仮想計算機を用いたファイルアクセス制御の二重化

滝澤 裕二　光来 健一　柳澤 佳里　千葉 滋

従来のシステムでは OS が認証やファイルアクセス制御を行い、不正アクセスを防いできた。しかし、OS カーネル自体に脆弱性が存在する場合には、攻撃者が認証をバイパスして管理者権限を取得したり、アクセス制御機構そのものが無効化されてしまう危険性がある。この問題を解決するために本稿では、仮想計算機（VM）を用いてファイルアクセス制御を二重化するシステム SAccessor を提案する。

SAccessor はユーザが作業を行う作業 VM とそれから安全に分離された認証 VM で独立したファイルアクセス制御を行う。作業 VM は認証 VM と通信してファイルアクセスを行い、認証 VM はポリシにしたがってダイアログを出して認証を行う。このようなシステムを実用的に利用可能するために我々は認証ダイアログ、ファイルキャッシュ、setuid されたプログラムに関する問題を解決した。

1 Introduction

ビジネスや個人利用でデスクトップ PC を使用し、重要なファイルをデスクトップ PC に保存されていることが多くなっている。しかしコンピュータにはセキュリティホールが数多く報告 [1] されておりデスクトップ PC は常にメールの添付ファイルや、トロイの木馬などを用いた攻撃を受ける可能性がある。従来は OS のファイルアクセス制御機構により、これらの攻撃の被害は限定されてきた。例えば、一般ユーザ権限で動く攻撃プログラムは管理者権限を必要とするファイルにはアクセスすることができない。

しかし、OS によるファイルアクセス制御は OS カーネルの脆弱性を利用した攻撃を受けると機能しなくなる可能性がある。近年、OS カーネルの脆弱性は数多く見つかっており、修正パッチが公開される前に行われるゼロデイアタックやデスクトップ PC にパッチを未適用であることが問題になっている。OS カーネルの脆弱性を攻撃されると、ユーザ認証をバイパスして管理者権限を奪われたり、

アクセス制御そのものを無効化される危険性がある。この問題を解決するために、仮想計算機 (VM) を用いてファイルアクセス制御を二重化する SAccessor を提案する。SAccessor はユーザが作業を行う作業 VM とそれから安全に分離された認証 VM で独立したファイルアクセス制御を行う。作業 VM は認証 VM と通信してファイルアクセスを行い、認証 VM はポリシにしたがってダイアログを出して認証を行う。このようなシステムを実用的に利用可能するために我々は認証ダイアログ、ファイルキャッシュ、setuid されたプログラムに関する問題を解決した。

以下、2 章では従来のファイルアクセス制御機構の問題点を示し、3 章ではその問題を解決するシステム SAccessor の基本的なアーキテクチャについて述べる。4 章ではシステムを構築する上での課題点とその解決策について述べ、5 章で SAccessor の安全性についての述べる。6 章で関連研究について述べ、7 章で本稿をまとめる。

Yuji Takizawa,Kenichi Kourai, Yoshisat Yanagisawa,Shigeru Chiba, 東京工業大学

2 従来のファイルアクセス制御

近年、デスクトップ PC に対する攻撃が増えてきている。例えば、攻撃者はメールに悪意のあるプログラムを添付し、メーラの脆弱性やユーザの不注意を利用してプログラムを実行させる。P2P ソフトウェアの場合は、ユーザの関心を誘うような名前のファイルを共有し、ファイルをダウンロードさせて実行させる。

従来、このような攻撃は OS による認証とファイルアクセス制御によって防がれていた。ユーザのログイン時に OS が認証を行い、認証が成功すると特定のユーザ ID とグループ ID が与えられる。OS のファイルシステムはアクセス制御リスト（ACL）を管理しており、アクセス要求にたいしてユーザ ID およびグループ ID と ACL を比較してアクセスの可否を決定する。

このように従来のシステムは OS のセキュリティは OS に依存していたが、OS にも脆弱性が見つかっている。例えば、ローカルユーザが管理者権限を取得できる脆弱性[2] が報告されている。また、OS カーネルをバッファオーバーフローさせて、任意のコードを実行させることができる脆弱性[3] も報告されている。このような脆弱性に対しては修正パッチが公開されるが、パッチが公開される前の攻撃（ゼロデイアタック）には対処できない。また、デスクトップ PC を使うユーザのセキュリティ意識が低い場合や、パッチを適用すると動作しなくなるアプリケーションがある場合には、パッチが適用されていないこともある。

このような OS 脆弱性を攻撃されると、従来のファイルアクセス制御は機能しなくなる可能性がある。悪意のあるプログラムに OS カーネルの脆弱性を利用して管理者権限を奪われると、管理者としてログインするための認証をバイパスされてしまう。さらに OS 自体の制御を奪われたとすると OS が提供しているファイルアクセス制御そのものを無効化されてしまう。

3 SAccessor

我々は、仮想計算機を用いてファイルアクセス制御を二重化するシステム SAccessor を提案する。SAccessor はユーザが作業を行う VM の OS が提供するファイルアクセス制御とは別に、ファイルサーバの役割を持つ VM でも独立してファイルアクセス制御を行う。分散ファイルシステムを用いればアクセス制御の二重化が可能だが、クライアントのマシンが攻撃されると両方のアクセス制御が無効化されたり、重要なファイルへのアクセスは別のマシンからしかできないといったの理由で実用的ではなかった。３章では SAccessor の基本的なアーキテクチャについて説明し、それを実用的にするための手法については４章で述べる。

3.1 アーキテクチャ

SAccessor では、図１のように２つの VM を同一マシン上で動かす。一つはユーザが作業を行う作業 VM であり、もう一つは作業 VM とは独立して認証とファイルアクセス制御を行う認証 VM である。SAccessor はデスクトップ PC を想定しており、ユーザの使うアプリケーションは作業 VM 上で実行され、ユーザには作業 VM の画面が見えている。ファイルは認証 VM によって管理されており、作業 VM は NFS を使って認証 VM のファイルにアクセスする。そのため、作業 VM は直接ファイルにアクセスすることはできない。SAccessor では作業 VM でのアクセス権の確認は作業 VM と認証 VM で二重に行われる。作業 VM でファイルアクセスを行う場合には、まず作業 VM 上の OS によりアクセス権をチェックし、その後、認証 VM 上でユーザ認証とアクセス権のチェックが行われる。OS の制御が奪われるまでは従来のファイルアクセスも機能する。認証 VM には管理者しかログインできず、認証 VM への攻撃を防ぐために外部からのアクセスも制限する。

作業 VM がファイルにアクセスするときにはポリシに応じて認証 VM がユーザ認証を行う。認証は作業 VM の画面上に認証ダイアログと呼ばれるダイアログボックスを出し、ユーザ名とパスワード入力させることにより行う。SAccessor は認証ダイアログで渡されたユーザ名を元にファイルアクセス制御で使用するユーザ ID を決定する。SAccessor では作業 VM 上の OS は攻撃されることを想定しているため NFS で渡されるユーザ ID は信用しない。

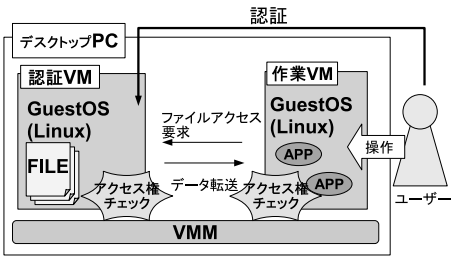

図 1 SAccessor のアーキテクチャ

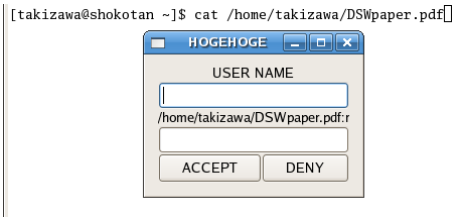

図 2 認証ダイアログ

3.2 認証ダイアログ

認証 VM は作業 VM からファイルアクセス要求を受け取ると必要に応じて作業 VM の画面の上に重ねて認証ダイアログを表示させる。画面構成は以下のように行う。認証 VM で X を起動させ、その画面をディスプレイに表示する。作業 VM でも X を起動し VNC を使って認証 VM の X 上にウィンドウを表示し最大化させる。認証ダイアログはこの VNC ウィンドウの上に重ねて認証 VM が表示する。図 2 は認証ダイアログが表示されている画面のスクリーンショットである。表示されている認証ダイアログは認証 VM によって出されたものであり、その後ろにあるターミナル画面は作業 VM の画面である。このようにするとユーザには作業 VM の画面だけが見え、認証 VM を意識させないようにすることができる。

ユーザはユーザ名とパスワードを認証ダイアログに直接入力して認証を行う。アクセスしようとしているファイルパスとアクセスモードがダイアログ中央に表示されている。これはどのファイルに対して認証を行うか確認するためである。ユーザが不正アクセスと判断した場合には、DENY ボタンをクリックすることで認証を失敗させることができる。

認証ダイアログは認証 VM の管理下にあるために、作業 VM からは操作することができない。そのため作業 VM 上で動く悪意のあるプログラムは認証を行うことができない。攻撃者に作業 VM の管理者権限を奪われても、攻撃者はここで認証を成功させることができないのでファイルアクセスすることはできない。また、入力した認証パスワードは作業 VM にわたらないので作業 VM にパスワードを盗まれることもない。パスワードが攻撃者に漏れないので、たとえ攻撃者が物理的にこの PC にアクセスできたとしても認証を成功させるのは難しい。

3.3 実装

SAccessor の実装には VMM として Xen [4]、作業 VM と認証 VM のゲスト OS として Linux をベースにカーネル内の NFS サーバを使用した。Xen において VM はドメインと呼ばれ、制御用のドメイン 0 とそれ以外のドメイン U が存在する。SAccessor は認証 VM、作業 VM をそれぞれドメイン 0、ドメイン U で動作させる。作業 VM は認証 VM のファイルシステムを NFS マウントして使用する。

4 Challenge

SAccessor を実用的に使うためには単に認証ダイアログを出せるようにするだけでは不十分であり、いくつか解決しなければならない問題がある。この章では SAccessor が問題解決のために用いた手法について説明する。

4.1 認証ダイアログの判別

SAccessor では認証 VM が認証ダイアログを作業 VM の画面上に重ねて表示されるため、作業 VM に偽のダイアログを出された場合、本物の認証ダイアログと区別するのが難しい。偽のダイアログにパスワードを入力してしまうと、攻撃者にパスワードを盗まれてしまう。

偽のダイアログと本物の認証ダイアログを区別するために、認証ダイアログのタイトルにシークレット文字列を表示させる (図 3)。シークレット文字列はあらかじめ認証 VM に登録しておく、ユーザのみが知っている文字列である。ユーザはダイアログに表示

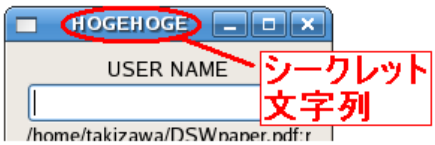

図 3　シークレット文字列

されたシークレット文字列を見て、ダイアログが認証 VM から出されたものであることを確認する。作業 VM に侵入した攻撃者にはこのシークレット文字列はわからないため、作業 VM から偽のダイアログを出されてもユーザは本物の認証ダイアログと区別することができる。作業 VM の画面をキャプチャしたとしても、認証ダイアログは認証 VM の管理下にあるので作業 VM に侵入した攻撃者からはシークレット文字列を見ることができない。

4.2　認証の頻度

ファイルアクセスに対して、認証 VM で毎回ダイアログを出して認証を行うのは実用的ではない。何故なら、ファイルアクセスの度にダイアログによる認証を行うと非常に多くのダイアログが出て著しく利便性が落ちてしまうからである。その一方で認証をログインのときにしか求めないようにすると作業 VM の制御を奪われた時の被害が大きくなる。

SAccessor では、ファイルやディレクトリを一つのグループとして定義することができ、それぞれのグループに対してアクセスを許可するパーミッションと認証の有効期間を設定できる。一度グループに対して認証を行ったら、認証の有効期間内ではグループのファイルへのアクセスに対する認証は省略される。また、従来システムのユーザ単位の認証と比べてよりきめ細かいアクセス制御が可能になる。たとえ認証をしたあとで作業 VM 上の OS の制御を奪われたとしても、作業 VM からは認証が有効になっているグループのファイルにしかアクセスできない。

図 4 は SAccessor のポリシ例である。この例では、`Report,ssh` の 2 つのグループが定義されている。<と>で括られているのはグループ名であり、ユーザが自由に決められる。グループの定義は</...>の部分

```
<Report> [3000]
/home/takizawa/Report/budget.xls  (rw)
/home/takizawa/Report/reference.pdf (r)
</Report>

<ssh>[10]
/home/takizawa/.ssh/id_rsa (r)
</ssh>
```

図 4　SAccessor ポリシ例

までである。`budget.xls,reference.pdf` が `Report` グループに、`id_rsa` が `ssh` グループに属している。ファイルパスの右に書いてあるアルファベットはアクセス権を表していて、グループ名の隣にある数字は認証の有効期間 (秒) を表している。設定できるパーミッションは読み込み (r)、書き込み (w)、追記のみ (a) である。このポリシを適用するとこれらのファイルにアクセスしようとしたときに認証を要求される。認証に成功すると、そのファイルの属するグループのアクセス権のみ与えられる。

ユーザが `budget.xls` を編集しようと認証を行うと `Report` グループに属するファイルへのみアクセスが可能になる。もし、`budget.xls` がウィルスに感染していて、OS の脆弱性を攻撃されたとしても、ウィルスによって `id_rsa` を読み書きされることはない。

4.3　作業 VM のファイルキャッシュ

作業 VM が認証 VM の NFS サーバから取得したファイルはファイルキャッシュとして保持され、その後のファイルアクセスはそのファイルキャッシュに対して行われる。ファイルキャッシュに残っていると、作業 VM は認証 VM での認証なしでファイルを読むことができてしまう。例えば図 4 のポリシにおいて、初めてユーザが `budget.xls` を読み込もうとしたときには認証を求められる。認証に成功すると実際にファイルを読み込み作業 VM のメモリにキャッシュする。認証の有効期間が過ぎた後に再び `budget.xls` を読もうとした場合、認証 VM と通信をして認証を行わなければならないが、作業 VM にファイルキャッシュが残っている場合には、認証 VM とは通信せず、作

業 VM のファイルキャッシュに直接アクセスする。

この問題を解決するために、認証 VM から作業 VM に対して、認証の有効期間が過ぎたことを通知し、作業 VM 上のファイルキャッシュをフラッシュさせる。作業 VM のゲスト OS の制御が奪われるとこの機構は機能しなくなるが、その時点で作業 VM 上に残っているファイルキャッシュは認証の有効期間内のものだけである。作業 VM はこのファイルにもともと認証なしでアクセスできるのでファイルキャッシュが残っていてもセキュリティが低下することはない。

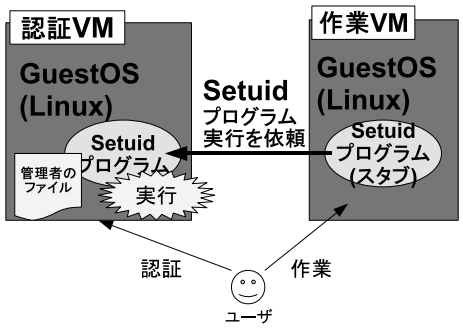

図 5 setuid されたプログラムの実行

4.4 setuid されたプログラムの実行

パスワード変更などのために setuid されたプログラムを実行する場合には一般ユーザが管理者のファイルにアクセスすることがある。通常のシステムの場合 setuid されたプログラムのユーザ ID はそのプログラムの所有者の ID として実行される。SAccessor における認証 VM での認証とファイルアクセス制御は作業 VM が攻撃されることを想定しており、作業 VM とは独立して行われるため作業 VM のユーザ ID は使用せず認証 VM での認証によってファイルアクセスの権限が決定される。そのため一般ユーザが setuid されたプログラムにより管理者のファイルにアクセスする場合には、管理者のパスワードが必要になってしまう。例えば一般ユーザがパスワード変更を行うときは passwd コマンドを実行する。passwd は/etc/shadow ファイルを修正しようとするが、このファイルは管理者のファイルであるので、SAccessor からは管理者のパスワードを入力しなければファイルのアクセス権が与えられない。

SAccessor では setuid されたプログラムによる管理者のファイルへのアクセスができるように、setuid されたプログラムは認証 VM が実行する。作業 VM の setuid されたプログラムは認証 VM と通信するプログラムに置き換えておき、そのプログラムを実行すると認証 VM に setuid されたプログラムの実行を依頼する。認証 VM は作業 VM からリクエストを受け取ると認証ダイアログによりユーザ認証を行い、認証に成功した場合には、対応するプログラムを認証 VM 上で実行する (図 5)。このときに入力するユーザ名とパスワードは一般ユーザのものである。このようにすることで、作業 VM に管理者のファイルへのアクセス権を与えずに setuid されたプログラムを実行することができる。ファイルにアクセスするのは認証 VM のプログラムなので、ファイルキャッシュが作業 VM に残ることもない。

setuid されたプログラムの実行は SSH で行う。プログラムに対する標準入出力は SSH でユーザと相互にやり取りし、ファイルに対する入出力は NFS のルートディレクトリを/に chroot して、setuid されたプログラムを起動させる。

4.5 システムファイルのアクセス制御

ログファイルや、設定ファイルなどはユーザが明示的にアクセスしなくても、システムによってアクセスされているものがある。システムによってアクセスされるファイルに対して認証ダイアログを表示させても、ユーザにはそれが攻撃者によるファイルアクセスなのか判断が難しい。されにファイルによっては管理者のパスワードが必要になることもある。

そこで SAccessor のポリシにはダイアログを出すかどうかを記述できる。ポリシに有効期間を記述しなければ認証は省略される。ダイアログを表示させない場合には認証は省略され、ファイルアクセス制御だけが行われる。例えばシステムの設定ファイルやプログラムファイルを読み込み専用、ログファイルを追記のみ認証なしで許可するというようなポリシを記述することで、システム起動時やロギング等に対してダイアログがでないようにでき、認証ダイアログを出さず

に設定ファイルの書き換えやログファイルの消去を防ぐことができる。実際システムファイルのアクセスパターンを調べたところ、ログファイルの追記、およびプログラムファイルのアクセスの読み込みだけであった。ログローテートなどの例外的な処理は setuid されたプログラムと同様に認証 VM に依頼して行う。

4.6 ポリシの登録

ポリシは認証 VM が保持しており、一般ユーザは認証 VM にログインすることはできず直接変更することはできないが、ポリシの登録は setuid されたプログラムと同様に認証 VM にポリシの登録を依頼して行うことで、一般ユーザが作業 VM から登録できる。ポリシを変更するときは、認証 VM からウィンドウを出し、認証 VM にユーザが直接ポリシを入力する。認証 VM は変更するポリシを画面に表示させ、ユーザに確認をさせてから認証を行う。

5 SAccessor の有効性

5.1 守れる攻撃

SAccessor を使うことで、バイナリファイルの書き換えや、起動スクリプトを書き換えてバックドアの設置、ログファイルの消去などの攻撃を防ぐことができる。図 6 はこれらの攻撃を防ぐためのポリシ例である。グループ emacs と emacs-update は emacs パッケージのファイルをまとめたグループである。このポリシでは、emacs に関するファイルの読み込みの認証は省略されるが、ファイルの書き換えを行うときには認証を求められる。このポリシを適用すると、ユーザが emacs を使って作業するときには認証なしで作業ができるが、

攻撃者が emacs のプログラムを書き換えようとすると認証ダイアログがでるために、その認証を拒否することでプログラムファイルの書き換えを防ぐことができる。管理者がパッケージのアップデートを行うときには、認証ダイアログがでるので認証を成功させればパッケージのアップデートを行うことができる。起動スクリプトに対しても同様のポリシを記述することで、攻撃者による書き換えを防ぐことができる。

SAccessor を使えばユーザの機密ファイルの改ざんや漏洩を防ぐことができる。図 7 はブラウザに記録されているパスワードファイルを守るポリシ例である。`signons.txt` には web サイトなどへのログインパスワードが保存されている。`signon.txt` は読み取り専用であり、ファイルアクセスには認証が必要である。ユーザがブラウザを開いていないのに認証を求められた場合には、攻撃者が `signon.txt` を盗み見ようとしているので認証を拒否すればファイルを見られることはない。ユーザがウェブサイトにログインしようとしたときに認証を成功させればよい。

setuid されたプログラムは認証 VM で実行されるため、作業 VM はそのプログラムが参照するファイルのアクセス権は必要ない。ポリシで認証の有効期間を-1 に設定すると、そのファイルはアクセス禁止にすることができる。図 8 は sudo のポリシ例である。sudo は
tt /etc/sudoers という設定ファイルを見るが、sudo は setuid プログラムなので`/etc/sudoers` にアクセスできる必要がない。作業 VM からのアクセスを全面禁止にすることで sudo のポリシが変更されるのを防ぐことができる。

5.2 limitation

ユーザが認証を成功させると作業 VM に対して認証の有効範囲内でのアクセス権が与えられ、認証の有効期間が切れるまでは認証なしでアクセスできる。従って、作業 VM の OS の制御が奪われる前に認証し、その認証が有効なファイルを攻撃者から守ることはできない。また、作業 VM の OS の制御が奪われた後に認証したファイルも守ることはできない。また、作業 VM にあるファイルキャッシュが攻撃者によって書き換えられると、ユーザは攻撃者によって書き換えられた偽のファイルの内容を読み込んでしまい、ファイルの改ざんと同じ被害を受ける可能性がある。

6 関連研究

SVFS [7] は VM を用いて、ファイルアクセス制御を攻撃対象の OS とは別の場所で行うシステムである。SVFS では Normal VM と Admin VM があり、重要ファイルへの書き込みは Admin VM からのみに

```
<emacs>
/usr/bin/emacs (r)
/usr/libexec/emacs/* (r)
/usr/share/pixmaps/emacs.png (r)
</emacs>

<emacs-update> [60]
/usr/bin/emacs (w)
/usr/libexec/emacs/* (w)
/usr/share/pixmaps/emacs.png (w)
</emacs-update>
```

図 6 プログラムファイルの書き換えを守るポリシ例

```
<firefox-password>[10]
/home/.../firefox/.../signons.txt (r)
</firefox-password>

<firefox>
/usr/bin/firefox (r)
      :
</firefox>
```

図 7 ユーザの機密ファイルを守るポリシ例

```
<sudo>[-1]
/usr/bin/sudo
/etc/sudoers
</sudo>
```

図 8 suid されたプログラムのためのポリシ例

制限されている。そのため、重要ファイルの書き換えを伴う管理タスクは、通常作業を行う VM とは別の AdminVM から行わなければならない。SAccessor では作業 VM から可能である。

Proxos [6] は通常のアプリケーションを実行する Commodity VM とは別に Private VM と呼ばれる VM を用意し、重要ファイルにアクセスするアプリケーションを Private VM 上に隔離して実行する。Private VM 上で実行されるアプリケーションによる重要ファイルに関する操作は Private VM で実行し、それ以外の操作は Commodity VM に転送する。しかし、どの操作を Private VM で実行するのかというポリシをプログラマが記述しなければならない。

Plan9 [5] もファイルサーバを用いているが、管理者権限を必要とするファイルにアクセスするためには、ファイルサーバの物理的なコンソールから操作する必要がある。これは管理者権限を必要とするファイルに対してアクセス制御と認証を分離していることになるが、管理者権限を必要とするたびにファイルサーバの前まで行って操作するのは不便である。SAccessor ではユーザの利便性を向上させるために、VM を用いてファイルサーバ機能を持つ VM とクライアントの VM を同一マシン上で動かし、認証はその場でシームレス行えるようにしている。

7 まとめと今後の課題

本稿では、仮想計算機を用いてファイルアクセスを二重化するシステム SAccessor を提案した。SAccessor では仮想計算機を用いて同一ホスト上に作業 VM と認証 VM を動作させ、作業 VM と認証 VM でファイルアクセス制御を二重化する。認証 VM では作業 VM とは独立して認証とファイルアクセス制御を行う。作業 VM の画面上に認証 VM が重ねて認証ダイアログを表示し、パスワードを入力させることで認証を行う。このシステムを実用的に利用可能にするために、認証ダイアログを判別できるようにし、認証の頻度を抑えることを可能にした。また、ファイルキャッシュを介した情報漏洩や setuid プログラムの実行の問題を解決した。

今後の課題は、setuid のときに認証 VM から入出力用のウィンドウを表示できるようにすることである。認証 VM のウィンドウで入出力を行うことで作業 VM に潜入している攻撃者に setuid プログラムからの標準出力を盗み見られることはなくなると考えている。

参考文献

[1] CERT Coordination Center

http://www.cert.org.
[2] CVE-2005-0750.
[3] CVE-2005-1263.
[4] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, and R. Neugebauer. Xen and the art of virtualization. In *Proceedings of the ACM Symposium on Operating Systems Principles*, October 2003.
[5] Bell Labs. Plan9:plan9 from bell labs. http://plan9.bell-labs.com/plan9/.
[6] Richard Ta-Min, Lionel Litty, and David Lie. Splitting interfaces: Making trust between applications and operating systems configurable. In *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2006)*, November 2006.
[7] Xin Zhao, Kevin Borders, and Atul Prakash. Towards protecting sensitive files in a compromised system. In *SISW '05: Proceedings of the Third IEEE International Security in Storage Workshop (SISW'05)*, pp. 21–28, Washington, DC, USA, 2005. IEEE Computer Society.