专题：网络安全的智能化和高对抗性发展

面向 Web 隐藏后门技术的防御

陈利跃[1]，孙歆[2]，成天晟[3]，吴春明[3]，陈双喜[3,4]

1. 国网浙江省电力有限公司，浙江 杭州 310027;
2. 国网浙江省电力有限公司电力科学研究院，浙江 杭州 310027;
3. 浙江大学，浙江 杭州 310058; 4.嘉兴职业技术学院，浙江 嘉兴 314036

摘　要：Rootkit 是一种持久且隐匿的攻击技术，通过修改操作系统软件或内核，更改指令执行路径，隐匿攻击行为和后门程序痕迹。首先介绍了 Rootkit 的基本定义及其演变过程，其次讨论了目前 Rootkit 工作原理、主流技术以及检测方法。然后通过安全性与性能对比实验，阐述了基于动态异构冗余架构搭建的拟态 Web 防御系统在木马攻击下的应用效果。实验结果表明，拟态 Web 防御系统能在较小开销的情况下有效地对木马攻击进行防御。最后总结了该系统在当前环境下所面临的机遇与挑战。

关键词：Rootkit；动态异构冗余架构；拟态防御

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−0801.2020142

Defense of hidden backdoor technology for Web

CHEN Liyue[1], SUN Xin[2], CHENG Tiansheng[3], WU Chunming[3],CHEN Shuangxi[3,4]

1. State Grid Zhejiang Electric Power Co., Ltd., Hangzhou 310027, China
2.State Grid Zhejiang Electric Power Co., Ltd. Research Institute, Hangzhou 310027, China
3. Zhejiang University, Hangzhou 310058, China
4. Jiaxing Vocational and Technical College, Jiaxing 314036, China

Abstract: Rootkit is a set of persistent and undetectable attack technologies, which can hide their attack behavior and backdoor trace by modifying software or kernel in operating system and changing execution path of instruction. Firstly, the basic definition and evolution of Rootkit were introduced, then the operating principle, current mainstream technology and detection methods of Rootkit were discussed. Then, through comparative experiments on performance and security, the application of mimic defense system was described for Web based on dynamic, heterogeneous, redundant structure under Trojan Horse attack. Experiments show that mimic defense system can effectively defend against Trojan Horse in tests in the premise of low overhead. At last, the opportunities and challenges of the DHR system were summarized.

Key words: Rootkit, dynamic heterogeneous redundant structure, mimic defense

---

收稿日期：2020−02−25；**修回日期**：2020−04−25

通信作者：陈双喜，abelchen@zju.edu.cn

基金项目：国家电网总部科技项目（No.52110118001F）

Foundation Item: The Science and Technology Funding Project of State Grid (No.52110118001F)

1 引言

随着互联网技术的飞速发展，网络已成为人们生活中不可或缺的一部分。与此同时，黑客通过不断出现的软件漏洞以及自动化攻击工具，利用隐匿技术，不断渗透、潜伏并控制网站，严重危害了个人、企业乃至国家的利益。据 2019 年第 44 次《中国互联网发展状况统计报告》[1]，仅 2019 年上半年，国家互联网应急中心（CNCERT）检测发现并协调处置我国境内被篡改网站近 4 万个，其中被篡改的政府网站有 222 个，CNCERT 检测境内外约 1.4 万个 IP 地址对我国境内约 2.6 万个网站植入后门，同比增长 1.2 倍，这充分说明网络犯罪已经对个人及国家构成了巨大威胁。

大多数入侵和计算机安全事件都遵循着一种常见的模式，即黑客扫描目标系统中易受攻击的服务，之后发起攻击获取系统的某种访问权限，并最终提升其权限，然后黑客利用这些特权创建后门，使自己在之后能重返系统，为了防止用户或管理员注意到安全漏洞，黑客往往采取隐藏系统中已被入侵的痕迹的方法，通常将这些持久且隐匿的攻击技术统称为 Rootkit[2]。

Rootkit 一词来源于 UNIX 系统。在 UNIX 中，root 也称为根用户，是 UNIX 和类 UNIX 系统中唯一的超级用户，拥有系统中的最高权限，Kit 则是管理工具，因此，Rootkit 便指能够维持 root 特权的工具，利用这类技术，黑客可以在用户毫无知觉的情况下获取信息或控制权限。简单来说，Rootkit 是一种特殊的攻击技术，它通常和木马、后门等恶意程序结合使用，在目标系统上隐藏自身及指定文件、进程、网络链接等信息，因而往往会造成比较大的危害。

参考文献[3]中给出了 Rootkit 在 Windows 下的技术演变过程。其演变遵循从简单到复杂、从高层到底层的演化趋势。按照 Rootkit 层次，可将其划分为 5 代：应用层 Rootkit、内核层 Rootkit、内存视图伪装 Rootkit、虚拟机层 Rootkit 以及硬件层 Rootkit[3]。其演化时间轴如图 1 所示。从 1999 年出现首个 Windows Rootkit 开始，Rootkit 不断前进演化，2004 年，首次出现了直接操纵内核对象的 Rootkit，从此 Rootkit 进入了内核对象技术发展阶段，2005 年，首次展示了内存伪装技术，2006 年，虚拟 Rootkit 技术逐渐发展，该技术专门为虚拟环境设计。而最早的硬件层 Rootkit 则是 2005 年的 eEyeBoot Rootkit，随后的 BIOS Rootkit 等也都属于硬件层 Rootkit 范畴。

2 Rootkit 工作原理与主流技术

为了了解 Rootkit 的工作原理，本文以 Windows 为例，介绍 Windows 系统结构。Windows 系统采用层次化设计，从底层到上层可以分为硬件层、内核层和应用层 3 层，每层由若干组件构

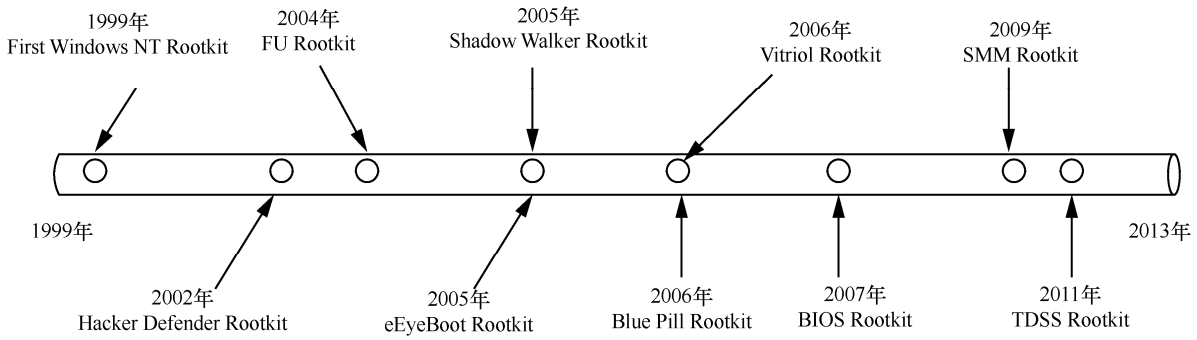


图 1 Rootkit 演化时间轴

成。作为一个整体，Windows 的运行依赖于上层对下层 API 的调用。Windows 系统层次结构如图 2 所示。

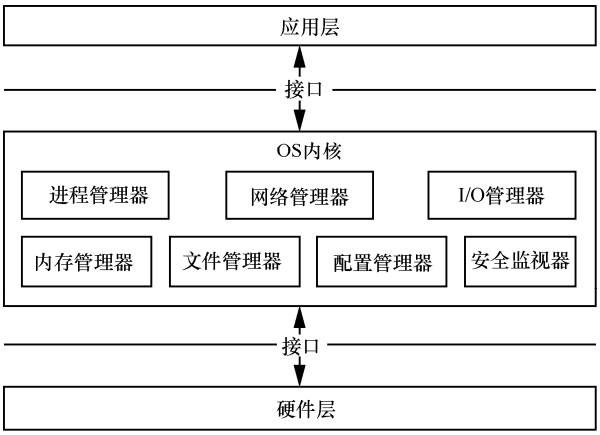


图 2　Windows 系统层次结构

Windows 系统层次化的结构有益于系统扩展、升级，但同时也给黑客提供了便利。Rootkit 正是通过修改下层模块的数据结构来隐藏自身的秘密行为。对于图 2 中不同的系统组件，Rootkit 使用不同的技术进行攻击。例如，内核层中的进程管理器提供对系统进程的可视化监控。Rootkit 可以通过修改 EPROCESS 数据结构来完成对自身的隐匿。

Rootkit 的主流技术包括以下 4 种。

（1）Hook 技术

Hook 技术又被称为钩子技术，最早在 Microsoft 中用作自定义查询用户信息技术的扩展[4]。Rootkit 使用这项技术来隐藏和自身有关的注册表信息，从而实现隐匿的目的。Hook 技术直接通过修改函数字节的开头（覆盖原始指令所在的区域）来跳转到 Rootkit 所在的代码。基于所在层次的不同，Hook 技术也有多种。在应用层，Hook 技术有 IAT Hook、Inline Hook 等；在内核层，Hook 技术有 IDT Hook、SSDT Hook、IRP Hook[5]等。

（2）Filter Drivers 技术

Filter Drivers 技术又名过滤驱动技术，利用 Windows 分层结构，将 Rootkit 插入驱动中，从而隐藏自身攻击。

以文件管理系统为例，Windows 系统采用了输入输出管理程序（input/output supervisor，IOS）技术管理文件系统。IOS 是一种层次结构的管理方案，在应用层中展现各类应用程序的读和写。应用层下一层是可安装文件系统，可安装文件系统下一层就是文件系统驱动（file system driver，FSD）所在的层，同样这也是到达 IOS 前的最后一层。黑客通过 Filter Drivers 来掌控 FSD，拦截、修改正常的读写请求，将涉及自身的文件修改为除自身外无法读取。

（3）DKOM 技术

区别于修改或重定向指令执行的 Hook 与 Filter Drivers，DKOM（direct kernel object manipulation）采用直接修改内核对象的方式来实现隐匿自身的目的。例如，从双向链表 Active Process Links 中删除某些进程的 EPROCESS 对象结构，可以实现隐匿进程的作用[6]。

（4）虚拟化技术

虚拟 Rootkit 通过软件和硬件的虚拟化将代码置于操作系统（operating system，OS）下，从而使得软件层面的检测方法失去作用，达到隐匿自身的目的。常见的虚拟 Rootkit 分为 VMM Rootkit 与 SMM Rootkit 两种。

3　Rootkit 检测与防御方法

Rootkit 检测方法可以通过被检测模块所在的位置进行分类。本文将 Rootkit 的检测方法分为 3 类。

（1）基于主机的方法

基于主机的 Rootkit 检测方法通常从受到感染的系统开始工作。参考文献[7]给出了一种早期依靠系统文件的副本存档的检测方法。Kerncheck 通过存储内核中使用的符号以及正在运行中的系统调用表，并警告不一致的地方进行检测[7]。但是这种方法无法检测所有类型的攻击，并且对于正

常攻击也不一定能正常进行检测，同时必须保证副本存档是干净的。参考文献[8]给出了一种基于行为的方法。通过静态分析内核模块是否恶意，再决定是否将被检测内核模块加载并执行。检测基于以下两个行为标准：

- 是否对非法内存进行写操作；
- 模块包含禁止使用的内核符号引用，该引用用来计算内核地址空间地址，并基于此地址进行写操作。

（2）基于虚拟化的方法

基于虚拟化的Rootkit检测方法主要原理是在高于内核的层上监视内核模块，从而能够更有效地跟踪操作系统活动。参考文献[9]中给出的Livewire系统正是基于虚拟机监视器（VMM）的应用。Livewire 由一个策略引擎组成，并利用了VMM的隔离、检查和插入属性。该策略引擎通过VMM界面对系统状态和事件进行解释，并以此确定是否发生攻击。参考文献[10]则提供了一种基于系统管理程序的系统HookSafe，该系统可以有效防止Hook技术的攻击。HookSafe将目标OS当作Guest OS加载并监视可能会进行修改的钩子。对于内核中钩子的访问，HookSafe提供了一个钩子间接层。只有对钩子进行写操作时，才会将控制转移到管理程序。

（3）基于外部观察者的方法

基于外部观察者的Rootkit检测方法主要检测内核代码的完整性或者关键内核是否正常散列。参考文献[11]中给出了一种基于安全优先架构的细粒度主动可信检测度量方法。首先在离线预处理阶段对内核进行编译，根据内核符号表文件生成语义信息文件和初始度量值，随后在启动阶段计算出每个符号初始和结束位置的物理地址，最后在运行阶段定期计算内核数据结构的度量值，将新值与基准值校验以发现恶意攻击。参考文献[12]在基于外部硬件的内核完整性监视器监视内核的静态区域的基础上，引入了一种事件触发的内核动态对象监视技术，通过一个基于硬件的平台 KI-Mon 检测主机总线上的内存写流量，并过滤掉所有有意义的流量以生成事件。

4 基于动态异构冗余的拟态防御系统

基于对目前的攻击手段和和检测方法的分析，本文提出了基于动态异构冗余结构的拟态防御系统。相比主流的防御技术，拟态防御系统在面对未知漏洞或隐匿攻击时有着更好的表现。

4.1 动态异构冗余结构

动态异构冗余（dynamic heterogeneous redundant，DHR）结构是拟态防御的基本原理，DHR结构[12]如图3所示。通过利用DHR架构技术将动态性、异构性、冗余性等基本的内生安全属性导入网络空间的基础设施服务中，构建内生安全的网络架构模型，达到阻断攻击、感知漏洞威胁的效果[13]。

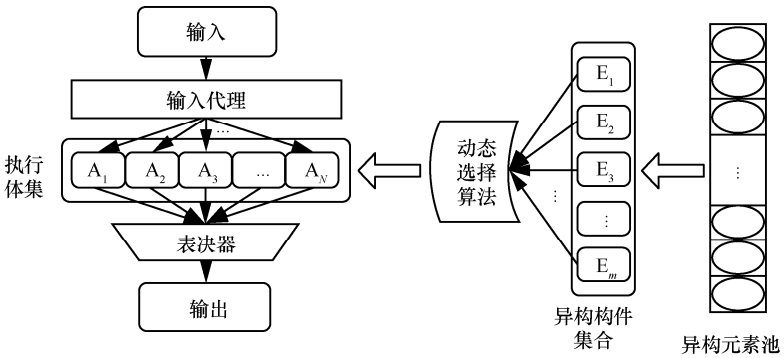


图3 DHR结构[12]

DHR 结构与传统计算机系统的区别主要表现在处理功能上。在处理过程中，输入代理将同一输入变为相同的多份，分别分发给 A_1、$A_2 \cdots A_n$ 等不同的执行单位。各执行单位将结果发送到表决器，通过表决得出唯一的输出。与此同时，动态选择算法会根据当前的反馈信息对执行单位进行合理的替换。

4.2　拟态 Web 防御系统

拟态 Web 防御系统采用了分层的结构设计[14]，如图 4 所示。在实战中，黑客往往会对 Web 服务器进行多层次全方面的攻击，因此在 Web 防御系统的多个层面做好防御极其重要。

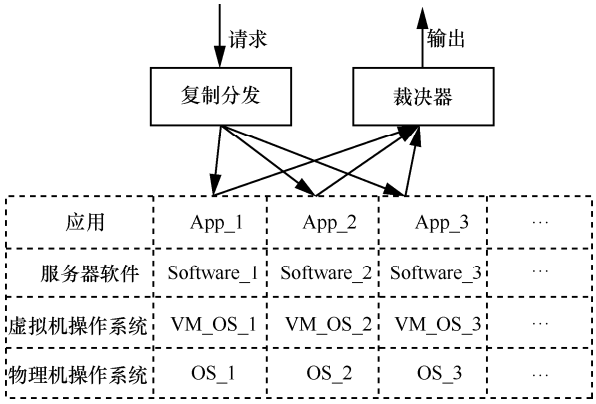


图 4　拟态 Web 防御系统结构

通过软件或硬件的支持，拟态 Web 防御系统在物理机和虚拟机 OS、服务器软件、SQL 脚本等多个层面进行了异构。在实际流程中，请求经过复制分发给了多个在不同层面的载体，经过层层处理，将处理发送至裁决器处进行唯一的输出表决。

5　实验设计与分析

本文采用了邬江兴院士提出的拟态防御机制[15]，结合第 4 节理论，设计了简化的基于动态异构冗余的拟态 Web 防御系统的实验。

5.1　实验环境搭建

本文的拟态防御系统由 node1、node2、node3、node4 共 4 台虚拟机和 1 台作为攻击机的 Windows10 机器组成。其中，node1、node2、node4 为异构服务器，它们的操作系统分别为 Ubuntu16、CentOS7、Windows2012。node3 担当数据和负载均衡服务器，操作系统为 CentOS7。各机器配置见表 1。

在 node1、node2、node4 上部署了相同的网站，将其置于负载均衡后端。系统结构如图 5 所示。

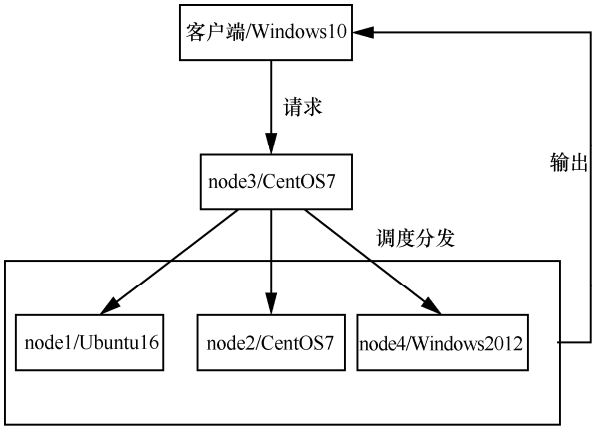


图 5　实验系统架构

node3 采用了 IP_hash 方法进行负载均衡，这样实验时可以将某个 IP 请求定向访问同一台机

表 1　实验机器配置

名称	用途	内存	处理器	硬盘	系统
node1	异构服务器	2 GB	Intel® Core™ i5-6300HQ CPU @ 2.30 GHz	20 GB	Ubuntu16.04.6
node2	异构服务器	1 GB		20 GB	CentOS7
node3	数据和负载均衡服务器	1 GB		20 GB	CentOS7
node4	异构服务器	2 GB		60 GB	Windows2012
Windows10	攻击机	8 GB		1 TB	Windows10

器，而防止在网页登录过程中 Web 服务器切换。

5.2 实验步骤与结果分析

由于常见的 Rootkit 一般都和木马、后门等恶意程序结合使用，在本次实验中，使用了 Reptile 木马进行测试。Reptile 是基于一种内核钩子框架 kHook 和动态模块加载器实现的 Rootkit，通过它可以隐藏文件、地址、内核模块、进程、TCP/IP 连接等信息。

为了测试木马的有效性，首先关闭了 node1 的所有防御，并主动在 node1 上放置了木马，随后通过攻击机进行检测，可直接获取 node1 的 root shell。直接通过 node1 的 shell 查看 Rootkit 效果，未搜索到进程和连接相关信息。

本文在 Web 服务器上设计了 3 种实验状态。状态 1：关闭所有安全防御软件；状态 2：开启操作系统自带的 WAF；状态 3：开启拟态防御。为了比较遭受攻击后系统的反应，Reptile 被直接植入 node1 中。实验结果见表 2。

从表 2 中可以看出，相比传统防护的 Web 服务器以及无防护的 Web 服务器，基于拟态防御的 Web 服务器针对 Rootkit 结合木马的攻击更有抵抗力，并能成功通过自身系统的特性来防御未知的攻击。

表 2 不同实验形态对比

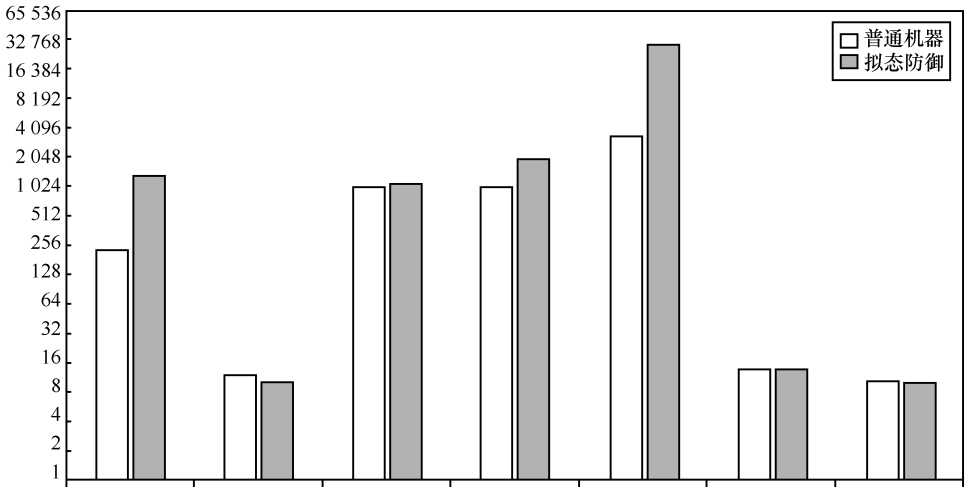
时间	无防护状态	正常防护状态	拟态防御状态
植入 Reptile 后	成功连接，并拿到 shell	成功连接，并拿到 shell	连接失败，攻击机再次尝试连接后，拿到 shell
一定时间后	成功连接，并拿到 shell	成功连接，并拿到 shell	连接被断开，尝试连接，无回应

5.3 系统业务性能测试与分析

在第 5.2 节拟态 Web 防御系统功能测试的基础上，测试拟态 Web 防御系统的业务性能。使用 Jmeter（压力测试工具）对普通机器和拟态防御下的网站分别进行测试。测试方法如下：10 min 内进行 6 000 次 HTTP 请求，通过响应时间、吞吐量等参数来比较网站在两种状态下的性能。实验结果如图 6 所示。

通过表 3 可得以下结论。

- 从中位数与 90%百分位可以发现，拟态 Web 防御系统在绝大多数情况下并没有造成特别明显的时延。
- 从异常率和吞吐量可以发现，拟态 Web 防



	平均值	中位数	90%百分位	95%百分位	99%百分位	异常率	吞吐量
普通机器	224 ms	12 ms	1 013 ms	1 023 ms	3 245 ms	13.53%	10 req/s
拟态防御	1 310 ms	10 ms	1 117 ms	1 991 ms	30 013 ms	13.48%	9.8 req/s

图 6 业务性能比较

御系统并不会造成额外的异常。

- 对于平均值和 99%百分位的巨大差异，参考吞吐量相差 0.2 req/s（请求次数/s），分析原因，是由极少数的 HTTP 请求超时响应导致的。因此，相较于普通机器，拟态 Web 防御系统在使用的过程中会付出一定的性能代价。

综合以上分析，对比同类服务器下的情况，拟态 Web 防御系统下的 Web 服务对绝大多数请求正常响应，并未有明显时延和异常。同时，极少数请求响应超时，也体现了该系统相比同类服务器的额外性能消耗。因此，本文所设计的简化系统仍然存在着较大的改进空间。

6　结束语

基于动态异构冗余结构设计的拟态防御技术，相比于传统典型的防御技术，能更有效地应对来自 Rootkit 的已知或未知威胁。但单从性能来看，拟态 Web 防御系统有着一定的损失。不过，拟态 Web 防御系统仍处在初步的研制阶段，有着较大的改进空间。作为一种主动防御的技术，拟态防御技术以其独特的异构技术、冗余裁决方式、动态选择算法，在安全防御中具有良好的应用前景。

参考文献：

[1] 中国互联网信息中心. 第 44 次中国互联网发展状况统计报告[R]. 北京: 中国网信网, 2018.
China Internet Network Information Center. The 44th statistical report on the development of China's Internet[R]. Beijing: Cyberspace Administration of China, 2018.

[2] KRUEGEL C, ROBERTSON W, VIGNA G. Detecting kernel-level rootkits through binary analysis[C]//Proceedings of Computer Security Applications Conference. Piscataway: IEEE Press, 2004: 91-100.

[3] 张瑜, 刘庆中, 李涛, 等. Rootkit 研究综述[J]. 电子科技大学学报, 2015(4): 563-578.
ZHANG Y, LIU Q Z, LI T, et al. Research and development of Rootkit[J]. Journal of University of Electronic Science and Technology of China, 2015(4): 563-578.

[4] FU D S, CAO C L. A windows rootkit detection method based on cross-View[J]. Information Technology, 2010(2): 1-3.

[5] 白光冬, 郭耀, 陈向群. 一种基于交叉视图的 Windows Rootkit 检测方法[J]. 计算机科学, 2009, 36(8): 133-137.
BAI G D, GUO Y, CHEN X Q. Windows rootkit detection method based on cross-view[J]. Computer Science, 2009, 36(8): 133-137.

[6] 陈晓苏, 黄文超, 肖道举. 一种基于交叉视图的 Windows Rootkit 检测方法[J]. 计算机工程与科学, 2007, 29(7): 1-3.
CHEN X S, HUANG W C, XIAO D J. A Windows rootkit detection method based on cross-view[J]. Computer Engineering &Science, 2007, 29(7): 1-3.

[7] LEVINE J G, GRIZZARD J B, OWEN H. A methodology to detect and characterize kernel level rootkit exploits involving redirection of the system call table[C]//Proceedings of Second IEEE International Information Assurance Workshop. Piscataway: IEEE Press, 2005: 107-125.

[8] KRUEGEL C, ROBERTSON W, VIGN G. Detecting kernel-level rootkits through binary analysis[C]//Computer Security Applications Conference. Piscataway: IEEE Press, 2004: 91-100.

[9] GARFINKEL T, ROSENBLUM M. A virtual machine introspection based architecture for intrusion detection[C]//Proceedings of Network and Distributed Systems Security Symposium. Piscataway: IEEE Press, 2003: 253-285.

[10] WANG Z, JIANG X, CUI W, et al. Countering kernel rootkits with lightweight hook protection[C]//Proceedings of the 16th ACM Conference on Computer and Communications Security. New York: ACM Press, 2009: 545-554.

[11] 田竞, 孙慧琪, 武希耀, 等. 一种基于安全优先架构的细粒度可信监测度量方法[J]. 信息安全学报, 2019, 4(5): 23-31.
TIAN J, SUN H Q, WU X Y, et al. A fine-grained trusted monitoring measurement method based on security-first architecture[J]. Journal of Cyber Security, 2019, 4(5): 23-31.

[12] LEE H, MOON H, HEO I, et al. KI-Mon ARM: a hardware-assisted event-triggered monitoring platform for mutable kernel object[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(2): 287-300.

[13] 仝青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报, 2017, 28(4): 883-897.
TONG Q, ZHANG Z, ZHANG W H, et al. Design and implementation of mimic defense Web server[J]. Journal of Software, 2017, 28(4), 883-897.

[14] 李卫超, 冯俊龙. 动态异构冗余的 Web 威胁感知技术研究[J]. 智能计算机与应用, 2018, 8(4): 42-46, 52.
LI W C, FENG J L. Research on dynamic heterogeneity redundant Web threat awareness technology[J]. Intelligent Computer

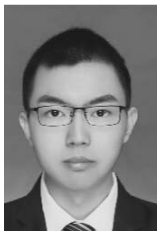
and Applications, 2018, 8(4): 42-46, 52.

[15] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报, 2016, 1(4): 1-10.
WU J X. Research on cyber mimic defense[J]. Journal of Cyber Security, 2016, 1(4): 1-10.

[作者简介]

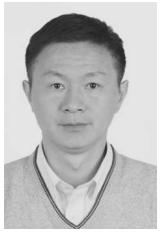
陈利跃（1973– ），男，国网浙江省电力有限公司处长、高级工程师，主要研究方向为信息安全管理。

孙歆（1981– ），男，国网浙江省电力有限公司电力科学研究院高级工程师，主要研究方向为新型网络安全攻防。



成天晟（1995– ），男，浙江大学硕士生，主要研究方向为网络空间安全。

吴春明（1967– ），男，浙江大学教授、博士生导师，主要研究方向为人工智能、柔性可重构网络体系、软件定义网络、网络主动防御创新安全技术等。



陈双喜（1980– ），男，嘉兴职业技术学院讲师，浙江大学博士生，主要研究方向为网络空间安全的渗透与主动防御。