仮想マシン上の特定プロセス向け通信のフィルタリング

安積武志† 光来健一†† 千葉滋†

1. はじめに

近年データセンタが VM を用いた仮想化ホスティングサービスを提供するようになったことで、ユーザが OS を管理するようになった。これに伴って充分にスキルの高くないユーザが OS を管理するケースが増えてきている。OS が正しく管理されていないと攻撃者の侵入を許してしまい、VM が踏み台攻撃に利用される可能性がある。その場合ににはデータセンタ管理者が通信を遮断すべきであるが、従来は VM 内の OS 情報を利用することができなかったため、大雑把な通信制御しかできなかった。これは VM に対してのサービス可用性を低下させる原因となる。

本稿では、VM モニタからゲスト OS のパケットをプロセス単位、ユーザ単位でフィルタリングすることを可能にする xFilter を提案する。xFilter は VMM からゲスト OS 内のプロセス情報を取得し、プロセスの ID や所有者の情報を用いてパケットフィルタリングを行う。VMM からゲスト OS のプロセス情報を取得するために、VMM のメモリにゲスト OS のメモリをマッピングする。ゲスト OS の情報を用いることで、指定した特定プロセスやユーザの通信のみを遮断することができ、その他のプロセスやユーザは通信を行うことができる。我々は Xen[1] 上に xFilter を実装した。

以下、2 章では既存の VM の通信制御の問題点について述べる。3 章では提案するフィルタリングシステムの詳細と実装を述べる。4 章では実験の結果を示す。5 章では関連研究について触れ、6 章で本稿をまとめる。

2. セキュリティとサービス可用性

従来のホスティングサービスではデータセンタが OS を用意し、ユーザはその上で動かすサービスだけを管理すればよい場合が多かった。VM への攻撃を防ぐには、ユーザは OS に最新のセキュリティパッチを適用し続け、ファイアウォールのルールなどを正しく設定する必要がある。しかし、OS を管理するユーザのスキルが低い場合には、システムの脆弱性を利用して攻撃者の侵入を許してしまう危険性がある。

そこで、我々はデータセンタの管理者が VM への攻撃に対処できるようにすべきであると考える。VM のユーザは常時管理しているとは限らず、攻撃が検出された時に即座に対処できるとは限らない。また、即座に対処し始めたとしても、ユーザのスキルが低いと問題の解決に時間がかかってしまう。

しかし、VMM からはゲスト OS 内の情報を利用できないため、通信の制御は大雑把になってしまい、VM に対してサービス可用性を低下させてしまう。

3. xFilter

3.1 xFilter

我々は、VMM からゲスト OS のパケットをプロセス単位、ユーザ単位でフィルタリングすることを可能にする xFilter を提案する。xFilter は、VMM からゲスト OS の内部情報を取得し、プロセスの ID や所有者の情報を用いてパケットフィルタリングを行う。これにより、VM に対するサービス可用性を上げることができる。

xFilter のフィルタリングの手順について説明する。まず VMM がパケットを受信したとき、ゲスト OS 内のどのプロセスがそのパケットを送信したかを調べる。ゲスト OS のメモリを VMM のメモリにマッピングすることによって、VMM から直接参照する。取得する情報は各プロセスのユーザ ID、名前、行っている通信のポート番号と IP アドレスの組である。VMM はパケットの到着毎にプロセス構造体が持つ通信の情報 (ポートや IP アドレス) と比較する。送信元のプロセスの ID またはユーザの ID がルールにマッチすれば送信を拒否する。

3.2 実　　装

xFilter の実装には、VMM として Xen[1]、対象 OS

† 東京工業大学
Tokyo Institute of Technology
†† 九州工業大学
Kyushu Institute of Technology

として Linux を使用した。今回 xFilter は、ドメイン 0 上のユーザランドプロセスとして実装した。

ドメイン 0 からドメイン U のプロセスを調べるために、task_struct 構造体をたどっていく。Linux ではプロセスに関する情報は task_struct 構造体に格納されている。

ドメイン U のメモリにアクセスするために、ドメイン 0 からドメイン U のメモリを操作する機構[4] を利用した。この機構を用いると、ドメイン U の仮想アドレスから Xen が管理するメモリフレーム番号を取得することができる。取得したメモリフレームをドメイン 0 のプロセスのアドレス空間に割り当てることで、ドメイン 0 のプロセスからドメイン U のメモリにアクセスすることができる。

現在のところ、xFilter によるパケットフィルタリングはドメイン 0 の iptables のルールを追加するという方法で実装している。Xen ではドメイン U の通信はすべてドメイン 0 を通過するため、ドメイン 0 のファイアウォールで制御が可能である。

4. 実　　験

ベンチマークとして httperf[3] を用いて通信の性能を測定した。実験対象の計算機として、Athlon(tm) 64 Processor 3500+ の CPU、メモリ 1GB の計算機を使用した。VMM としては、Xen3.1.0 、VMM 上で動く OS には Linux2.6.18 を用いた。ドメイン 0 にはメモリを 512MB、ドメイン U にはメモリを 256MB 割り当てた。

今回、フィルタリングにマッチしないルールを設定して、間隔を変えてポーリングを行った。条件は毎秒 150 リクエストを送り、総リクエスト数 100000 とした。実験は、xFilter を使わない場合、ポーリング間隔を 5 秒、3 秒、2 秒、1 秒にした場合の 5 通りで行った。実験結果は以下の表のとおりである。

表 1 コネクションの処理時間 (ミリ秒)

polling 間隔	なし	5 秒	3 秒	2 秒	1 秒
最小	0.2	0.2	0.2	0.1	0.1
平均	0.4	0.4	0.5	0.5	0.7
最大	31.0	31.2	33.4	31.0	39.1
中央値	0.5	0.5	0.5	0.5	0.5
標準偏差	1.0	1.4	1.7	1.9	2.5

ポーリングの間隔を短くすると、平均処理時間は長くなった。これは、ドメイン U のメモリを調べるときにドメイン U を停止しているためであると思われる。さらに、最小、最大、中央値はほぼ変わっていないにもかかわらず、ポーリング間隔が短くなるにつれて標準偏差が大きくなっていることを見ても、最大値に近い大きな値を取る回数が多くなっていることが分かる。つまり、ドメインの停止が処理にかかる時間に影響している。また実験所要時間は全て同じであり、つまりスループットには影響はなかった。これはサーバにまだ余裕があったためと思われる。

5. 関 連 研 究

Antfarm[2] は、VMM 上からドメインに手を加えずにプロセスの状態を取得する技術である。ドメイン上の OS のソースコードに手を加えずに、OS の情報を取得する点は本研究と同じである。しかし、取得できる情報はプロセスの状態の変化だけであり、何のプロセスかまでは分からない。ドメイン上の OS に依存せず、Linux 以外でも通用する技術であるという点で本研究より優れている。

6. ま と め

本稿では VMM からゲスト OS 内の情報を取得し、利用できるシステムである xFilter を提案した。ゲスト OS 内の情報の取得は、ゲスト OS のメモリのマッピングによって実現した。また、パケットのフィルタリングには iptables を用いた。xFilter を用いることで、データセンタ管理者が OS 内の情報を利用してきめ細かい通信制御行うことができ、サービス可用性をできるだけ保つことができるようになった。

参 考 文 献

1) Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 164–177, 2003.

2) Stephen T. Jones, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. Antfarm: tracking processes in a virtual machine environment. In *USENIX-ATC'06: Proceedings of the Annual Technical Conference on USENIX'06 Annual Technical Conference*, pp. 1–1, 2006.

3) David Mosberger and Tai Jin. httperf-a tool for measuring web server performance. *SIGMETRICS Perform. Eval. Rev.*, Vol. 26, No. 3, pp. 31–37, 1998.

4) 田所秀和, 光来健一, 千葉滋. 仮想マシン間にまたがるプロセススケジューリング. 情報処理学会論文誌:コンピューティングシステム. ACS 23, 2008.