IPA

2019 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

田中 邦裕（さくらインターネット株式会社 代表取締役社長）

2. クリエータ氏名

森 瑞穂（電気通信大学）

3. 委託金支払額

1,994,400 円

4. テーマ名

高速な VMI 機構を実装したバイナリ解析基盤

5. 関連 Web サイト

なし

6. テーマ概要

ソフトウェアが広く利用されるようになる中で、セキュリティは非常に重要なテーマであるが、その中でもマルウェアを解析し防衛することは非常に重要である。

本プロジェクトでは、マルウェアのバイナリを動的に解析するためのプラットフォーム『FastVMIX』を開発した。

マルウェアの解析には静的解析と動的解析の 2 種類の手法が存在するが、本プロジェクトではハードウェア仮想化支援技術を用いて、特権的にゲストのレジスタやメモリを読み書きできるという強みを活用し、ゲストから不可視の解析を行うためのプラットフォームを実装した。

実装にあたっては、ベアメタルハイパーバイザの BitVisor をベースとして、ゲスト空間でマルウェアを動作させることとした。

7. 採択理由

コンピューターが世界中に浸透し、ソフトウェアがその重要性を増している昨今、マルウェアやウイルスなどが含まれるソフトウェアの解析への需要が高

まっているが、解析している事を該当のソフトウェアに悟られずに高速かつ簡便に外部から動的解析することは難しい。本提案では、仮想環境内でその解析を行い、加えて CPU 時間を改ざんして解析中である事を悟られない実装を示しており、実際に実現できれば画期的な解析環境になると考えられる。セキュリティインシデントは永遠に無くならないが、ソフトウェアの解析面からインシデントを無くすために、プロジェクトの成果が活用される事を期待して採択した。

8. 開発目標

本プロジェクトでは、前節で述べたとおり、バイナリの動的解析プラットフォーム FastVMIX を、ベアメタルハイパーバイザの BitVisor をベースに開発することを目的とした。また機能として、高速なゲストの解析を行える機構、マルウェアの解析耐性機能の迂回機構を実装することも目的とした。

まず、高速なゲストの解析だが、これはコンテキストスイッチを極力抑えるシステムを構築する。コンテキストスイッチには、エミュレーションに必要なコンテキストスイッチと、解析時にゲストからハイパーバイザに制御が移るコンテキストスイッチの大きく二種類が存在する。本プロジェクトでは後者の解析時におけるコンテキストスイッチを極力減らすデザインで開発を進めた。

次に、マルウェアの解析耐性機能の迂回機構だが、今回はタイミング解析と呼ばれるマルウェアの機能を迂回することを目標とした。タイミング解析は実環境と解析環境とで発生する実行時間の差異から解析環境であると断定する機能である。これを、CPU 時間の偽装という手法で迂回する機能を提供することとした。

9. 進捗概要

FastVMIX の全体像を図 1 に示す。FastVMIX を構成するのは BitVisor ベースのハイパーバイザと、ハイパーバイザを操作したりカーネル空間での準備を行ったりするカーネルモジュール、カーネルモジュールを操作するためのフロントエンドコマンド、特殊なメモリ空間 1GB Huge Page で動作する PIC バイナリの大きく四つで構成されている。

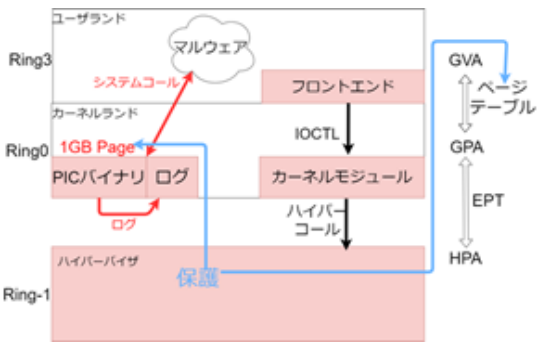

図 1：システムの全体像

FastVMIX は解析時におけるコンテキストスイッチを可能な限り廃したデザインになっている。これを実現するために、ゲストのセンシティブな動作をトラップし、ログを取る解析機構をゲスト内部に挿入した。システムのコンポーネントでは PIC バイナリにあたる。PIC バイナリはゲストの内部（Ring0）で動作する。すると、ゲスト内部（Ring3 or Ring0）に侵入したマルウェアがシステムへ攻撃を行うことができる。これを防ぐためにハイパーバイザがシステム全体を保護する。

保護の手法としてページテーブル保護とメモリマップの動的変更をハイパーバイザに実装した。

攻撃者はゲスト内部のページテーブルを操作することにより、システムの機能を停止させることができる。例えば PIC バイナリやログをアンマップしたりすることが可能である。そこで FastVMIX では PIC バイナリとログのページテーブルエントリを保護する。具体的な手法としてはページテーブルが切り替わるたびに、該当するエントリが書き換わっていないか監視する。例えば MOV-TO-CR3 な操作をハイパーバイザでトラップし、ページテーブルエントリを監視する。しかし、ページテーブルエントリの保護はプロセスが切り替わるたびに監視を行わなければならず、オーバーヘッドの要因となりうる。そこで 1GB Huge Page を採用し、PIC バイナリとログをこの特殊なページにマップすることでオーバーヘッドを解消した。ページテーブルは通常 4 レベルページングが採用されており、エントリが階層ごとに別れている。通常の 4KB もしくは 2MB ページのエントリを保護する場合、下のレベルだけでなく、自身より上のレベルまで保護してやる必要がありコストが増大する。しかし、1GB Huge Page ならば保護するレベルが 2 レベルで済む。したがって、1GB Huge Page の採用により保護するエントリ数の削減が達成できた。

次に、メモリマップの動的変更による保護について述べる。図 1 で示したように通常のメモリマップと解析用のメモリマップの二種類を用意し、通常は通常用のメモリマップ（解析コード PIC バイナリがアンマップ、ログは読み出しのみ）を使用し、解析時のみに解析用のメモリマップ（PIC バイナリがマップされ、ログが読み書き可能、カーネルのコード領域は読み込みのみ）を使用する。メモリマップにはゲストの物理メモリアドレスとホストの物理メモリアドレスの変換テーブルである EPT（Extended Page Table）を用いた。また、メモリマップの動的変更には EPTP Switching を用いて、コンテキストスイッチなしでメモリマップの動的変更が行えるようにした。

最後に、CPU 時間の偽装についてだが、RDTSC 命令をトラップし、解析にかかった時間分もとに戻す実装をハイパーバイザに行った。ベンチマークプログラムではこれが有効に働いていることが確認できた。

10. プロジェクト評価

クリエータ自らがセキュリティ分野やCPUのアーキテクチャに対する知識が深く、それを組み合わせたプロジェクトであり、比較的スムーズに進むかと思われたが、実装方式や先行研究に対する未踏性などの部分で開始当初は非常に苦労が多かった。

しかしながら、先行する研究室においてのアドバイスや、実装の方向性に対するディスカッションなどを経て、プロジェクト期間中に多くのことを吸収し、クリエータ自身も成長し、最終的に目的とする実装の大部分を完了させる事ができ、大変大きな成果が上がったものと考える。

11. 今後の課題

現在の PIC バイナリでは機能が限られているため、より詳細な解析が行えるPIC バイナリの開発も行う必要がある。加えて、Linux 以外に Windows など他OS への対応も行う必要があると考えられる。

またステルス性能の向上、マルウェアによる高度な動的解析機能の迂回などの機能の実装も期待される。

今後の展望として、ソースコードは近日中にオープンソースソフトウェア化される。これに伴い他の開発者とともに開発を加速することができる。

また、これまでにない VMI 機構は新規性にも優れるため、学会への論文投稿についても期待したい。