ソフトウェアのインストールを必要としない NIC 型セキュリティ機構

― Bubo：マルウェアの先を行く動的解析フレームワーク ―

1. 背景

マルウェアの検知、解析などを行うセキュリティ機構にとって、それらのセキュリティ機構がマルウェア側からその存在を認知されにくいこと（隠密性）、それらのセキュリティ機構が取得する情報が取得してくる情報が改ざんされにくいこと（透明性）等が重要であることが指摘されている。現状、最も隠密性、透明性の高いセキュリティ機構の実現手法の一つとして、ハードウェアを用いるものが提案されている。これらのうち多くは、専用ハードウェアを用意し、監視対象のホストにインストール 、DMA（Direct Memory Access）により取得したメモリ上のデータを監視を行うホストまで送信することで実現されている。

しかし、従来の手法には、監視対象のホストにソフトウェア的変更を要求し、隠密性、透明性を損なってしまう、DMA によるメモリ取得はアトミック性が低く、I/O などのリアルタイム性の高い情報を取得することが出来ない、Volatility、Rekall などの既存の有力メモリ解析ソフトウェアとの連携が弱くユーザが取得できる情報が限られるといった問題があった。

2. 目的

本プロジェクトの具体的な目標は、一切のソフトウェア的変更を加えることなく、物理マシン、その上で動作する仮想マシンのメモリ空間、更に I/O を監視し、高い隠密性、透明性をもったセキュリティ機構を既存の有力オープンソースソフトウェアと連携して手軽に実装できるハードウェアベースのフレームワーク、Bubo の開発である。

Bubo の開発により、ユーザが手軽に高い隠密性、透明性を持ったセキュリティ機構を実装できるようになり、よりセキュアな世界の実現に貢献することを最終的な目的とした。

3. 開発の内容

本プロジェクトで実装したフレームワークは大きく分けて 3 つのコンポーネントからなり、監視対象のホスト（Target）と、監視を行うホスト（Monitor）のそれぞれ 2 つのホストの中で使用される（図 1）。

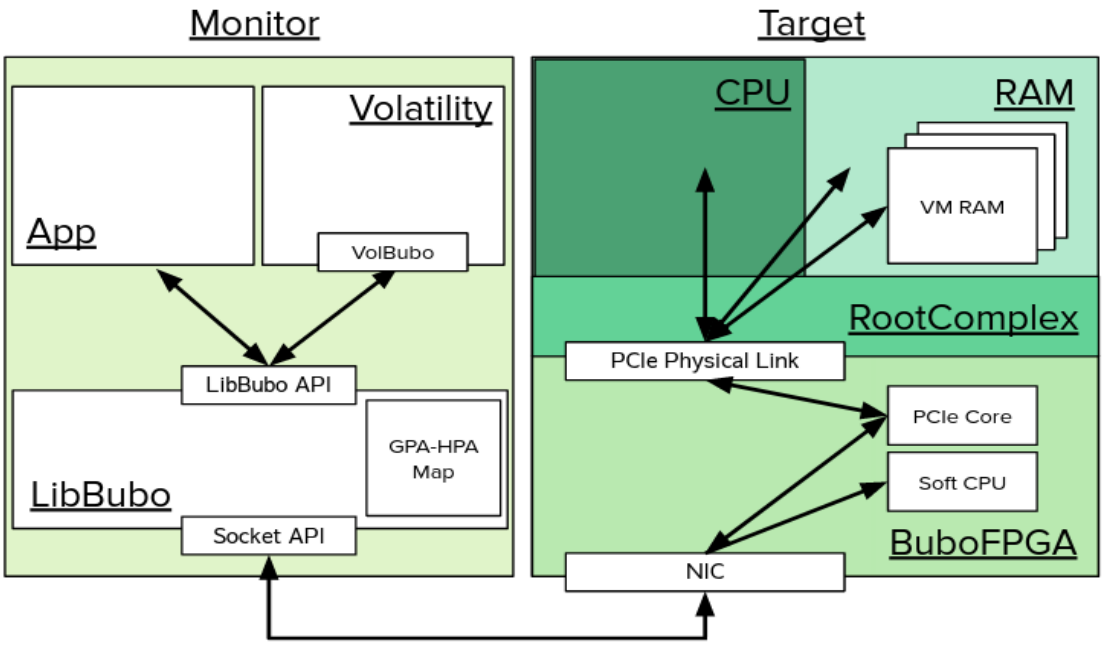

図 1: Bubo の構成図

(1) BuboFPGA

BuboFPGA は Target のメモリ空間の取得と、PCI Express デバイスのエミュレーションを実現するためのハードウェアである。NIC を通じて Monitor から受信した UDP パケットで制御される。Xilinx Kintex-7 KC705 FPGA 評価ボード上に PCI Express デバイスとして実装された。また、サイドチャネル攻撃を防止するために、ソフト CPU を搭載している。

メモリ空間の取得は、BuboFPGA が NIC を通じて受信した UDP にカプセリングされた Transaction Layer Packet をそのまま Target のルートコンプレックスまで中継し、DMA を行い、返されたメモリ上のデータを含む Transaction Layer Packet をそのまま Monitor まで UDP にカプセリングして返すことで実現した。

PCI Express デバイスのエミュレーションは、後述する LibBubo との連携により実現されている。BuboFPGA はこのために、自身の PCI コンフィグレーション空間を Monitor から UDP パケットで読み書きできる仕様となっている。また、自身の PCI コンフィグレーション空間、特定の BAR がマップするデバイスメモリ空間への Target からの Transaction Layer Packet による読み書きを、全て Monitor に Transaction Layer Packet を UDP にカプセリングし、送信することで中継している。これにより PCI Express デバイスの Monitor 側でのソフトウェアによるエミュレーションが可能となる。

ソフト CPU は Target のルートコンプレックスに Transaction Layer Packet を発行することが出来、その結果として受け取ったデータを処理した後に Monitor に送信することが出来る。また、LibBubo からの割り込みで一定の制御を行うことが出来る。

これにより、サイドチャネル攻撃への対策を行っている。

(2) LibBubo

LibBubo は BuboFPGA を通じて物理マシン、その上で動作する仮想マシンそれぞれのメモリ空間の取得、PCI Express デバイスのエミュレーションを実現するためのソフトウェアライブラリである。

BuboFPGA は UDP パケットで制御できる。物理マシンのメモリ空間を取得するためには、Socket API のラッパーを用意し、TLP を UDP 上にカプセリングして BuboFPGA の NIC へ送信するのみで良い。仮想マシンのメモリ空間の取得は、はじめに物理マシンのメモリ空間を取得し、ゲスト物理アドレスからホスト物理アドレスへの変換を担うページテーブルである EPT（Extended Page Table）を復元することで実現される。これにより、ゲスト物理アドレスを用いた透過的なメモリ取得が可能となる。EPT を復元する過程において、VMM が VM を制御するための構造体である VMCS（Virtual Machine Control Structure）を復元する必要があるが、その詳細なレイアウトはマイクロアーキテクチャ毎に異なる上に非公開である。LibBubo には、これらの問題を解決する、先行研究の手法を改良した VMCS のリバースエンジニアリングツールも含まれている。更に、BuboFPGA と連携して PCI Express デバイスのエミュレーションを行うための API、BuboFPGA 上のソフト CPU へ割り込みをかけ、制御を行うための API も提供されている。

(3) VolBubo

VolBubo は LibBubo、BuboFPGA を通じて取得できる、物理マシン、その上で動作する仮想マシンのメモリ空間上のデータを、Volatility への入力として提供するための Address Space Plugin である。これにより、Volatility プラグインを Target の物理メモリ空間や仮想メモリ空間にそのまま使用することが出来、ユーザがセキュリティ機構の実装を行うコストが大幅に低下する。

図 2 に実際に監視対象のホスト（Linux KVM）上で動作する仮想マシン（Windows）に対して Bubo 経由で Volatility プラグインを実行した様子を示す。

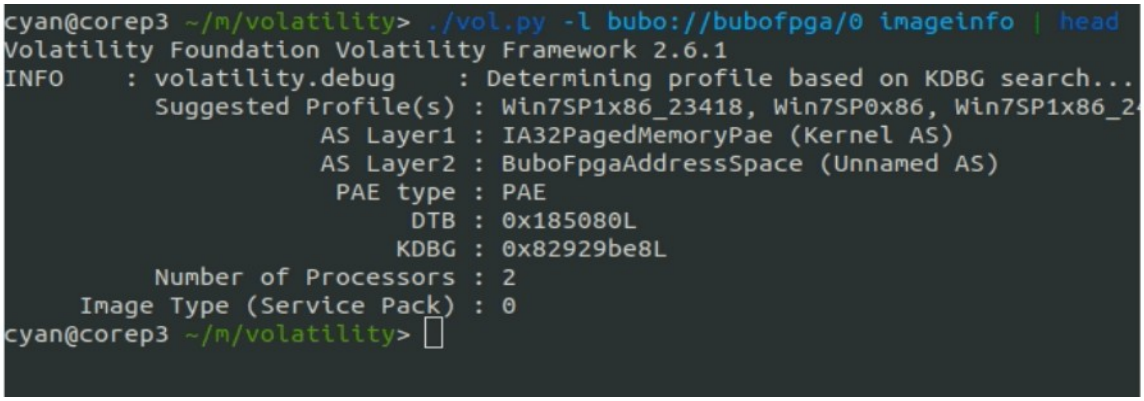

図 2: Bubo 経由のプラグイン実行

4.　従来の技術（または機能）との相違

これまでにも、専用のハードウェアを用いた DMA によるメモリ上のデータ取得、ディスク I/O の監視により、物理マシン、仮想マシンの上で動作するマルウェアを解析をできるシステムは存在した。しかし、監視対象のホストへのソフトウェア的な変更が要求されていた。また、取得できる情報も、メモリ上のデータ、SATA フレーム、それらを解析する専用のプラグインの出力と限られていた。また、サイドチャネル攻撃の可能性が想定されたものは存在しなかった。更に、システムを使用するためには多数のハードウェアを監視対象のホストに接続する必要があった。Bubo は監視対象のホストへソフトウェア的な変更を一切加えずに、DMA ベースでの物理マシン、仮想マシンのメモリ空間への透過的なアクセスを実現し、各メモリ空間に対して Volatility プラグインを使用することを可能にしている。更に、PCI Express デバイスのエミュレート機能により、任意の I/O を監視することを可能にした。また、ソフト CPU を BuboFPGA に搭載し、外部から割り込みベースで制御を行うことでサイドチャネル攻撃の対策も行った。これらの機能を実現するために必要な監視対象のホストへの変更は PCI Express デバイスをスロットに挿しこむことだけである。これらは全て世界初の成果である。

5.　期待される効果

本プロジェクトの開発成果により、ユーザは非常に高い隠密性、透明性を持ったマルウェア検知、解析にはじまるセキュリティ機構を Volatility プラグインとして素早く実装することが可能となった。Bubo は監視対象のホストの CPU リソースを一切消費しないため、従来のセキュリティ機構とも競合しない。よって、この成果を単体で、あるいは従来のセキュリティ機構と併わせて用いることで、より信頼性の高いセキュリティ機構を実現できる。また、この成果をきっかけに、セキュリティ機構のトラストモデルという課題により注目が集まり、更なる研究が実施され、その成果が社会に還元されることで、よりセキュアな世界が実現されることも期待している。

6.　普及（または活用）の見通し

本プロジェクトの開発成果は、強固なセキュリティを要求されるサーバ管理者や、セキュリティ研究者などを広く対象としたものである。普及のための第一歩として学会発表を予定している。発表を通じ、実用化に向けた更なる議論が出来ることを期待している。

7.　クリエータ名（所属）

上田 侑真 （慶應義塾大学 総合政策学部）