B-022

KVM における機密情報の拡散追跡機能の高速化
Improving Performance of Function for Tracing Diffusion of Classified Information on KVM

森山 英明† 山内 利宏‡ 佐藤 将也‡ 谷口 秀夫‡
Hideaki Moriyama Toshihiro Yamauchi Masaya Sato Hideo Taniguchi

1. はじめに

計算機内で管理されている機密情報は，外部に漏えいすることで，企業や個人にとって大きな損失となる．機密情報を保持したファイルへの誤操作や，外部からの機密情報へのアクセスを検知するために，仮想計算機モニタ（VMM: Virtual Machine Monitor）を利用した機密情報の拡散追跡機能が提案され，実現されている．この機能では，機密情報を操作するシステムコールをフックして確認することで，機密情報が格納されているファイルへの操作内容を利用者に通知することができる．しかし，処理のオーバヘッドが大きいという問題がある[1].

本稿では，オーバヘッドを分析した結果，およびオーバヘッドを削減する対処について述べ，評価した結果について報告する．

2. KVM における機密情報の拡散追跡機能

計算機内の機密情報の利用状況を把握するために、仮想計算機モニタにおける機密情報の拡散追跡機能を、KVM（Kernel-based Virtual Machine）上に実現し，評価結果を報告した[2]．機密情報の拡散追跡機能は，機密情報を有する可能性のあるファイル（以降，管理対象ファイル）に対して操作を行ったプロセスを，管理対象プロセスとして登録し，管理対象プロセスからプロセスの複製やファイルへの書き出しがあれば，機密情報の拡散として検知し情報を記録する．この機能を VMM 上に実装することにより，オペレーティングシステム（OS）よりも攻撃が困難である VMM で機密情報を管理でき，また，ゲスト OS のソースコードを改変することなく VMM の改変のみで機能を提供できるという利点がある．

KVM における機密情報の拡散追跡機能では，ゲスト OS 上で発行されるシステムコールをフックするために，ハードウェアブレークポイントを用いてシステムコール発行前（SYSCALL 命令）と終了直前（SYSRET 命令）でデバッグ例外を発生させる．これにより，処理をゲスト OS から VMM へ移行させる． VMM 側では，デバッグアドレスレジスタによるデバッグ例外であることを確認し，システムコールに応じて，拡散追跡に必要な情報を取得する．

KVM における機密情報の拡散追跡機能の処理フローを図 1 に示す．ユーザプロセスがシステムコールを発行すると，デバッグ例外が発生し，SYSCALL 命令と SYSRET 命令のどちらの実行前に発生したものか確認する．次に，システムコール番号から，機密情報の拡散に関係するシステムコールであるか否かを判定する．関係するシステムコールの場合は，SYSCALL 命令実行前の拡散追跡処理として，プロセスが発行したシステムコール番号，ページテーブル情報，扱うファイルのファイルディスクリプタの値などを取得する．SYSRET 命令実行前の拡散追跡処理では，システムコール処理の成否，システムコールを発行したプロセスが扱うファイルの情報などを取得する．もし，機密情報の拡散に関係しないシステムコールの場合は，拡散追跡にともなう処理を行わず，システムコール処理を続行する．

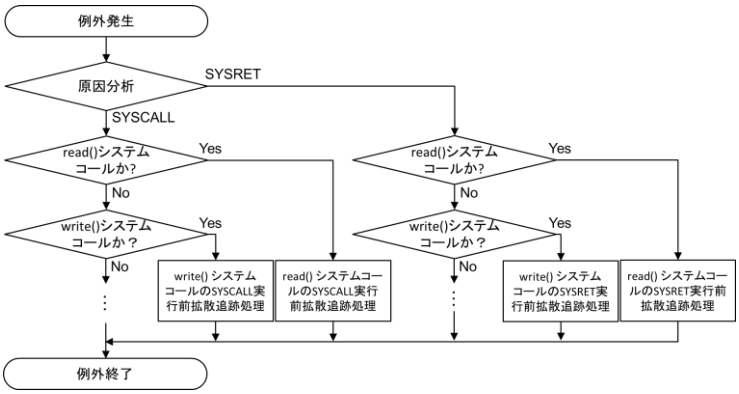

図 1 機密情報の拡散追跡機能の処理フロー

表 1 測定環境

測定用計算機	
CPU	Intex Xeon CPU E5-2609
コア数	8個
メモリ	64GB
OS	Fedora18 (Linux Kernel 3.6.10) (64bit)
VMM	KVM-kmod-3.6
仮想計算機	
コア数(vCPU)	1個
メモリ	1GB
OS	Fedora18 (Linux Kernel 3.6.10) (64bit)

3. オーバヘッドの分析

KVM における機密情報の拡散追跡機能において，オーバヘッドとなる処理を明確化するために，前段階の測定を行った[1]．この結果より，各システムコールの発行において，SYSRET 命令実行前の拡散追跡処理は， SYSCALL 命令実行前の拡散追跡処理と比べて，オーバヘッドが大きいことが明らかになっている．そこで，SYSRET 命令実行前の拡散追跡処理を詳細に分析し，処理時間の測定を行った．測定環境を表 1 に示す．測定では，仮想計算機上で作成した管理対象ファイルに対して，cp コマンドによるファイル複製を行い，read()と write()システムコールに対する SYSRET 命令実行前の拡散追跡処理の時間を測定した． 測定結果から，処理を分析し内訳を示した結果を図 2 に示す．図 2 から以下のことがわかる．

まず，read()システムコールの発行時において，SYSRET 命令実行前の拡散追跡処理を分析したところ，read 時検査処理が 90.1%を占めていることがわかった．この処理では，

†有明工業高等専門学校
‡岡山大学大学院自然科学研究科

発行された read()システムコールが管理対象ファイルを対象としているかを確認し，対象としていた場合は，read()システムコールを発行したプロセスを管理対象プロセスとして登録し，ログに出力する．さらに，この処理を詳細に分析すると，追跡対象プロセス表示処理は，SYSRET 命令実行前の拡散追跡処理の 89.9%を占めていることがわかった．この処理は，新たに機密情報が拡散したプロセス（管理対象プロセス）が増えた際に，すべてのプロセスの情報をログに出力するため，処理のオーバヘッドが大きいと考えられる．

また，write()システムコールの発行時において，SYSRET 命令実行前の拡散追跡処理を分析したところ，write 時検査処理が 87.0%を占めていることがわかった．この処理では、write()システムコールが管理対象ファイルを対象としているかを確認し，対象としていた場合は，write()システムコールによってデータ更新のあったファイルを管理対象ファイルとして登録し，ログに出力する．さらに，この処理を詳細に分析すると，SYSRET 命令実行前の拡散追跡処理のうち，ログ作成処理は 60.6%を，追跡対象ファイル表示処理は 26.4%を占めていることがわかった．これより，新たに機密情報が拡散したファイル（管理対象ファイル）が増えた際に，追跡対象ファイル表示処理ですべてのファイル情報をログに出力する処理のオーバヘッドが大きいと考えられる．また，write 時検査処理の一部であるフルパス構築処理は，管理対象ファイルのフルパスを取得する処理であり，オーバヘッドは 37.0%と大きい．

4. 対処と評価

4.1 対処

3 章で述べたオーバヘッドを削減するために，以下の 2 つの対処を行った．

（対処 1）read 時検査処理では，すべてのプロセス情報をログに出力せず，新たに管理対象プロセスとなったプロセスの情報のみを出力する．

（対処 2）write 時検査処理では，すべてのファイル情報をログに出力せず，新たに管理対象ファイルとなったプロセスの情報のみを出力する．

なお，フルパス構築処理における管理対象ファイルのフルパスを取得する処理は，現状のままとする．管理対象ファイルの追加時に必ずしもフルパスを取得する必要は無いため，今後，この処理のオーバヘッドを削減するには，フルパスを含む情報をユーザが参照する際にフルパスを取得するように修正することで対処できると考えられる．

4.2 評価結果

read 時検査処理と write 時検査処理について，4.1 節で述べた対処を行う前と後を比較した際の高速化の効果を図 3 に示す．まず，図 3 (A)から，read()システムコールの発行時における SYSRET 命令実行前の拡散追跡処理は，対処により，追跡対象プロセス表示処理を 97.6%削減できた．これより，対処後の追跡対象プロセス表示処理は，対処後の全体処理時間の 18.4%となる．また，図 3 (B) から，write()システムコールの発行時における SYSRET 命令実行前の拡散追跡処理は，対処により，追跡対象ファイル表示処理を 72.1%削減できた．これより，対処後の追跡対象ファイル表示処理は，対処後の全体処理時間の 9.4%となる．

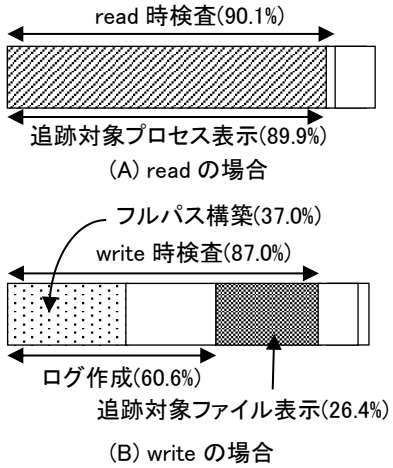

図 2 SYSRET 実行前の拡散追跡処理の内訳

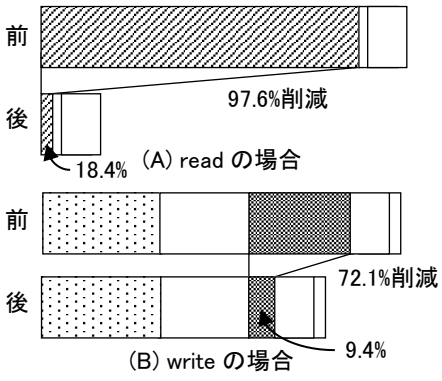

図 3 高速化の効果

対処を行う前の測定では，機密情報の拡散追跡機能の処理時間は，read()システムコール発行時に約 60 μ 秒，write()システムコール発行時は約 25 μ 秒であった．本研究の対処により，read()システムコール発行時の拡散追跡処理は 7.1 μ 秒， write()システムコール発行時の拡散追跡処理は，19.9 μ 秒となり，処理オーバヘッドを大幅に改善できたことがわかる．

5. おわりに

本稿では，KVM における機密情報の拡散追跡機能に関して，オーバヘッド箇所の特定，およびオーバヘッドを削減する対処を検討し，評価した．評価より，オーバヘッド削減の効果を確認した．

残された課題として，対処後の方式のアプリケーションによる評価，および参照時にフルパスを取得する機能の実現がある．

謝辞

本研究の一部は JSPS 科研費 16H02829（基盤研究(B)）の助成を受けたものです．

参考文献

[1] 森山 英明，山内 利宏，佐藤 将也，谷口 秀夫，“KVM における機密情報の拡散追跡機能における性能改善策”，情報処理学会第 79 回全国大会講演論文集第 1 分冊，pp.13-14（2017）．

[2] Fujii, S., Sato, M., Yamauchi, T., and Taniguchi, H. “Evaluation and Design of Function for Tracing Diffusion of Classified Information for File Operations with KVM”, The Journal of Supercomputing, Vol.72, Issue 5, pp.1841-1861 (2016).