平成 22 年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	09237206	学生氏名	塩田　裕司
論文題目	サスペンドした仮想マシンのオフラインアップデート		

1　はじめに

仮想マシンはサーバ用やデスクトップ用として使用されているが、どちらも複数の仮想マシンを並行で使用し、使用していない仮想マシンはサスペンド状態（オフライン）にしていることが多い。そのため、使用用途によっては仮想マシンを長期間使用しない場合がある。サスペンド状態の仮想マシンはセキュリティアップデートが行われておらず、最新の状態でないままレジュームすることになる。このような仮想マシンは、ネットワークに接続した途端に攻撃される危険性が高い。従来は仮想マシンをレジュームした後でセキュリティアップデートを行うことが多いが、ネットワークに接続してアップデータをダウンロードしなければならず、アップデートにも時間が掛かるため、アップデート中に攻撃を受ける危険性が高い。

本研究では、オフラインでセキュリティアップデートのエミュレートを行い、仮想マシンをレジュームした後で行う処理を最小限にする OUassister を提案する。

2　仮想マシンのアップデート

仮想マシンを用いると一台の計算機上に複数の計算機を仮想的に作成することができるため、長期間使われない仮想マシンが多く存在している。仮想マシンは使う必要がある時だけ動かすという使い方が一般的なためである。例えば、サーバは負荷によって仮想マシンの数を調整できるように予備の仮想マシンを用意しており、デスクトップでは別の OS を使いたい時のみ仮想マシンを使用する。

長期間サスペンドして停止させていた仮想マシン内の OS やアプリケーションには脆弱性が見つかっていることが多い。このような仮想マシンをレジュームして再開するとその脆弱性を利用した攻撃を受ける可能性が高く危険である。同様に、仮想マシンをレジュームした後でセキュリティアップデートを行うのも危険である。図 1 のようにネットワークに接続してアップデータをダウンロードしなければならず、ネットワークからの攻撃を防ぐのが難しい。また、アップデートの処理に時間が掛かることも多く、その間に攻撃される可能性が高い。

そこで、仮想マシンを停止させたままアップデートする手法が提案されている [1]。この手法は、仮想マシンの仮想ディスクを直接更新することで、仮想マシンを再開した時に最新の状態に保つことができる。しかし、この手法は OS をシャットダウンさせた仮想マシンに対してのみ有効であり、サスペンドした仮想マシンに適用することはできない。サスペンド状態の仮想マシンの仮想ディスクを更新すると仮想ディスクが壊れるためである。

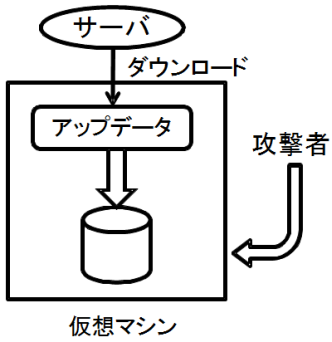

図 1　レジューム時の脆弱性

3　OUassister

本研究では、サスペンドした仮想マシンのオフラインアップデートを可能にする OUassister を提案する。オフライン時に仮想マシン内の OS やアプリケーションのアップデートのエミュレーションを行っておき、仮想マシンをレジュームした直後にエミュレーション結果の反映を行う。オフライン時にアップデータのダウンロードを行い、アップデートの実行を可能な限り終わらせておくことによって、レジューム直後に行わなければならない処理を最小限にすることができる。その結果、アップデート中に攻撃を受ける可能性を減らすことができる。

3.1　アップデータ実行のエミュレーション

OUassister では、Transcall [2] を用いてアップデートのエミュレーションを行うための実行環境を構築する。Transcall は図 2 のような VM シャドウと呼ばれる実行環境を提供して、ホスト OS から仮想マシン内の情報を参照可能にするシステムである。Transcall は仮想マシンの仮想ディスクをホスト OS 上にマウントし、仮想ディスク内のファイルへのアクセスを可能にする。これにより、VM シャドウの中のアップデータは仮想マシン内で動作しているのと全く同じようにファイルを扱うことができる。

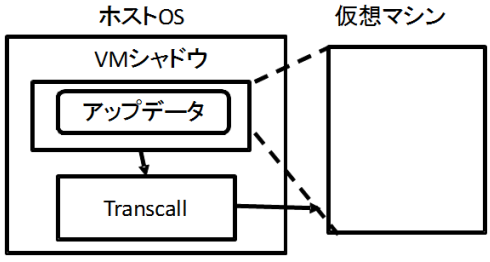

図 2　Transcall によって作られる実行環境

Transcall はアップデータのダウンロードやアップデートの実行中に発行されるシステムコールをトラップし、必要に応じてエミュレーションを行う。例えば、アップデータをダウンロードする際に使われるネットワーク用のシステムコールについては、エミュレーションを行わずにホスト OS に実行させる。一方、仮想マシン内の OS の情報を取得するシステムコールについては、仮想マシンのメモリ上の情報を取得することでエミュレーションを行う。

3.2 ファイル更新のエミュレーション

アップデータが Transcall によってマウントされた仮想ディスクへの書き込みを行うとディスクが壊れてしまうため、aufs を用いて書き込みのエミュレーションを行う。aufs [3] は複数のディレクトリを透過的に重ねることができるファイルシステムである。それぞれのディレクトリはブランチと呼ばれ、読み取り専用または読み書き可能の属性を設定できる。重ね合わせた上側のブランチが優先して読み込まれ、一番上のブランチからファイルを読み出そうとすると、上から順にブランチを調べて最初に見つかったファイルを読み出す。また、ファイルを書き込む際には一番上の書き込み可能なブランチに書き込まれる。ファイルを削除する際にはホワイトアウトと呼ばれる特殊なファイルを書き込むことで下側のブランチのファイルを隠す。

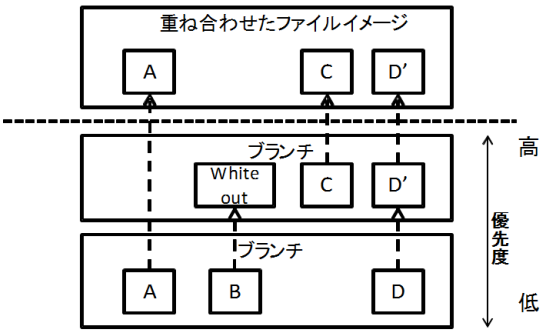

図 3　aufs によるディレクトリの重ね合わせ

OUassister では、仮想マシンの仮想ディスクをマウントしたディレクトリを読み取り専用、ホスト OS のファイル保存用のディレクトリを読み書き可能に設定し、ファイル保存用のブランチが上側になるようにマウントを行う。アップデート前にファイルを読み込むとファイル保存用のブランチにはまだ何も書き込まれていないため、仮想ディスクから読み込まれる。アップデータがファイルを更新すると、ファイル保存用のブランチに書き込まれ、仮想ディスクへの変更は行われない。アップデート後に再度、変更したファイルにアクセスすると、ファイル保存用のブランチにアクセスすることになるため、VM シャドウからはファイルが更新されたように見える。

3.3 エミュレーション結果の反映

仮想マシンがレジュームされた時、セキュリティアップデートによって更新されたファイルを仮想マシンに転送し、実際に仮想ディスクに更新を反映させる。aufs を用いることで、アップデータによって変更されたファイルは全てホスト OS の保存用ディレクトリに書き込まれる。そこで、このディレクトリを tar コマンドで固め、図 4 のように scp コマンドで仮想マシンに送る。次に、ssh コマンドを使って仮想マシン上で tar コマンドを実行することで、変更されたファイルを展開する。これらの処理にはネットワークを利用するが、インターネットとの接続は禁止した状態で行うことで、外部からの攻撃を防ぐことができる。

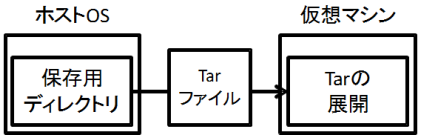

図 4　更新ファイルの反映

4 実験

従来のアップデートと OUassister を用いたオフラインアップデートについて、仮想マシンをレジュームした後のオンライン処理時間の比較を行った。実験に使用したマシンは、CPU Intel Core2 Quad 2.83GHz、メモリ 4GB であった。Xen 3.4.0 を用い、ホスト OS で Linux 2.6.32.25、ゲスト OS で Linux 2.6.27.24 を動作させた。ホスト OS、ゲスト OS 共に Ubuntu10.04 をインストールした。

bcc パッケージのインストールを仮想マシンのレジューム後に行った場合と OUassister を使用した場合について、オンラインでの処理時間を測定した。表 1 に測定結果を示す。この表から OUassister を用いた場合のオンライン処理時間の方が短いことが分かる。ダウンロードやパッケージの処理などをオフライン時に行うことによって、OUassister ではオンライン時の処理時間を削減することができている。

表 1　オンライン処理時間

	処理時間 (秒)
従来のアップデート	43.3
OUassister によるアップデート	7.23

5 おわりに

本研究では、サスペンドした仮想マシンのオフラインアップデートを行う OUassister を提案した。OUassister は、オフライン時にアップデータ実行のエミュレーションを行い、aufs を使って更新ファイルを抽出しておく。仮想マシンのレジューム後にエミュレーション結果の反映を行うことで、仮想マシンのアップデートを短時間で完了させる。今後の課題は、アップデータによるファイルの削除に対応し、ネットワークを使用せずにエミュレーション結果の反映を行えるようにすることである。

参考文献

[1] Shavlik. Netchk protect. http://www.shavlik.com/.
[2] 飯田 貴大 , 光来 健一 . 仮想マシンを用いた既存 IDS のオフロード, 2010. SWoPP 金沢 2010.
[3] Junjiro R. Okajima. aufs. http://aufs.sourceforge.net/.