平成 23 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	08237050	学生氏名	土田　賢太朗
論文題目	ファイルキャッシュを考慮したディスク監視のオフロード		

1　はじめに

近年、インターネットに接続されたサーバへの攻撃を検知するための侵入検知システム（IDS）の重要性が高まっている。IDS はディスクを監視し、攻撃の兆候が見つかると管理者に通知を行うが、IDS 自身が攻撃を受けた場合、監視を行うことができなくなってしまう。この問題を解決するために、仮想マシンを用いて IDS をオフロードする手法が提案されている。この手法は、監視対象システムを仮想マシン上で動作させ、別の仮想マシン上で IDS を安全に動かすことができる。しかし、オフロードした IDS は仮想ディスクを直接監視するため、監視対象 OS 内のファイルキャッシュ上にあるファイルは監視できなかった。そのため、ファイルキャッシュ上のファイルはディスクに書き戻されない限り監視することができず、攻撃者は IDS に検知されずに不正ファイルを使うことができていた。

本研究では、オフロードした IDS が仮想ディスクとファイルキャッシュを統合し監視を行えるようにする CacheShadow ファイルシステムを提案する。

2　ファイルキャッシュを利用した攻撃

仮想マシンを用いた IDS のオフロードは、監視対象のシステムを仮想マシン上で動作させ、IDS だけを別の仮想マシンで動作させる手法である。監視対象を動作させる仮想マシンをサーバ VM、IDS を動作させる仮想マシンを IDS-VM と呼ぶ。サーバ VM では IDS を動作させる必要がなくなるため、侵入されたとしても IDS を攻撃されることはない。一方、IDS-VM では IDS 以外のサービスをできるだけ動作させないようにすることで、攻撃を受けにくくする。オフロードされた IDS はサーバ VM の仮想ディスクや仮想ネットワークを監視することで攻撃を検知する。

従来、オフロードされた IDS は仮想ディスクを監視する際にファイルキャッシュを考慮していなかった。ファイルキャッシュはディスクから読み込んだファイルを一時的に保持しておくための領域であり、OS のメモリ上に作成される。アプリケーションはファイルキャッシュ上のファイルを読み書きすることで高速にファイルアクセスを行うことができる。アプリケーションがファイルを書き換えるとまず、ファイルキャッシュ上のファイルが更新され、一定時間が経過した後などにディスクへ書き戻される。IDS を IDS-VM にオフロードするとサーバ VM の仮想ディスクを直接参照して監視を行う。そのため、サーバ VM 内のファイルキャッシュ上にあるファイルに関しては監視することができなかった。

このように IDS オフロードではファイルキャッシュの監視を行っていなかったため、ファイルキャッシュ上の不正ファイルをディスクに書き戻させないという攻撃が可能になる。例えば、Linux では管理者権限を奪うだけでファイルキャッシュの書き戻しまでの時間を長く設定することができる。その上で図 1 のように、攻撃者がファイルを不正に書き換えても、オフロードした IDS はそれを検知することができない。その一方で、サーバ VM 内のアプリケーションにはファイルキャッシュ上の不正なファイルを使わせ続けることができる。

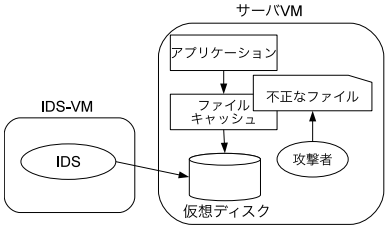

図 1　ファイルキャッシュを利用した攻撃

3　CacheShadow ファイルシステム

本研究では、オフロードした IDS がサーバ VM 上の仮想ディスクとファイルキャッシュを統合して監視を行えるようにする CacheShadow ファイルシステムを提案する。CacheShadow ファイルシステムは図 2 のように IDS-VM 上で動作し、サーバ VM のメモリからファイルキャッシュに関する情報を取得する。ファイルキャッシュ上にファイルが存在する場合には、それを IDS-VM 上の IDS に返す。ファイルキャッシュ上に存在しない場合は、サーバ VM の仮想ディスク上のファイルを返す。このようにして、サーバ VM 上で IDS を動作させる場合と同じ監視結果をより安全に得ることができる。

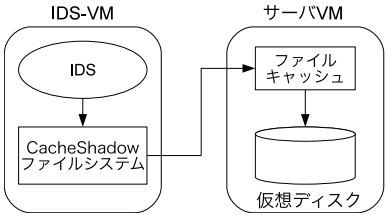

図 2　システムの構成

3.1　ファイルキャッシュ情報の取得

CacheShadow ファイルシステムはサーバ VM の OS のメモリを解析することでファイルキャッシュに関する情報を取得する。しかし、ファイルキャッシュに使われているデータ構造を直接解析するのは容易ではない。Linux ではファイルごとにファイルキャッシュを管理しているため、まずファイルのパス名を一つずつたどってファイル構造を見つける必要がある。さらに、ファイルキャッシュは Radix Tree と呼ばれる

データ構造でファイルブロックを管理しているため、目的のファイルブロックを探すのはかなり複雑で時間がかかる。

そこで、サーバ VM の物理メモリを管理しているページ構造体を解析することでファイルキャッシュを見つける。ページ構造体には対応するメモリページがどのような用途で使われているかという情報が格納されている。まず、サーバ VM のページ構造体の配列が置かれているメモリを IDS-VM にマップする。物理メモリ 1 ページ 4KB であるので、サーバ VM に割り当てた物理メモリのサイズからページ構造体の配列のサイズを求めることができる。ページ構造体の配列の先頭の仮想アドレスは固定であり、OS のシンボル情報から取得することができる。得られた仮想アドレスをサーバ VM のページテーブルを参照することにより物理アドレスに変換してマップする。

次に、ページ構造体のフラグを参照して対応するメモリページがファイルキャッシュとして使われているかどうかの判別を行う。ページ構造体はメモリページの用途をフラグで管理しており、予約済みのページや OS 内のデータ構造用に使われるページなどのフラグがある。しかし、ファイルキャッシュかどうかを示すフラグは存在しないため消去法でファイルキャッシュとして使われているページを割り出す。あるメモリページが予約されていない、OS 内のデータ構造用に使われていない、など様々な条件を満たす場合に、ファイルキャッシュと判定する。

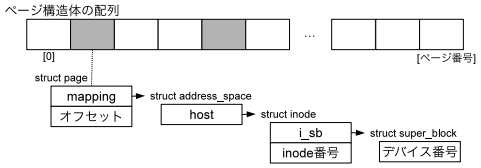

図 3　ファイルキャッシュ情報の取得

ファイルキャッシュと判定されたページについて、ページ構造体を解析することでキャッシュされているファイルが存在するデバイスの番号、ファイルの inode 番号、ファイルブロックのオフセットを取得する。デバイス番号と inode 番号は図 3 のようにデータ構造をたどることで取得することができる。オフセットはページ構造体の中に格納されている。このようにしてファイルキャッシュ情報を取得すると、デバイス番号、inode 番号、オフセットから対応するファイルキャッシュのページ番号を見つけられるようにハッシュ表に格納する。

3.2　仮想ディスクとファイルキャッシュの統合

CacheShadow ファイルシステムはシャドウファイルシステム [1] をベースとして、サーバ VM の仮想ディスクとファイルキャッシュを統合する。シャドウファイルシステムは、IDS-VM からサーバ VM の仮想ディスクにアクセスできるようにするためのファイルシステムであり、IDS-VM 上にサーバ VM の仮想ディスクをマウントすることで実現されている。シャドウファイルシステムはプログラムを実行する時には、改ざんを防ぐために IDS-VM 上のファイルにアクセスさせる。同様に、共有ライブラリや設定ファイルなど、IDS の実行に影響を及ぼすものについても IDS-VM 上のファイルにアクセスさせる。

CacheShadow ファイルシステムは FUSE を用いて実装し、ファイルの読み込み時に呼ばれる read 関数の中で読み込み元を切り替える。read 関数で読み込みを行うファイルのデバイス番号、inode 番号、オフセットを取得し、これらをキーとしてファイルキャッシュのハッシュ表を検索する。ハッシュ表に登録されていれば、ファイルキャッシュ上のファイルを読み込む。ハッシュ表から得られたページ番号に対応するページをマップし、read 関数が読み込んだデータを返すバッファにコピーする。ハッシュ表に登録されていない場合は仮想ディスク上のファイルを読み込む。

4　実験

IDS-VM からサーバ VM のファイルキャッシュ情報を取得するのにかかる時間を調べた。実験には Intel Core i7 870 の CPU、4GB のメモリを搭載したマシンを用いた。VMM として Xen 4.1.1 を動作させ、IDS-VM とサーバ VM の OS には Linux 2.6.39.3 を用いた。サーバ VM にはメモリを 1GB 割り当て、ファイルキャッシュの大きさを変化させて取得時間の測定を行った。結果を図 4 に示す。

ファイルキャッシュのサイズが増加すると解析するデータ構造の量も増えるため、取得時間も長くなった。ただし、Tripwire のように実行するのに長時間かかる IDS の場合には、最初にファイルキャッシュ情報を取得する時間の割合はそれほど大きくないと考えられる。

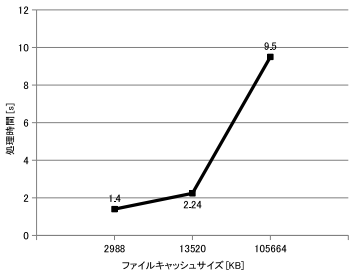

図 4　ファイルキャッシュ情報の取得時間

5　まとめ

本研究では、オフロードした IDS が正しくディスク監視を行えるようにするために仮想ディスクとファイルキャッシュを統合する CacheShadow ファイルシステムを提案した。CacheShadow ファイルシステムはサーバ VM のファイルキャッシュ情報を OS のメモリを解析して取得し、ファイルキャッシュ上にファイルが存在する場合にはそのファイルを返す。CacheShadow ファイルシステムを用いることにより、オフロードされた IDS による検知を回避するためにファイルキャッシュの書き戻し時間を長くするという攻撃を防ぐことができる。今後の課題は、FUSE を用いた実装を完成させ、ファイルキャッシュ情報の取得にかかるオーバヘッドを減らすことである。

参考文献

[1] 飯田貴大, 光来健一 . VM Shadow: 既存 IDS をオフロードするための実行環境. 第 119 回 OS 研究会, 2011.