2ZA-05

KVMにおける機密情報の拡散追跡機能を用いた複数VM監視手法の評価

岡崎 俊樹† 森山 英明‡ 山内 利宏† 佐藤 将也† 谷口 秀夫†

†岡山大学 大学院自然科学研究科 ‡有明工業高等専門学校 創造工学科

1 はじめに

計算機上で機密情報を扱う機会の増加とともに，機密情報が計算機外部に漏えいする事例が増加している．そこで，仮想計算機モニタ (VMM) である KVM を用いて，計算機内部における機密情報の拡散状況を追跡する機能（以降，VMM における拡散追跡機能）を提案した [1]．また，VMM における拡散追跡機能を用いて，複数の VM を監視対象とする手法（以降，複数 VM 監視手法）を提案した [2]．本稿では，複数 VM 監視手法について，監視対象とする VM 数を増加させた場合の影響について評価した結果を報告する．

2 VMMにおける拡散追跡機能

2.1 基本機構

機密情報の拡散は，プロセスがファイル形式で存在する情報を開いてその内容を読み込み，さらに他のプロセスやファイルなどにその内容を伝えることにより発生する．VMM における拡散追跡機能は，監視対象の VM が発行するすべてのシステムコールをフックし，機密情報が拡散する経路を追跡し，ログに出力する．具体的には，機密情報が拡散する処理であるファイル操作，子プロセス生成，およびプロセス間通信の際に発行されるシステムコールの処理を VMM 上で監視し，計算機内に拡散したファイル情報，プロセス情報，およびソケット情報を拡散情報として管理する．この機能を VMM 上に実現することにより，ゲスト OS よりも高い権限で動作する VMM 上で機密情報を管理でき，ゲスト OS のソースコードを改変することなく機能を提供できる利点がある．

2.2 複数 VM 監視手法

VMM 上で多数の VM を動作させる環境を想定し，複数の VM を監視対象とする拡散追跡機能を提案した [2]．複数 VM 監視手法は，System Management BIOS が VM ごとに一意に割り当てる UUID と，仮想 CPU に紐づいた構造体である kvm 構造体の情報を用いて，システムコールを発行した VM を VMM 上で判別し，監視対象の各 VM 内における機密情報の拡散状況を追跡する．

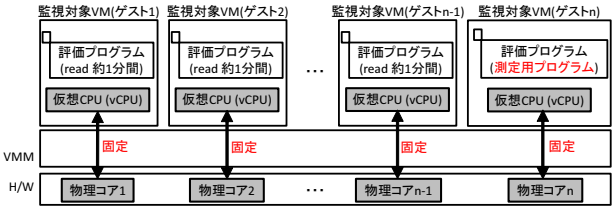

図 1 評価に用いた計算機環境

3 評価

3.1 評価内容

VMM 上で複数の VM を監視対象として機密情報拡散を追跡することによる性能面の影響を明らかにするため，図 1 に示す計算機環境において，システムコール実行時間に生じるオーバヘッドを測定した．具体的には，監視対象とする VM を n 台 ($2 \leq n \leq 7$) 動作させ，1 台目から n - 1 台目までの VM において，read() を約 1 分間発行し続けるプログラムを走行させ，n 台目の VM においてシステムコール実行時間を測定した．また，監視対象 VM が 1 台の場合と機能導入前（監視対象 VM が 0 台）の場合では，VM を 1 台動作させ，システムコール実行時間を測定した．測定では，VMM における拡散追跡機能が追跡対象としている read，および VMM における拡散追跡機能が追跡対象としていない getpid を測定対象とした．

評価には，64GB のメモリと Intel Xeon E5-2609 V4（1.7GHz，8 コア）の CPU を搭載した計算機を用い，ホスト OS として Fedora18 (Linux 3.6.10 64bit)，VMM として KVM-kmod-3.9 を用いた．また，監視対象 VM 上で動作する OS として Fedora18 (Linux 3.6.10 64bit) を用い，動作させる VM には，1GB のメモリ，1 コアの仮想 CPU を割り当てた．

3.2 評価結果と考察

システムコール実行時間の測定結果を表 1 に示す．「機能導入後」における「非管理対象資源の操作」と「管理対象資源の操作」の各項目は，それぞれ，測定用プログラムの read の処理において，機密情報拡散の追跡対象となるファイルを操作しない場合と操作する場合である．表 1 から以下のことがわかる．

Evaluation of Multiple VMs Monitoring Method Using Tracing Diffusion of Classified Information on KVM
Toshiki Okazaki†, Hideaki Moriyama‡, Toshihiro Yamauchi†, Masaya Sato†, Hideo Taniguchi†
†Graduate School of Natural Science and Technology, Okayama University
‡National Institute of Technology, Ariake College

表 1 システムコール実行時間 (μs)

		機能導入前 (監視対象なし)	機能導入後 (監視対象あり)							
			監視対象の VM が 1 台		監視対象の VM が 2 台		監視対象の VM が 4 台		監視対象の VM が 7 台	
			非管理対象資源の操作	管理対象資源の操作	非管理対象資源の操作	管理対象資源の操作	非管理対象資源の操作	管理対象資源の操作	非管理対象資源の操作	管理対象資源の操作
getpid	最大値	0.11	7.81	-	8.05	-	21.11	-	26.71	-
	最小値	0.094	7.19	-	7.22	-	7.31	-	7.26	-
	平均値	0.097	7.19	-	7.36	-	7.49	-	7.77	-
	中央値	0.097	7.19	-	7.34	-	7.35	-	7.36	-
	分散	$1.67 * 10^{-6}$	0.0056	-	0.0060	-	1.89	-	6.16	-
read	最大値	7.52	17.95	18.22	46.06	87.64	40.97	133.66	45.23	676.04
	最小値	7.22	16.72	17.05	17.11	17.05	16.99	17.02	17.13	17.14
	平均値	7.27	17.36	17.89	21.15	21.92	17.95	19.13	18.85	27.29
	中央値	7.26	17.16	17.26	17.25	17.72	17.12	17.16	17.25	17.31
	分散	0.0014	0.18	0.37	69.54	125.69	10.13	146.09	32.36	385.63

(1) 機密情報の拡散に関係しないシステムコールである getpid では，監視対象 VM が 1 台の場合と比較して，監視対象 VM 数が 2 台，4 台，および 7 台の場合で最大値が 0.24μs，13.30μs，18.90μs 増加しており，監視対象 VM 数が多いほど大きな増加量となっている．VMM における拡散追跡機能は，フックしたシステムコールが機密情報の拡散に関係しないと判定した場合，制御をゲスト OS に戻し，システムコール処理を続行する．このため，監視対象の VM 間において，VMM によるシステムコールのフック処理が重複した場合，システムコール実行時間に生じるオーバヘッドが大きくなると推察できる．

(2) read の測定結果では，非管理対象資源を操作する場合，監視対象の VM が 1 台の場合と比較して，監視対象 VM 数が 2 台，4 台，および 7 台の場合で最大値が 28.11μs，23.02μs，27.28μs 増加しており，getpid よりも大きな増加量となっている．これは，機密情報の拡散に関係のあるシステムコールである read をフックした際，VMM における拡散追跡機能は，機密情報拡散の追跡に必要となるプロセスとファイルの情報を取得する処理を行い，VMM 上で管理している拡散情報を更新するためであると推察できる．

また，管理対象資源を操作する場合では，監視対象 VM が 1 台の場合と比較して，監視対象 VM 数が 2 台，4 台，および 7 台の場合で最大値が 69.42μs，115.44μs，657.82μs 増加しており，非管理対象資源を操作する場合よりも，さらに大きな増加量となっている．これは，プロセス情報の管理表を走査してログに書き出す処理を行うため，非管理対象資源を操作する場合と比較して処理時間が長大化するためであると推察できる．

さらに，最大値のように，オーバヘッドの大きな増加が発生する割合は，1〜2 %であった．今回の測定のように，システムコールを連続で呼び出し続けるアプリケーションでなければ，複数 VM の実行におけるオーバヘッドが特に問題となることは少ないと推察できる．

(3) getpid では，最小値と中央値は，監視対象 VM 数に関係なく，7μs 程度であり，監視対象 VM 数を増加させた場合の値の変化が小さい．また，getpid と同様に，read についても，最小値と中央値の値は，監視対象 VM 数と操作する資源の種類に関係なく，17μs 程度である．このため，システムコール実行時間が大きく増加する場合は比較的少なく，性能面への影響は小さいと推察できる．

4 関連研究

VMM を用いてゲスト OS 上のファイルを保護する手法である ForenVisor[3] や CFWatcher[4] は，VMM における拡散追跡機能と異なり，ゲスト OS として Linux のみでなく，Windows についても監視対象とできる利点がある．しかし，複数 VM 監視手法と異なり，監視対象とできる VM 数が 1 台のみであるため，サーバ用途をはじめとした複数の VM の動作が前提となる環境には導入できない問題点がある．

5 おわりに

VMM における拡散追跡機能を用いた複数 VM 監視手法の評価結果について述べた．評価により，複数 VM 監視機構の導入による性能面への影響が比較的小さいことを示した．

謝辞 本研究の一部は，科学研究費補助金基盤研究 (B)（課題番号：16H02829）の助成を受けたものです．

参考文献

[1] Fujii, S., Sato, M., Yamauchi, T., and Taniguchi, H.: Design of Function for Tracing Diffusion of Classified Information for IPC on KVM, Journal of Information Processing, Vol.24, No.5, pp.781–792, (2016).

[2] 岡﨑俊樹，森山英明，山内利宏，佐藤将也，谷口秀夫：KVM 上の複数 VM の動作に対応した機密情報の拡散追跡機能，コンピュータセキュリティシンポジウム 2017 (CSS2017) 論文集, vol.2017, no.2, pp.1295–1301 (2017).

[3] Zhengwei, Q., Chengcheng, X., Ruhui, et al.: ForenVisor: A Tool for Acquiring and Preserving Reliable Data in Cloud Live Forensics, IEEE Transactions on Cloud Computing, Vol.5, No.3, pp.443–456, (2017).

[4] Dongyang, Z., Lin, Y., Binxing, et al.: Protecting Critical Files Using Target-Based Virtual Machine Introspection Approach, IEICE Transactions on Information and Systems, Vol.E100.D, No.10, pp.2307–2318, (2017).