdf.

[15] LE Q V, NGIAM J, COATES A, et al. On optimization methods for deep learning [EB/OL]. [2015-04-07]. http://ai.stanford.edu/~ang/papers/icml11-OptimizationForDeepLearning.pdf.

Background

This work is partially supported by the National Natural Science Foundation of China (61273225) and the National Key Technology R&D Program (2012BAC22B01).

DENG Junfeng, born in 1989, M. S. candidate. His research interests include machine learning, data mining.

ZHANG Xiaolong, born in 1963, Ph. D., professor. His research interests include data mining, machine learning, biological information processing.

(上接第664页)

参考文献:

[1] ZHOU H F, WU C M, JIANG M, et al. Evolving defense mechanism for future network security [J]. IEEE Communications Magazine, 2015, 53(4): 45 - 51.

[2] ATIGHETCHI M, PAL P, WEBBER F, et al. Adaptive use of network-centric mechanisms in cyber defense [C] // Proceedings of the 2003 6th IEEE International Symposium on Object-oriented Real-time Distributed Computing. Washington, DC: IEEE Computer Society, 2003: 183 - 192.

[3] KIM Y-H, PARK W H. A study on cyber threat prediction based on intrusion detection event for APT attack detection [J]. Multimedia Tools and Applications, 2014, 71(2): 685 - 698.

[4] TRIFERO S, CALLAWAY D. Linux stealth patch [EB/OL]. [2013-10-29]. https://packetstormsecurity.com/files/download/29706/linux-2.2.22-stealth.diff.gz.

[5] REHMET G. FreeBSD blackhole [EB/OL]. [2013-10-29]. http://www.gsp.com/cgi-bin/man.cgi?section=4&topic=blackhole.

[6] HARTMEIER D. OpenBSD packet filter [EB/OL]. [2013-10-06]. http://www.openbsd.org/faq/pf/index.html.

[7] ROUALLAND G, SAFFROY J M. IP personality [EB/OL]. [2013-10-06]. http://ippersonality.sourceforge.net.

[8] DARREN REED. Fingerprint Fucker [EB/OL]. [2013-10-06]. http://packetstormsecurity.org/UNIX/misc/bsdfpf.tar.gz.

[9] CRENSHAW A. OSfuscate: change your Windows OS TCP/IP fingerprint to confuse P0f, Network Miner, Ettercap, Nmap and other OS detection tools [EB/OL]. [2013-11-01]. http://www.irongeek.com/i.php?page=security/osfuscate-change-your-windows-os-tcp-ip-fingerprint-to-confuse-p0f-networ km iner-ettercap-nmap-and-other-os-detection-tools.

[10] 马君亮,汪西莉,何聚厚,等. 增强型 Anti-Xprobe2 的研究与设计 [J]. 计算机工程与应用, 2012, 48(32): 1 - 4. (MA J L, WANG X L, HE J H, et al. Research and design of enhanced Anti-Xprobe2 [J]. Computer Engineering and Applications, 2012, 48(32): 1 - 4.)

[11] PRIGENT G, VICHOT F, HARROUET F. IpMorph: fingerprinting spoofing unification [J]. Journal in Computer Virology, 2010, 6(4): 329 - 342.

Background

CAO Xu, born in 1983, Ph. D. candidate. His research interests include cloud computing and network information security.

FEI Jinlong, born in 1981, Ph. D., lecturer. His research interests include network information security.

ZHU Yuefei, born in 1964, Ph. D., professor. His research interests include cryptology and network information security.