自动探测和保护确保内核完整性

何 进，范明钰，王光卫

(电子科技大学计算机科学与工程学院 611731)

【摘要】内核rookits攻击对内核完整性构成致命威胁，因此对内核rootkits探测和防护确保内核完整性是当前研究的热点，然而现有的研究总存在不足：要么侧重内核rootkits防护，要么侧重内核rootkits探测，并未将两者相结合确保内核完整性。鉴于此，本文将探测和保护相结合形成一个自动联动机制，从而构成了基于探测保护的一体化系统ADPos来确保内核完整性。实验表明ADPos系统既能自动全面有效地探测与防护，而且又不牺牲系统性能为代价，并且兼容多种OS系统、同时防零日攻击。

关 键 词 ADPos; 探测模式; 联动; 内核完整性; 保护模式; rootkits
中图分类号 TP309 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.04.020

Automatic Detection and Protection System to Ensure Kernel Integrity

HE Jin, FAN Ming-yu, and WANG Guang-wei

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract Kernel-level rootkits pose a fatal threat to kernel integrity, so kernel-level rootkits detection and protection has become a hot topic. However, there are some drawbacks in these existing efforts: either focusing on rootkits protection, or focusing on rootkits detection, without the combination of both to ensure kernel integrity. In view of this situation, this paper designs a complete automatic interactive mechanism based on the detection and protection of kernel-level rootkits, thus forming an integrated detection and protection system (ADPos) to guarantee kernel integrity. The experiments show that the ADPos system can not only automatically detect and protect kernel integrity, but also does not sacrifice the system performance for the price. Moreover, the system is compatible with a variety of OS systems and against zero-day attacks.

Key words ADPos; detection mode; interactive mechanism; kernel integrity; protection mode; rootkits

Rootkits是攻击者向计算机系统植入的，能够隐藏自身踪迹并保留超级用户访问权限的恶意程序。rootkits分为用户级和内核级两种[1]，本文只对内核级rootkits进行探测和防护。内核级的rootkits对整个计算机安全构成致命威胁，它对内核发起攻击，主要通过如下途径：1) 带有rootkits的模块插入到内核；2) 内核自身安全漏洞，注入rootkits到内核；3) 通过应用安全漏洞提取最高权限，再向内核注入rootkits。

无论采用何种途径探测或防止rootkits攻击，最终目的是确保内核完整性。根据最近rootkits注入方式、特征及防范措施等研究，将这些工作分为两大类：1) 探测内核rootkits。Copilot[2]和改进的Copilot[3]都采用独立PCI卡周期获取内核内存，并判断内核完整性是否被破坏；SBCFI[4]探测控制流；OSck[5]基于不变式探测。这些探测模式共同缺陷：即便探测到rootkits攻击，也不能事后采取措施来保障内核完整性。2) 阻止rootkits注入。Secvisor[6]确保内核代码完整性；Patagonix[7]通过探测执行文件格式确保代码完整性性；NICKLE[8]通过内核代码认证进行保护；HookSafe[9]控制流的保护，侧重保护钩子函数。这些保护方式只针对内核的某一方面进行保护，即代码完整性保护[6,8]、数据完整性保护[7]、控制流完整性保护[10~12]，而缺乏统一全面保护措施。

上述探测模式和保护模式都各自存在弊端，也无法将二者优点相结合，既能探测到rootkits攻击，又能全面保护内核完整性。鉴于此，本文提出了将探测与保护相结合，确保内核完整性的方案，称为ADPos(automatic detection and protection OS)，ADPos系统基于VMM[13]，将被保护的内核置入客户

收稿日期：2013－03－21； 修回日期：2013－06－18
基金项目：国家863重点项目(2009AA01Z435,2009AA01Z403)；国家自然科学基金(60373109,60272091)
作者简介：何进(1977－)，男，博士，主要从事操作系统安全、虚拟技术及云安全等方面的研究.

机上(GOS)，客户机和VMM承载宿主机(HOS)之上，ADPos系统作为GOS一个安全部件，放置于VMM中。ADPos在探测模式下周期性自动探测，如果发现内核完整性被破坏，则迁移到保护模式下进行认证访问，并且ADPos对破坏部分的内核进行分析，并生成漏洞补丁或者认证证书下载到内核，并完成保护模式与探测模式自动转换从而高效确保内核完整性。同时大量实验表明ADPos针对rootkits攻击是一个高效的自动探测与保护系统。

1 ADPos设计

ADPos系统引入hyperwall[14]和HyperLock[15]技术确保ADPos运行的硬件平台和VMM是安全可信的。

定义 1(原始内存OM)：内核和内核模块转载的物理内存称为原始内存。将OM划分为多个原始内存逻辑块，表示为OMLB[i]且$\forall i \in \{1,2,\cdots,n\}$，本文用OMLB[$i$]表示任意一个原始内存逻辑块，用OMLB表示所有原始内存逻辑块。

定义 2(备份内存BM)：保存OM备份的内存称为备份内存。上述同理BMLB[i]表示任意一个备份内存逻辑块，BMLB表示所有备份内存逻辑块。

定义 3(认证内存AM)：存放内核认证访问的内存称为认证内存。AMLB[i]表示任意一个认证内存逻辑块，AMLB表示所有认证内存逻辑块。

定义 4(映射表MT)：记录OM和AM之间映射关系称为映射表。MTE[i]表示任意一个映射表项，MTE表示所有映射表项。每个映射表项包含以下元素：MTE表项标号(MTE[i].seq)、OMLB块起始地址(OMLB[i].osa)、OMLB块大小(OMLB[i].os)、OMLB块Hash值(OMLB[i].ohv)、BMLB块号和BMLB块起始地址(BMLB[i].bsa)等。

定义 5(通用访存)：不加入安全系统或加入安全系统但不改变访问流程称为通用访存，如图1a所示：GOS物理地址→ADPos→VMM。加入ADPos通用访存，如图1b所示：GOS物理地址→SPT[13](Shadow Page Table)→OM内存。

定义 6(认证访存)：加入ADPos安全部件，对AM写访存需要认证才能访问称为认证访问，如图1c所示：GOS物理地址→SPT→AM内存。

定义 7(探测模式)：通用访存的内核位于OM内存中，ADPos的线程CHKPrecess周期检测完整性是否被破坏，称这一探测过程为探测模式，如图1d所示。

定义 8(保护模式)：认证访存的内核位于内存AM中，GOS访问(写访问)该内存需要通过认证才能访问，称这一保护过程为保护模式，如图1d所示。

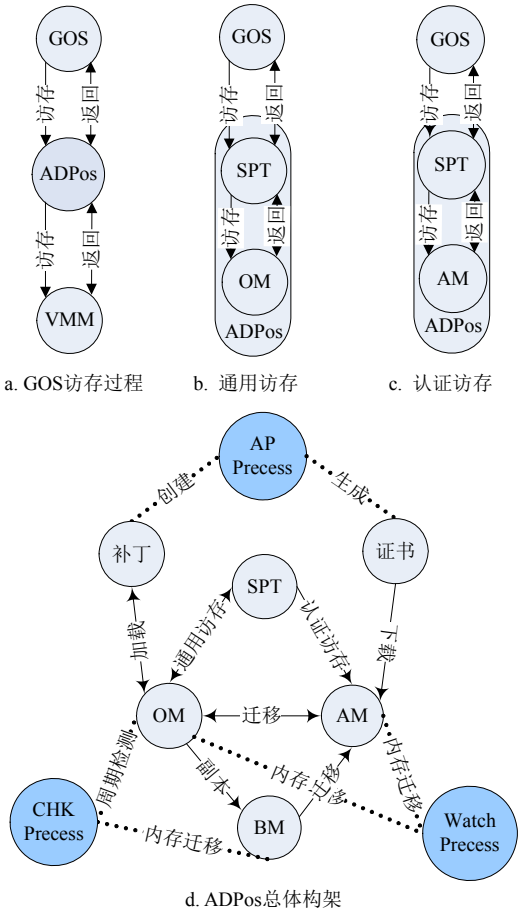

图1 ADPos访存方式和总体构架

ADPos系统是一个自动探测与防御的综合系统，如图1d所示。CHKPrecess线程周期探测内核完整性，如果被破坏，则终止访问该内存的任务，并自动迁移被破坏的内存OMLB[i]的副本BMLB[i]至认证内存AMLB[i]，同时向应用程序APPrecess发送告警信息(包括破坏相关信息)，这时被破坏部分内核的内存由探测模式转成保护模式；任务APPrecess收到告警信息进行分析，如果能生成漏洞补丁，则加载到内核并向WatchPrecess线程发送补丁加载消息，否则创建认证证书，下载至内核，并向WatchPrecess线程发送证书命令；当WatchPrecess线程收到补丁信息，则认为认证部分的内核加载补丁之后是可信的，将内存AMLB[i]迁移至OMLB[i]，这样内存从保护模式转化为探测模式。当WatchPrecess线程收到证书命令，则内存AMLB[i]还是处于保护模式之下，写操作时需要认证访问。

2 ADPos实现

ADPos系统被分为内核态和用户态两个部分，

如图2所示。内核态完成自动探测和保护功能，包括通用访存和认证访存两种访存方式及两种模式迁移。用户态包括交互模块、告警信息分析、补丁模块和License证书4个部分，以及内核态与用户态交互，模式转化等。详细过程见3.3节。

ADPos启动时分配AM、BM和MT共3部分内存，并分别分成多个逻辑块或表项AMLB、BMLB、MTE。被保护内核加载至内存后，但还未执行之前，OM与BM之间建立副本关系，并对OMLB逻辑块进行Hash校验，将Hash值$Hash_{[OMLB[i]]}$存入BM中。启动3个任务：CHKPrecess线程、WatchPrecess线程和APPrecess应用程序，前两个任务运行在内核态，后面这一任务运行在用户态，就此完成ADPos初始化过程。

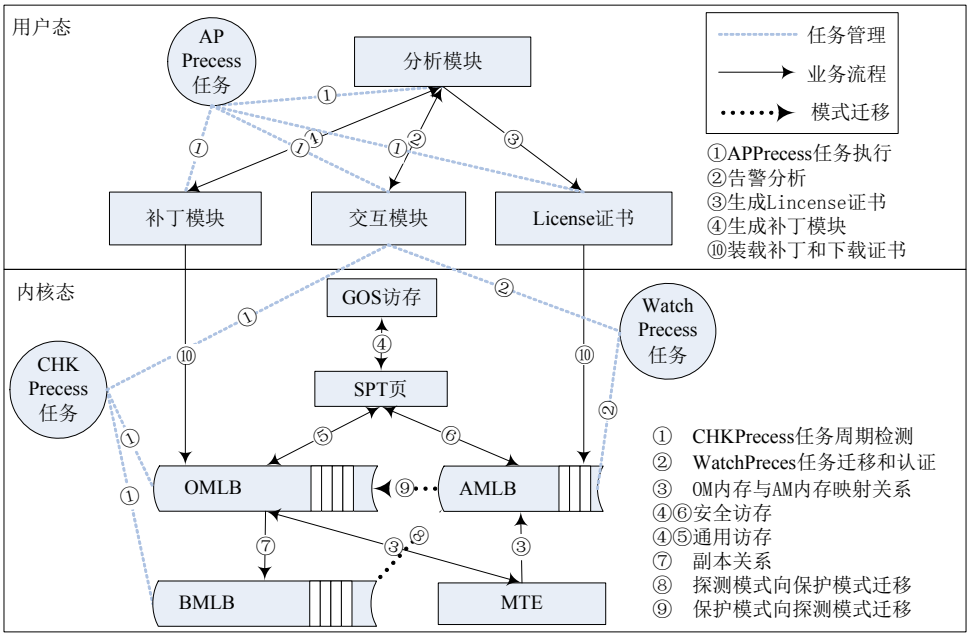

图2　ADPos系统实现逻辑

2.1　完整性检测

ADPos完整性检测方法：CHKPrecess线程逐一读取每个表项MTE[i]，根据OMLB[i]起始地址和OMLB[i]大小Hash获得值$Hash_{[OMLB[i]]}$，将校验的$Hash_{[OMLB[i]]}$和MTE[i]初始化保存的Hash值进行比较，如果不相等则表明OMLB[i]块已经被破坏，需进一步处理：从AM中分配逻辑块AMLB[i]，将OMLB[i]块的备份BMLB[i]迁移至AMLB[i]，并修改SPT页表指向OMLB[i]块，同时将OMLB[i]和AMLB[i]之间的映射关系写入MTE[i]。详细过程见算法1。

算法1：ADPos完整性检测算法

CHKPrecess线程检测

for MMT[i]且$\forall i \in \{1,2,\cdots,n\}$ {

获取MTE[i].OMLB[i].os，and MTE[i].OMLB[i].os;

$Hash_{[OMLB[i]]}$=

Hash(MTE[i].OMLB[i].osa，MTE[i].OMLB[i].os);

if(MTE[i]. OMLB[i].ohv = $Hash_{[OMLB[i]]}$){

Continue;

}

Else{

AMLB[i]=malloc(AM);

Get idle MTE[i];

BMLB[i]→AMLB[i];

MTE[i]={ OMLB, BMLB };

Modify SPT→AMLB[i] ;

Alarm Log(MTE[i]) To Admin;

Set MTE[i].lb标示not uesed

}

}

等待下一个检测周期

2.2　完整性保护

定义 9(认证中断)：ADPos初始化时，创建一个认证中断向量及与之对应的认证中断服务例程。任何访问写AM都要触发认证中断向量进入认证中断服务例程，认证中断服务程序查找与之对应的AMLB[j]逻辑块，从而获得MTE[j]对应的认证证书$Licence_{[MTE[j]]}$。根据$Licence_{[MTE[j]]}$认证证书，判断

该次访问是否可信，如果可信则进行访问；否则拒绝访问。

GOS访问AM见算法2，SPT映射至AMLB[*j*]内存，如果该次访问为写访存，则触发认证中断向量，跳转至认证中断服务例程，该例程根据认证证书Licence[$_{MTE[j]}$]，判断该次访问是否可信，如果可信则访存，否则返回失败；如果该次访问为读访存则直接访问。

算法2：ADPos完整性保护算法

```
GOS访存；
SPT页表映射至AMLB[j]逻辑块；
if (visit AMLB[j] is writable){
    触发认证中断向量；
    jmp 认证中断服务程序；
    认证中断服务程序处理{
        查找AMLB[j]逻辑块；
        获取AMLB[j]对应表项MTE[j]；
        根据表项MTE[j]获取对应的认证证书Licence[MTE[j]]；
        if (Licence[MTE[j]]允许访存)
            访存AMLB[j]；
        else
            访存失败。
    }
}
允许访存。
```

2.3 探测与保护联动机制

2.2节和2.3节分别阐述了ADPos检测和保护，它们是两个独立的个体，没有形成联动，ADPos核心点就是建立完整性探测和保护联动机制，并形成一个完整的自动探测保护系统。需要引入转换点和切换点，转换点是探测模式与保护模式之间相互转化过程，切换点是内核模式与用户模式之间相互转化过程，ADPos系统的切换点是探测模式和保护模式与用户模式之间相互转化过程。

转换点1(探测模式→保护模式)：CHKPrecess线程探测到OMLB[*i*]被篡改，迁移其副本BMLB[*i*]到认证AMLB[*i*]内存，也就是探测到内核漏洞的逻辑块自动转入认证保护，并且修改SPT页表指向AMLB[*i*]。

转换点2(保护模式→探测模式)：WatchPrecess线程收到补丁加载命令Patch[$_{MTE[i]}$]，将AMLB[*i*]迁移至OMLB[*i*]中，重建它们之间的副本关系，并修改SPT页表指向OMLB[*i*]逻辑块。

切换点1(探测模式→用户模式)：CHKPrecess线程向APPrecess应用程序发送告警信息Alarm[$_{MTE[i]}$]，VMM和Qemu自动完成内核模式→用户模式转化，而ADPos系统完成探测模式→用户模式转化。

切换点2(用户模式→保护模式)：APPrecess应用程序对告警信息进行分析，如能生成补丁模块则加载到内核，并向WatchPrecess线程发送命令Patch[$_{MTE[i]}$]，VMM和Qemu自动完成用户模式→内核模式转化，ADPos系统完成用户模式→保护模式转化。

ADPos系统通过探测发现内存被篡改或注入，触发转换点1，将其副本迁移到保护模式下，并向APPrecess应用程序发送告警信息Alarm[$_{MTE[i]}$]，触发切换点1，切换到用户模式。APPrecess对警信息Alarm[$_{MTE[i]}$]进行分析处理，如能生成漏洞补丁模块，加载该模块，并向WatchPrecess线程发送命令Patch[$_{MTE[i]}$]；否则向WatchPrecess线程发送Licence[$_{MTE[i]}$]，无论何种方式都触发切换点2，切换到保护模式。如果WatchPrecess收到命令Patch[$_{MTE[i]}$]，则触发转换点2，将保护模式转换到探测模式；如果收到Licence[$_{MTE[i]}$]，维持保护模式不变。这一系列过程构成了自动探测与保护联动机制。

3 实验及其分析

ADPos系统的实验主要侧重两个方面：1)防护能力；2)系统性能。实验环境：Dell PowerEdge T310、2.4G主频、Intel Xeon X3430、4 GB内存，Xen[16]ypervisor基于3.4.2版本，dom 0系统为Fedora 12，使用64位Ubuntu、内核为2.6.24作为客户机OS。

防护能力：采用当前内核rootkits攻击工具对载入ADPos模块的系统进行攻击，测试结果见表1。攻击类型：插入模块修改控制流程、全局变量、修改或注入调用表或中断表等。实验表明ADPos不仅能够有效探测这些rootkits攻击，并且防止再次攻击，确保内核的完整性。

表1 通过现有内核rootkits攻击ADPos系统，探测与防护情况

rootkit	攻击对象	探测情况	补丁模块	保护情况
Adore-ng	控制流	√	无	√
eNYeLKM	代码	√	有	不需防护
extable	控制流	√	无	√
Superkit	控制流	√	无	√
mood-nt	数据	√	无	√
hideme.vfs	代码	√	有	不需防护

系统性能：采用XEN作为性能测试基准，Lmbench[17]作为测试性能工具。

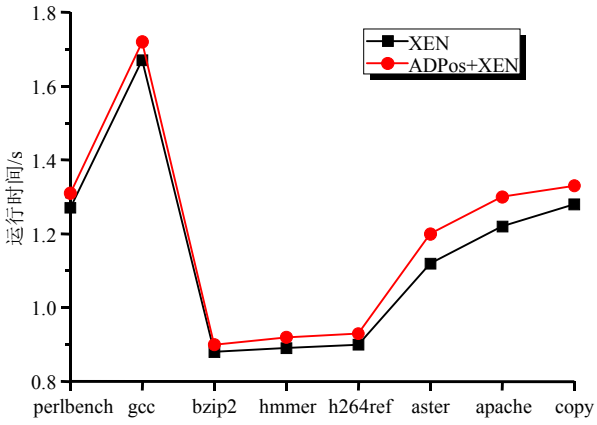

图3　应用程序执行效率方面，ADPos和XEN性能对比

ADPos对应用程序的性能影响：应用程序访存过程，应用程序→系统调用→GOS虚拟地址→GOS物理地址→SPT→AM或OM。尽管ADPos只对内核完整性进行保护，不对应用程序完整性进行保护，但ADPos在两个方面对应用程序性能可能造成影响：调用系统接口，内核需要访问ADPos模块以及线程ChkPrecess周期检测OMBL。图3通过XEN系统对比了ADPos+XEN对应用程序的影响。实验表明ADPos系统对应用程序性能影响很小，可以忽略。主要原因是线程ChkPrecess尽管周期扫描OMBL内存，但只有内核被破坏，迁移内存至认证部分时，才导致性能下降，其他时候对系统性能不造成影响。

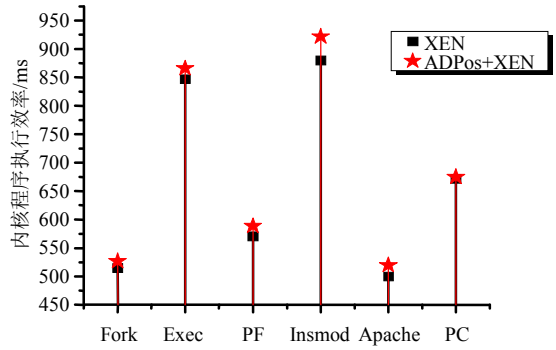

图4　ADPos和XEN在内核程序执行效率方面性能对比

ADPos系统对内核程序的性能影响：测试并分析ADPos对系统接口的调用、内核模块加载、内核执行等操作对系统性能的影响。

1) ADPos＋XEN系统与XEN系统性能对比：XEN作为性能基准。图4可以看出加入安全部件ADPos对系统运行效率未构成大的影响(<3%)，分析影响性能主因是Fork、Exec、FC(File Copy)和Apache由于线程ChkPrecess周期探测对性能造成一定的影响；Insmod插入模块对系统性能影响主要做内存备份。说明了ADPos是一种高效的自动探测与防御系统。

2) 图5为ADPos系统与其他同类系统的对比。实验表明了ADPos系统比SecVisor和NICKLE系统性能好，主要由于ADPos只有遭受到攻击才能导致系统性能有所下降，正常状态几乎对系统性能不产生任何影响，并且ADPos将OM内存切分多个逻辑块进行快速迁移和认证。

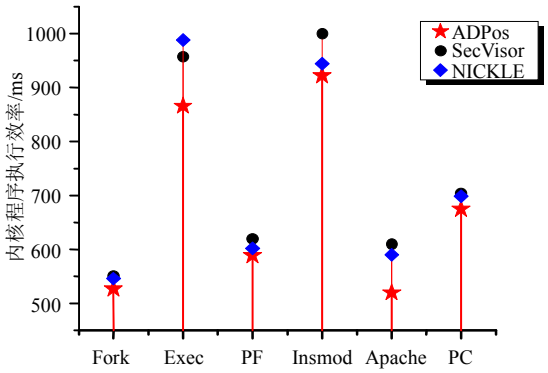

图5　ADPos与SecVisor和NICKLE内核程序执行效率方面性能对比

4　结　论

本文基于VMM虚拟技术，呈现ADPos的设计、实现及其ADPos实验分析，表明了ADPos不仅能够全面高效自动探测和防护确保内核完整性，同时兼容多操作系统。因此ADPos可以广泛应用于商业领域。

参　考　文　献

[1] NGUYEN A Q, YOSHIYASU T. Towards a tamper resistant kernel rootkit detector[C]//SAC'07. Seoul, Korea: [s.n.], 2007.

[2] PETRONI N, FRASER T, MOLINA J, et al. Copilot: a coprocessor-based kernel runtime integrity monitor[C]// Proceedings of the 13th USENIX Security Symposium. [S.l.]: [s.n.], 2004.

[3] PETRONI N L Jr, FRASER T, WALTERS A, et al. An architecture for specification-based detection of semantic integrity violations in kernel dynamic data[C]//Proceedings of the 15th USENIX Security Symposium. [S.l.]: [s.n.], 2006.

[4] PETRONI N L Jr, HICKS M. Automated detection of persistent kernel control-flow attacks[C]//Proceedings of the ACM Conference on Computer and Communications Security. [S.l.]: ACM, 2007.

[5] HOFMANN D, KIM R. Ensuring operating system kernel integrity with osck[C]//ACM—SIGPLAN Notices. [S.l.]: ACM, 2011.

[6] SESHADRI A, LUK M, QU N, et al. A tiny hypervisor to provide lifetime kernel code integrity for commodity oses.

[C]//Proceedings of twenty-first ACM SIGOPS Symposium on Operating Systems Principles. New York, USA: ACM, 2007.

[7] LITTY L, LAGAR-CAVILLA H A, LIE D. Hypervisor support for identifying covertly executing binaries[C]// Proceedings of the 17th USENIX Security Symposium. [S.l.]: [s.n.], 2008.

[8] RILEY R, JIANG X, XU D. Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing[C]// Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection. Berlin, Heidelberg: Springer-Verlag, 2008.

[9] 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, USA: ACM, 2009.

[10] CHECKOWAY S, DAVI L, DMITRIENKO A, et al. Return-oriented programming without returns[C]// Proceedings of the 17th ACM Conference on CCS. New York, USA: ACM, 2010.

[11] BLETSCH T, JIANG X, FREEH V. Mitigating code-reuse attacks with control-flow locking[C]//ACSAC 11. Orlando, Florida USA: [s.n.], 2011.

[12] LI J, WANG Z, BLETSCH T, et al. Comprehensive and efficient protection of kernel control data[J]. IEEE Transactions on Information Forensics and Security, 2011, 6(4): 1404-1414.

[13] DANIEL P. Virtual machine manager[R/OL]. [2013-2-19]. http://virt-manager.org/.

[14] SZEFER J, LEE R B. Architectural support for hypervisor-secure virtualization[J]. SIGARCH Comput Archit News, 2012, 40(1): 437-450.

[15] WANG Z, WU C, GRACE M, et al. Isolating commodity hosted hypervisors with HyperLock[C]//Proceedings of the ACM European Conference on Computer Systems (EuroSys). New York, USA: ACM, 2012.

[16] Linux Foundation. The XEN project[R/OL]. [2013-2-25]. http://www.xen.org/.

[17] Bitmover Company. Performance analysis tool[R/OL]. [2013-2-19]. http://www.bitmover.com/lmbench/.

编 辑 税 红

(上接第556页)

LI Hui, YANG Ming-hao. Robust estimation algorithm for distribution system leverage measurements[J]. Automation of Electric Power Systems, 2005, 29(3): 31-35.

[7] 李慧. 用于配网负荷处理的全面抗差估计方法[J]. 电工技术学报, 2008, 23(2): 138-142.
LI Hui. A comprehensive robust estimation algorithm for distribution network load data processing[J]. Transactions of China Electrotechnical Society, 2008, 23(2): 138-142.

[8] 蔡凝露, 幺莉, 林济铿, 等. 基于指数权函数的抗差状态估计算法[J]. 中国电力, 2013, 46(4): 69-73.
CAI Ning-lu, YAO Li, LI Ji-keng, et al. A robust state estimation method based on exponential weight functions[J]. Electric Power, 2013, 46(4): 69-73.

[9] 周江文. 抗差最小二乘法[M]. 武汉: 华中理工大学出版社, 1997.
ZHOU Jiang-wen. Poor resistance to the least square method[M]. Wuhan: Huazhong University of Science Press, 1997.

[10] 吴文传, 郭烨, 张伯明. 指数型目标函数电力系统抗差状态估计[J]. 中国电机工程学报, 2011, 31(4): 67-71.
WU Wen-chuan, GUO Ye, ZHANG Bo-ming. A robust state estimation method with exponential objective function[J]. Proceedings of the CSEE, 2011, 31(4): 67-71.

[11] 郭烨, 张伯明, 吴文传. 指数型目标函数电力系统抗差状态估计的解法与性能分析[J]. 中国电机工程学报, 2011, 31(7): 89-95.
GUO Ye, ZHANG Bo-ming, WU Wen-chuan. Solution and performance analysis to a robust state estimation method with exponential objective function[J]. Proceedings of the CSEE, 2011, 31(7): 89-95.

[12] MILI L, CHENIAE M G, VICHARE N S, et al. Robust state estimation based on projection statistics of power systems[J]. IEEE Transactions on Power Systems, 1996, 11(2): 1118-1127.

[13] ZHAO Liang, ABUR A. Multi area state estimation using synchronized phasor measurements[J]. IEEE Transactions on Power Systems, 2005, 20(2): 611-617.

编 辑 漆 蓉