L-017

仮想計算機を用いた重要ファイル保護手法の評価
Evaluation of File Protection Method on Virtual Machine Monitor

佐藤 将也† 谷口 秀夫† 山内 利宏†
Masaya Sato Hideo Taniguchi Toshihiro Yamauchi

1. はじめに

攻撃の検知や防止を行うためのソフトウェアは，攻撃者にとって不都合であることから，攻撃の対象となる．ここでは，これらのソフトウェアを重要サービスと呼び，重要サービスの設定などに用いられるファイルを重要ファイルと呼ぶ．重要サービスが攻撃を受け停止すると，攻撃の検知や防止が困難になり，被害が拡大する可能性がある．攻撃者は，重要サービスを攻撃する際に，攻撃対象のサービスの存在を特定し，攻撃を行う．このため，重要サービスの特定を困難にすることで，重要サービスへの攻撃を防止できる．そこで，著者らはこれまでに，重要サービスの特定を困難化する攻撃回避手法[1]した．本稿では，特定手段として重要ファイルを用いられた際の重要ファイル保護手法[2]について，評価を述べる．

2. 仮想計算機を用いた重要ファイル保護手法[2]

重要ファイル保護手法の全体像を図 1 に示す．提案手法では，重要サービスを提供するプロセス（以降，重要プロセス）が動作する環境と重要ファイルの配置場所を分ける．このため，仮想計算機（Virtual Machine, 以降 VM）を用いる．重要プロセスは保護対象 VM で動作し，重要ファイルはファイル提供 VM 上の代理プロセスによりアクセスされる．攻撃者は，保護対象 VM に不正なソフトウェアを挿入することで攻撃を行うことを想定する．このため，攻撃者が通常プロセスを攻撃し，保護対象 VM の仮想ディスクの内容を読み込み可能になった場合でも，重要ファイルを検出できない．これにより，重要ファイルの検知による重要プロセスの特定を困難にする．また，重要プロセスから重要ファイルへのアクセスは，VMM が検知し，代理プロセスに代理実行を依頼する．代理プロセスは，代理実行の結果を VMM 経由で重要プロセスに返却する．これにより，重要プロセスは，重要ファイルが異なる VM に管理されていることを意識することなく，重要ファイルにアクセスできる．

重要プロセスから重要ファイルへのアクセスの検知には，VMM による保護対象 VM のシステムコール検知手法を用いる．システムコール検知手法では，VMM がデバッグレジスタを用い，保護対象 VM のシステムコール開始処理にハードウェアブレークポイントを設定する．また，デバッグ例外の発生により，VM exit を発生させるよう設定する．これにより，保護対象 VM 上でシステムコールが実行されると，VM exit により VMM へ処理が遷移する．これにより，システムコールの発行を検知する．

提案手法におけるモード遷移を図 2 に示す．VMM は，システムコールの発行を検知すると，発行元のプロセスが重要プロセスから否かを判定し，重要プロセスの場合にはシステムコールの　制御を行う．システムコールの制御では，システムコールの種別がファイル操作であり，かつ操作対象が重要ファイルの場合のみ，代理プロセスに代理実行を依頼する．その他の場合は，保護対象 VM 上のオペレーティングシステム（以降，OS）による本来の処理に返却する．また，VMM は，代理実行の結果を代理プロセスから受け取ると，重要プロセスに実行結果を返却する．この際，本来の処理が終了し，ユーザモードの処理に復帰する処理に返却することで，本来の処理を行わない．このため，システムコール処理を攻撃者が監視することで，ファイルアクセスを監視されることを防止する．

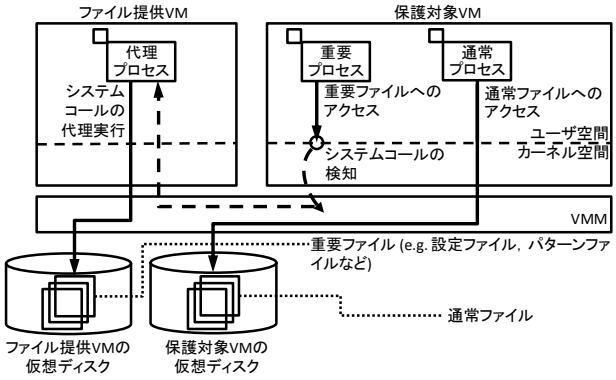

図 1 重要ファイル保護手法の構成

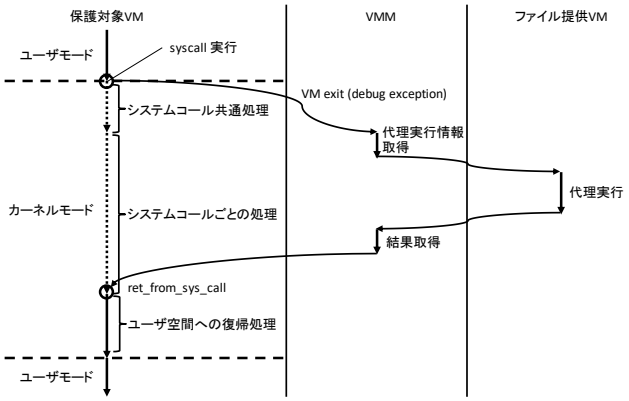

図 2 代理実行におけるモード遷移

3. 評価

3.1 目的と環境

基本評価では，提案手法を導入した際のファイル操作に関するシステムコールの処理性能を明らかにする．応用プログラムの性能の評価では，セキュリティソフトウェアである ClamAV 0.100.0 によりファイルを検査した際の処理性能を評価した．評価は，Intel Core i7-2600 (3.6 GHz，4 コア)，16GB メモリを搭載した計算機で行った．保護対象

† 岡山大学大学院自然科学研究科，Graduate School of Natural Science and Technology, Okayama University

VM には 1 仮想 CPU と 1GB RAM，ファイル提供 VM には 3 仮想 CPU と 15GB RAM を割り当て，各 VCPU は物理 CPU コアに固定して評価した．ファイル提供 VM では Linux 3.2.0 が動作し，保護対象 VM では，Linux 3.2.65 が動作する環境で評価した．VMM は，Xen 4.2.3 を用いた．

3.2 基本性能

ファイル操作に関するシステムコールの処理性能を評価し，提案手法の導入による性能低下を明らかにする．また，上記以外のシステムコールにおける性能への影響を評価する．評価対象のシステムコールは，ファイル操作に関するもの（read, write, open, close）である．read と write では，それぞれ 10 バイトの入出力を行った．なお，本評価に用いた提案手法では，open 実行時にカレントディレクトリを取得する処理を実装していない．カレントディレクトリを取得する処理は残された課題である．評価では，それぞれのシステムコールで対象とするファイルは，すべて重要ファイルとして測定した．

評価結果を図 3 に示す．システムコールごとの処理時間では，open の処理時間の増加が最も大きい．これは，提案手法により代理プロセスが代理実行する際に，VMID と重要プロセスの PD に対応するディレクトリを生成するためであると考えられる．ディレクトリ生成の際，代理プロセスにおいて，既に当該ディレクトリが生成されているかを確認し，生成されていない場合には，ディレクトリを生成する．これらの処理が処理時間の増加の原因であると考えられる． read と write の性能低下は，close よりも大きい．これは，read と write の代理実行において，read と write のバッファを保護対象 VM-VMM 間と VMM-ファイル提供 VM 間でコピーするためである．

open における処理時間の増加を抑制するためには，ディレクトリ生成の契機を変更する方法が考えられる．提案手法では，重要ファイルに対する open を最初に発行した際に，重要プロセスの VMID と PD の値からディレクトリを作成している．このため，open を最初に発行した際の処理時間の増加が大きい．そこで，重要プロセス起動を契機として，事前にディレクトリを生成する方法が考えられる．

3.3 応用プログラムの性能

提案手法の主な対象は，セキュリティソフトウェアや管理ツールである．そこで，本節では，セキュリティソフトウェアである ClamAV [3]の実行プログラムを重要プロセスとした場合に，提案手法の導入による性能への影響を評価した．ClamAV は複数のコマンドから構成されており，本節では，指定したディレクトリ下のファイルが検知パターンに一致するか否かを検査する clamscan コマンドを用いた．評価では，100 個のファイルを持つディレクトリを検査対象とした場合に，保護対象 VM 上で clamscan を実行した際の処理時間を測定した．検査対象のファイルサイズは，すべて 4KB である．提案手法が想定する重要ファイルは，ファイルにより重要サービスの存在が推測されるものである．このため，本評価では，重要ファイルとして，clamscan が読み込むデータベースファイル(main.cvd)を指定した．データベースファイルのファイルサイズは，113MB である．検査対象のファイルは，すべて通常ファイルとした．

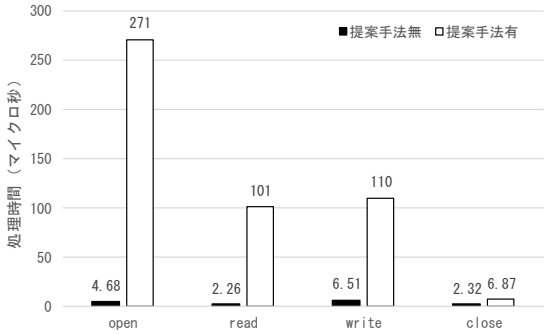

図 3 システムコールの処理時間

clamscan の処理時間を表 1 に示す．評価結果より，提案手法を導入した場合と導入しない場合では，処理時間の増加は 1%未満であり十分小さいことがわかる．また，提案手法を導入した場合に，重要ファイルの有無により処理時間に差があることが分かる．重要ファイルを操作する際は，代理実行が発生するため処理時間が増加する．clamscan は，実行時にデータベースファイルを読み込み，その内容をもとに検査対象のファイルを検査する．このため，本評価では，データベースファイルの読み込みでのみ，処理時間が増加する．しかし，clamscan の検査処理の処理時間の多くは，検査対象ファイルの検査であるため，データベースファイルの読み込みにおえる処理時間の増加が全体の処理時間に与える影響は小さい．

表 1 clamscan の処理時間（秒）

	提案手法無	提案手法	
		重要ファイル無	重要ファイル有
処理時間	12.24	12.25	12.31

4. おわりに

仮想計算機における重要ファイル保護手法の評価を述べた．提案手法は，重要サービスの利用するファイルである重要ファイルをもとに重要サービスが特定されるのを困難にする．このため，重要サービスを別 VM に配置し，重要プロセスからのみ重要ファイルを参照可能な機構について，評価結果を述べた．提案手法の評価では，基本性能として，提案手法の導入により open, read, write, および close システムコールの処理時間が最大で 266 マイクロ秒増加することを示した．また，応用プログラムの性能へ与える影響を評価した．評価では，重要プロセスとして clamscan を実行し，重要ファイルとしてデータベースファイルを指定した際の処理時間を測定した．評価結果より，提案手法の導入が clamscan の性能に与える影響は 1%未満であることを示した．残された課題として，重要ファイルに多くアクセスする重要プロセスを対象とした処理時間の評価がある．

謝辞

本研究の一部は JSPS 科研費 JP18K08151 と JP16H02829 の助成を受けたものです．

参考文献

[1] Masaya Sato, Toshihiro Yamauchi, Hideo Taniguchi: Process Hiding by Virtual Machine Monitor for Attack Avoidance, Journal of Information Processing, Vol. 23, No. 5, pp. 673–682 (2014).
[2] 佐藤将也，山内利宏，谷口秀夫：仮想計算機を用いた重要ファイル保護手法，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，pp. 1302–1308 (2017).
[3] Cisco: ClamAV, https://www.clamav.net/ (accessed 2018-06-28).