システムコールフックを用いた仮想マシン上のマルウェア検知と抑止

本田　惇[†]　高橋一志[†]　大山恵弘[†]

1. はじめに

近年、クラウドサービスが普及しており、その基盤として仮想マシンモニタ (VMM) が用いられている。その中で、“IaaS (Infrastructure as a Service)” と呼ばれる形式のサービスでは、VMM と各ゲスト OS の管理者は異なるため、ゲスト OS 上でマルウェアが実行されていても VMM の管理者はゲスト OS の状態を変更できない。したがって、マルウェアへの対処は、仮想マシン (VM) 全体を停止させるなど、被害者であるユーザまで VM を利用できなくなってしまうような粒度の粗い方法に限られる。

そこで本研究では、ゲスト OS 上で動作するマルウェアの実行を、VMM がプロセスレベルで抑止するシステムを構築する。具体的には、ゲスト OS 内で実行されているマルウェアを VMM が検知し、そのプロセスの実行速度のみを著しく低下させるシステムを構築する。マルウェアの検知はゲスト OS におけるプロセスごとのシステムコールのシーケンスを動的に解析することにより行う。そのためのシステムコール情報の取得は、ゲスト OS が発行する SYSENTER 命令をフックすることで実現する。また、マルウェアの実行速度低下は、仮想タイマ割り込みの間隔を調整することやシステム時間のエミュレートを行う[4] ことで実現する。

2. 提案システムの概要

本研究では、KVM[1] 上のあるドメイン (VM) 内でマルウェアが実行されているという状況を仮定する。このマルウェアが動作しているドメインをマルドメインと呼ぶ。KVM 上では、マルドメイン以外の善意のドメインや、マルドメイン内においてもマルウェア以外の善意のプロセスが動作している可能性があり (図 1)、この状態において、マルドメインの外部からマルウェアの実行を検知し実行速度を低下させるシステムを提案する。

† 電気通信大学 総合情報学専攻

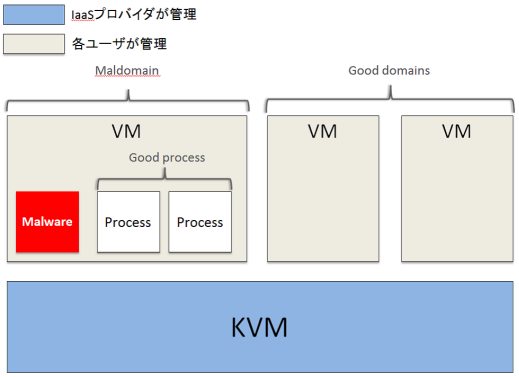

図 1　想定する脅威シナリオ

提案システムの構成を図 2 に示す。提案システムはプロセス挙動解析部とインターフェース部、仮想タイマ割り込み発行部から構成される。

プロセス挙動解析部は VM 上のゲスト OS が発行するシステムコールを、SYSENTER をフックすることによってインターセプトし、その引数や CR3 の値を取得する。得られた CR3 の値を擬似的なプロセス ID とすることで、VMM 層からプロセスの識別を行なうことが可能となる[3] ので、それらを用いてプロセスごとのふるまいを解析する。そして、プロセスがマルウェアだと判定した場合、CR3 の値をインターフェース部に通知する。

インターフェース部は、プロセス挙動解析部と連携をとり、マルウェアの CR3 の値を受け取る。この値は仮想タイマ割り込み発行部へと通知される。

仮想タイマ割り込み発行部は、マルドメイン内で動作するマルウェアの実行を抑止するための処理を行う。

3. マルウェアの検知

3.1 SYSENTER のフック

ゲスト OS が発行するシステムコールの情報は SYSENTER 命令をフックすることによって得ることができる。SYSENTER はシステムコールを高速に呼び出す命令である。MSR (Model Specific Register) にカーネルモードのルーチンを指すセレクタなどを入れてお

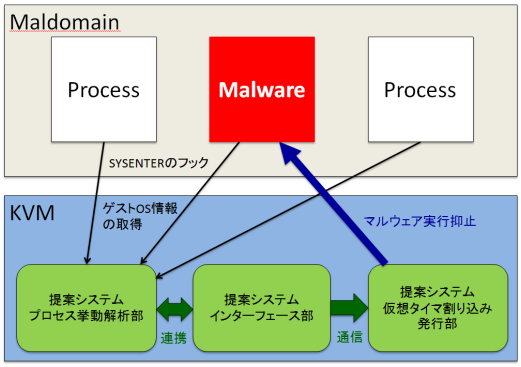

図 2 提案機構の概要図

き、カーネルモードに移行する時にそのレジスタから CS レジスタ、EIP レジスタ、ESP レジスタに値をロードすることでオーバーヘッドの少ないカーネルモード移行を実現している。上記の MSR は、`rdmsr` 命令および `wrmsr` 命令を使用して読み取りおよび書き込みを行っている。

そこで提案システムでは、`SYSENTER` 実行時に実在しないアドレスを EIP と ESP レジスタに割り当てる。具体的には、VMM は `wrmsr` 命令が呼ばれる時に、該当の MSR への書き込みのみをフックし、本来書き込む値の代わりに 0xffffffff という実在しないアドレスを書き込む。すると、`SYSENTER` 命令によって MSR の値を読み込む時にページフォールトが起こることで VM Exit が発生し、CPU のコントロールが VM から VMM に切り替わるので、VMM はその間にシステムコールの引数や CR3 の値などゲスト OS の情報を取得することができる。

3.2 ふるまい解析

上述で得られた CR3 レジスタの値を元にプロセスを識別することで、プロセス別のシステムコールのシーケンス情報を得る。それらは VM 内で動作しているプロセスがマルウェアであるかどうかを動的に検知するために用いられる。

実際の解析方法は検討中であるが、現在候補にあげているのは、システムコール列の N-gram を用いて既存のマルウェアとの類似度を判定する手法である。また、k 近傍法を用いたクラスタリング手法を取り入れるという方法も考えている。

4. マルウェアの実行抑止

マルウェアの実行を抑止する機構では、タイマ割り込みの間隔を短くし、システム時間の経過速度を速くすることで、マルウェアの実行速度を低下させる。実装の詳細については、岡村らの論文[4] で説明されている。簡単に説明すると、インターフェース部から通知された CR3 の値を持つプロセスが、マルドメインの仮想 CPU にスケジューリングされ、かつその仮想 CPU が物理 CPU に割り当てられている間だけ、その仮想 CPU に配送するタイマ割り込み間隔を短くする。さらに、このタイマ割り込み発生時に、システム時間のエミュレートを行う。これは、ハイパーバイザとマルドメインの共有メモリ内に存在するシステム時間を表す変数を書き換えることで実現する。

5. 現状と今後の課題

現在は KVM の中に、マルウェアを検知するシステムの実装を行っている段階であり、KVM 上でテスト OS[2] を動作させての実験では `wrmsr` と `rdmsr` をフックすることで MSR を書き換えて `SYSENTER` をトラップすることに成功している。マルウェアの具体的な解析方法や検知方法は検討中である。

今後の課題は、まずはシステムコールの解析方法を決定し、簡易的なマルウェア検知機構を完成させることである。その次に実行抑止機構を KVM に実装し検知機構と連携させ、提案システムの有効性を評価したい。

参 考 文 献

1) KVM: Kernel Based Virtual Machine. `http://www.linux-kvm.org/page/Main_page`.
2) tinyos. `http://github.com/ddk50/tinyos`.
3) Xuxian Jiang, Xinyuan Wang, and Dongyan Xu. Stealthy malware detection through vmm-based out-of-the-box semantic view reconstruction. In *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 128–138. ACM, 2007.
4) Keisuke Okamura and Yoshihiro Oyama. Controlling the speed of virtual time for malware deactivation. In *Proceedings of the Asia-Pacific Workshop on Systems*, p. 6. ACM, 2012.