マイクロソフトは標準として効果的なセキュリティ対策を組み込むことができたかもしれませんが、良識が広まりました。  
–リック・メイベリー  
最適化は、機能するものを取り、ほとんど機能するが安価なもので置き換えることで構成されます。  
  
6.1はじめに  
一連のカードを並べて列を作り、ジョブを実行して、印刷物を持ち帰りました。  
その後、同時に複数のプログラムを実行するマシンが登場し、あるプログラムが別のプログラムに干渉するのを防ぐという保護問題が発生しました。  
そして、多くの信頼性の問題は、アプリケーションが互いに誤解し合ったり、互いに争ったりしていることに起因しています。  
電子メールクライアントとブラウザーが別々のマシンで実行されている場合、フィッシングがさらに困難になるため、電子メール内のURLをクリックするだけではうまくいきませんが、それは人生を難しくします。  
セキュリティエンジニアリングとコンピュータサイエンスが出会う場所です。  
。  
 システム内のどのリソースにアクセスできます–読み取ることができるファイル、実行できるプログラム、他のプリンシパルとデータを共有する方法など。  
Armのアーキテクチャのリファレンスマニュアルの7000以上のページ、またはWindowsのO / Sレベルでの同様に複雑な配置から始めると、最初の反応は「代わりに音楽を勉強したい！」私はあなたがそれをすべて理解するのを助けるように努めます。  
アプリケーションレベルでのアクセス制御は、非常に豊富なドメイン固有のセキュリティポリシーを表す場合があります。  
一部のトランザクションでは、監督者の承認が必要になる場合もあります。  
2。  
これらは多くの保護特性を強制します。  
3。  
Androidフォンは、さまざまな企業が作成したアプリをさまざまなユーザーとして扱い、データを相互に保護します。  
4。  
ハードウェアからオペレーティングシステム、ミドルウェア、アプリケーションレイヤーへと進むにつれて、コントロールは徐々に複雑になり、信頼性が低下します。  
たとえば、Androidによって提供されるさまざまな電話アプリの分離は、ブラウザでミラーリングされます。これにより、Webページの素材がドメイン名に基づいて分離されます（この分離は、多くの場合、あまり完全ではありません）。  
また、アプリケーション層またはミドルウェア層で構築されたアクセス制御は、基盤となるオペレーティングシステムまたはハードウェアのアクセス制御と大部分が重複する場合があります。  
まず、複数のプロセスの分離をサポートするオペレーティングシステムの保護メカニズムについて説明します。  
多くの場合、随意アクセス制御（DAC）と呼ばれます  
 通常ベンダーの管理下にあり、オペレーティングシステム自体をマルウェアによる変更から保護するメカニズム。  
最新のハードウェアは、仮想化と機能のためのCPUサポートを提供するだけでなく、マルウェアが永続化するのを防ぐためのトラステッドブート用のTPMチップなどのハードウェアサポートも提供します。  
  
オペレーティングシステムで提供されるアクセス制御は、通常、電話の場合はパスワードや指紋、サーバーの場合はパスワードやセキュリティプロトコルなどのメカニズムを使用してプリンシパルを認証し、ファイル、通信ポート、その他のシステムリソースへのアクセスを承認します。  
図6.1に示すように、読み取りの許可にはrを、書き込みの許可にはwを、プログラムの実行許可にはxを、アクセスをまったく許可しない場合は-を書き込みます。  
。  
また、データの読み書きも必要です。  
これで十分なことがよくありますが、簿記システムの特定のケースでは、必要なものではありません。  
また、サムにはこのアクセス権がないことも望まれます。  
アクセス許可は、図6.2のようになります。ユーザー操作アカウント会計監査システムプログラムデータ証跡Sam rwx rwx rr Alice rx x – –アカウントプログラムrx rx rw w Bob rx rrr図6.2 –簿記のアクセス制御マトリックス別の表現方法このタイプのポリシーには、（ユーザー、プログラム、ファイル）のアクセストリプルがあります。  
一般的なケースでは、同じリソースへのアクセスを共有するプロセスまたはスレッドのセットである保護ドメインほどプログラムに懸念はありません。  
 保護メカニズムを実装するだけでなく、それらをモデル化するためにも使用できます。  
この情報を保存および管理するより良い方法が必要です。2つの主なオプションは、ユーザーを圧縮することと、権利を圧縮することです。または行（機能、プロトコルエンジニアへの「チケット」とも呼ばれ、携帯電話の「権限」）  
  
大規模な組織を見ると、通常、ほとんどのスタッフが少数のカテゴリの1つに適合していることがわかります。  
ほんの数十人（セキュリティマネージャー、主な外国為替ディーラー、...）  
したがって、スタッフを割り当てることができる一連のグループ、つまり機能的な役割を設計する必要があります。  
 グループとロールという言葉はほとんど同じ意味で使用されますが、より慎重に定義すると、グループはプリンシパルのリストであり、ロールは1つ以上のプリンシパルが一定期間想定することができるアクセス許可の固定セットです。  
常に1人のウォッチキーパーがいて、ウォッチが変わったときに1人の警官が別の警官を救うという正式な手続きがあります。  
グループとロールは組み合わせることができます。  
銀行業務では、ケンブリッジ支店のマネージャーは、グループマネージャーのメンバーシップとケンブリッジ支店のマネージャーの役​​割を演じるという想定によって表される特権を持っている可能性があります。  
 一方、役割演技マネージャーには、マネージャー、副マネージャー、支社の会計士がすべて病気でいる間、副主任がいる場合があります。  
軍艦では、より年上のすべての人が殺されれば、普通の船員でさえ警戒するかもしれません。  
支社のマネージャーが病気の場合、マネージャーとして機能する会計士補は、大規模な転送で2番目の署名を提供するために地域の本社に依頼する必要がある場合があります。  
6.2.2アクセス制御リスト  
これは、アクセスコントロールリストまたはACL（「ackle」と発音）と呼ばれます。  
上記の最初の例では、ファイル3（アカウントファイル）のACL  
ユーザーアカウンティングデータSam rw Alice rw Bob r図6.3 –アクセス制御リスト（ACL）  
これらは、ユーザーが自分のファイルセキュリティを管理する環境では自然な選択であり、1970年代からUnixシステムで広く使用されるようになりました。  
Windowsのアクセス制御もACLに基づいていましたが、時間の経過とともに複雑になってきました。  
ACLは実装が簡単ですが、実行時のセキュリティチェックには効率的ではありません。通常のオペレーティングシステムは、起動後にアクセスを許可されているファイルではなく、特定のプログラムを実行しているユーザーを認識しているためです。  
最後に、アクセスルールをACLに配布すると、ユーザーがアクセスできるすべてのファイルを見つけるのが面倒になります。  
サーバー上のファイルにセキュリティポリシーに違反するACLがあるかどうかを確認するスクリプトを作成することはできますが、テクノロジの変更によって問題が発生する可能性があります。コンテナへの移行により、管理者がコンテナのACLをチェックするのも忘れたため、多くの企業データが漏洩しました。  
 また、雇用されたばかりの従業員のアクセスを取り消すには、通常、パスワードまたは認証トークンをキャンセルする必要があります。  
。  
6.2.3 Unixオペレーティングシステムのセキュリティ  
通常表示されるアクセス制御リストには、ファイルがディレクトリであるかどうかを示すフラグがあり、次に、所有者、グループ、およびワールドのそれぞれにr、w、xのフラグが付いています。所有者の名前とグループ名が含まれます。  
 読み取ることはできますが、書き込むことはできません。非グループメンバーはまったくアクセスできません。ファイルの所有者がアリスであること。グループはアカウントです。  
 スーパーバイザーとして実行され、マシン全体に無制限にアクセスできます。  
アクセスの決定は、プログラムに関連付けられたユーザーIDに基づいて行われます。  
の場合、アクセス制御の決定は「はい」です。  
さらに、特定の通信プロセスの開始など、rootだけが実行できる特定のことが存在します。  
つまり、システム管理者は何でもできるため、監査証跡を変更できないファイルとして実装することは困難です。  
ルートの侵害からログを保護する従来の、そして最も一般的な方法は、ログを分離しておくことです。  
後で説明するように、必須のアクセス制御もますます必要になる可能性があります。  
。  
 ファイル属性。  
したがって、上記の2番目の例で必要な機能を実現するために、ユーザー「account-package」を作成して、ファイル2（accountsパッケージ）を所有することができます。  
この特別なユーザーには、アカウントプログラムに必要なアクセス権を付与できます。  
 –そして、2次元のメカニズムを使用してそれを実装すると、結果はトリプルよりもはるかに直感的でなく、人々は間違いを犯しがちです。この方法は、いくつかの衝撃的なセキュリティホールにつながります。  
（suid rootプログラムを呼び出す人がその環境を制御し、これを予期しない方法で操作する可能性があるため、何をチェックするかを知ることは困難です。）  
トランザクションが実行される前に、マネージャーと会計士によってトランザクションが承認されることを望んでいるとします。アプリケーションレベルでこれを行うこともできます（たとえば、トランザクションのキューに2番目の署名を待機させるなど）  
ステートフルアクセスルールの管理は困難です。開いたファイルを追跡することは困難であり、物事が行き詰まる可能性があるため、それらは解雇されたばかりのユーザーの失効を複雑にする可能性があります。  
リソースが複数のユーザーによって使用され、OSレベルでアクセス制御を行いたい場合は、いくつかのオプションがあります。  
理論的には、ACLとsuidメカニズムは、多くの場合、目的の効果を達成するために使用できます。  
したがって、人々は自分のコードを設計して、厳密に本来持つべき特権よりもはるかに多くの特権を必要とするようにする必要があります。  
6.2.4機能  
これらは機能と呼ばれ、上の図6.1の例では、Bobの機能は図6.4のようになります。 ACLの反対。  
一方、どのユーザーがアクセス権を持っているのかを見つけるのは難しいため、ファイルのステータスを変更するのはより難しくなります。  
実際、システム全体の即時の失効は高すぎるため、スケーラブルなシステムは事実上、事実上機能を内部で使用します。 Unixでは、ファイル記述子は実際に機能であり、ACLのアクセス許可やファイルの所有者が変更された後でも、しばらくの間アクセスを許可し続けます。  
それでは、ACLを完全に廃止できますか？  
IBM AS / 400シリーズシステムは、1988年に機能ベースの保護を主流のコンピューティング市場にもたらし、商業的に成功しました。  
後で説明するように、Windows、FreeBSD、iOSの最新バージョンを含むオペレーティングシステムのACLを補完する機能が開始されました。  
たとえば、病院は「看護師は自分の病棟にいる患者、または過去90日間そこにいたすべての患者にアクセスできる」のようなアクセス規則を持っている場合があります。  
問題は、看護師に、多くの病棟または病院の部門に関連するファイルへのアクセスを許可する証明書付きのIDカードを与えることによって修正されました[535、536]。  
グレインを操作すると、スタックのすべてのレベルで利点が得られ、物事がより使いやすくなり、より安全なデフォルトがサポートされ、エラーが削減され、エンジニアリングの労力が削減され、コストも節約されます。  
6.2.5 DACおよびMAC  
これは、随意アクセス制御（DAC）の背後にあるモデルです。  
しかし、これは物事を大規模に管理することを困難にする可能性があり、1970年代に米軍は、機密情報を保護することを目的とした巨大なコンピューターセキュリティ研究プログラムを開始しました。一般ユーザーや監督者の行動に関係なく、「秘密」のクリアランスのみを持つユーザー。  
 システムでは、sysadminはもはやボスではありません。最終的な制御は、セキュリティポリシーを設定するリモートの政府機関にかかっています。  
。  
これは、強制アクセス制御のためのテクノロジーの開発を推進しました。これは、本のパート2で私が語る興味深い話です。  
彼らは、マルチレベルのセキュリティに似た何かが安全のために必要であることに気づき始めました。  
1990年代、コンピュータとネットワークがオーディオとビデオを処理するのに十分な速度になったので、クリエイティブ産業はデジタル著作権管理（DRM）に働きかけました。  
これは必須のアクセス制御の一種でもあります。サブスクライバー以外のユーザーと曲を共有しているサブスクライバーを停止することは、シークレットユーザーとインテリジェンスレポートを共有しているトップシークレットユーザーを停止することと同様です。  
触媒は、DRMをサポートするために暗号化をPCプラットフォームに導入するための、MicrosoftとIntelによる構想でした。  
MicrosoftはDRMから始めて、ドキュメントの著作権管理を提供することも、顧客をWindowsとOfficeにしっかりと固定する方法の1つかもしれないことに気づきました。これを行うには、オペレーティングシステムを改ざん防止する必要があります。これは、独立したプロセッサであるTrusted Platform Module（TPM）によって実現されます。  
TPMはブートプロセスを監視し、各ステージで、これまでにロードされたすべてのハッシュが、次のステージの復号化に必要なキーを取得するために必要です。  
TPMとトラステッドブートに基づくMACは、Windows 6（Vista）で使用されていました  
TPMの標準とアーキテクチャは他のオペレーティングシステムベンダーやデバイスOEMによって採用されており、Googleの製品をベースにしたオープンソースのTPMチップであるOpenTitanのプロジェクトさえあります。  
1Microsoftにはもっと野心的な計画があった。そのプロジェクトPalladiumは、レガシーソフトウェアの通常のアプリケーションと並んで、権利管理アプリに新しい、より信頼性の高い世界を提供するでしょう。  
2つの世界での実装はVistaでは複雑すぎることが判明し、2つの個別の開発作業の結果、放棄されました。しかし、ビジョンは2004年からArm’s TrustZoneで持続しました。  
6.2.6 AppleのmacOS  
 Machカーネル上で実行されるUnixのFreeBSDバージョンに基づいています。  
 高度な権限で実行しない限り。  
このUnixコアの上には、OpenGL、Quartz、Quicktime、Carbonを含む多数のグラフィックコンポーネントがあり、Aquaユーザーインターフェイスは、表面的にエレガントで一貫したビューをユーザーに提供します。  
デフォルトのインストールではrootアカウントが無効になっていますが、システムを管理するユーザーは、rootにsuできるグループ「ホイール」に属しています。  
。  
、それはマルウェアによる改ざんからコアシステムコンポーネントを保護するために使用される必須のアクセス制御メカニズムを組み込んだBSDの変形であるTrustedBSDに基づいています。  
6.2.7 iOS  
 iOSオペレーティングシステムを使用している-現在（2020年）  
iOSはUnixベースです。 AppleはCMUからMachカーネルを取得し、それをUnixのFreeBSDバージョンと融合し、パフォーマンスと堅牢性のために多くの変更を加えました。  
この場合も、MACコンポーネントがあり、ドメインとタイプの施行（DTE）からのメカニズムがあります。  
。  
アプリには、権限である機能もあります。モバイルネットワーク、電話、SMS、カメラなどのデバイスサービスにアクセスする機能、およびアプリがそのようなサービスを初めて使用するときにリクエストします。  
多くのデバイスサービスにより、サイドチャネル攻撃の可能性が開かれています。たとえば、キーボードへのアクセスを拒否されたアプリは、加速度計とジャイロを使用してキープレスを推定できます。  
iPhoneがアプリのみを実行するという意味で、Appleエコシステムは閉鎖されています。2ファーストユーストラストモデルは、Symbianによって普及したJava標準J2MEと1990年代に遡り、同じくらいから復活したアヒルのモデル時間。  
AppleとAndroidが登場したとき、最初は異なる選択をしました。  
アプリストアは、実行できるアプリを決定することにより、アクセス制御の一部を実行します。  
Appleが署名したこと3。  
iPhone 5Sは指紋認証と支払いを導入し、安全な飛び地（SE）を追加しました  
脆弱性はパッチが適用されるまで一時的にアクセスできるため、AppleはiOSもTrustZoneもこのような機密データで信頼しないことを決定しました。  
。  
キー管理は、システムオンチップ上のヒュージブルリンクに焼き付けられた一意の256ビットAESキーによってブートストラップされます。  
デバイスがロックされていても、iOSが着信メッセージの送信者を確認して通知できるように、一部のキーはまだ使用できます。この便利さの代償は、法医学機器がユーザーデータにある程度アクセスできることです。  
このような公開情報は、iOSセキュリティホワイトペーパー[128]に記載されています。  
これについては、第2部の電話に関する章で詳しく説明します。  
6.2.8 Android  
AndroidはLinuxベースです。異なるベンダーのアプリは、異なるユーザーIDで実行されます。  
iOSと同様に、アプリには権限があり、これは事実上機能です。SMS、カメラ、アドレス帳などのデバイスサービスへのアクセスを許可します。  
これは、オープンなエコシステムを維持しながら、更新の整合性をサポートします。  
Androidの初期のバージョンでは、ユーザーはインストール時にロットを承認するか、アプリを実行しなかった。  
そのため、Android 6は3に移行しました。いくつかの例外があります。企業は内部アプリの署名キーを取得できますが、乱用された場合、ブラックリストに登録される可能性があります。Appleモデルの最初の使用に対する信頼。以前のバージョン用にコンパイルされたアプリは、インストール時に機能を必要とします。  
SELinuxは、政府システムでMACをサポートするためにNSAによって開発されました。これについては、第9章でさらに説明します。  
iOS（そして実際にはWindows）と同様  
新しい電話のエコシステムは古いPCのエコシステムとは十分に異なりますが、パート2の電話に関する章で個別に説明するに値するほど、古い有線電話システムの特性を十分に継承しています。  
  
Windowsの現在のバージョン（Windows 10）  
。  
初期のバージョンのWindowsにはアクセス制御がありませんでした。  
、これはUnixに非常によく似ていて、それに触発されましたが、いくつかの拡張機能が付いています。  
これらの属性はユーザーだけでなくグループにも適用され、グループの権限により、UNIXのsuidプログラムとほぼ同じ効果を得ることができます。  
これらはこの順序で解析されます。関連するユーザーまたはグループのACLでAccessDeniedが検出された場合、競合するAccessAllowedフラグに関係なく、アクセスは許可されません。  
次に、ユーザーとリソースを個別の管理者がいるドメインに分割し、ドメイン間で信頼を一方向または両方に継承できます。  
部門のリソースドメインがユーザードメインを信頼するように配置できますが、その逆はできません。そのため、ハッキングされた、または不注意な部門管理者が外部に大きなダメージを与えることはできません。  
 ユーザーがローカル権限を必要とするタスク（ソフトウェアパッケージのインストールなど）を実行できるようにするため  
ハッキングされた管理者が行うことができる被害を制限するには、注意深い組織が依然として必要です。  
そのコアは、リモート認証を管理するActive Directoryでした– KerberosバリアントまたはTLSのいずれかを使用し、セキュリティサポートプロバイダーインターフェース（SSPI）の背後にカプセル化されました  
Active Directoryは基本的に、ドメイン内のユーザー、グループ、マシン、および組織単位を階層的な名前空間に編成するデータベースです。  
Windowsには、ACLを上書きまたは補完できる2つの方法で機能が追加されています。  
セキュリティポリシーは、システム全体ではなく、グループによって設定されます。グループポリシーは個々のプロファイルを上書きし、サイト、ドメイン、または組織単位に関連付けることができるため、複雑な問題への取り組みを開始できます。  
機能がWindowsに組み込まれる2つ目の方法は、多くのアプリケーションで認証にTLSを使用し、TLS証明書はActive Directoryの範囲外にある別の機能指向のアクセス制御層を提供することです。  
ルートとして実行されているすべてのソフトウェアの以前のデフォルトを回避するために、3つの保護メカニズムが追加されました。  
 代わりに、デフォルトの管理者権限をユーザーのデフォルトに置き換えました。  
]）  
Microsoftのエンジニアによると、これがWindowsの堅牢性の欠如の主な理由でした。アプリケーションは互換性のない方法でシステムリソースを使用しているためです。  
Vista以降、デスクトップは後のユーザープロセスの親プロセスとして機能するため、管理者も通常のユーザーとしてWebを閲覧し、ダウンロードしたマルウェアは、後の承認がない限りシステムファイルを上書きできません。  
（AppleのmacOSは似ていますが、内部の詳細は多少異なります。）  
基本的な考え方は、整合性の低いプロセス（インターネットからダウンロードしたコードなど）です。  
 信頼できるプロセスがない場合（問題のコードに対するMicrosoftによる署名の検証など）  
2012年に、Windows 8は動的アクセス制御を追加しました。これにより、ユーザーの仕事用PC、自宅のPC、電話などのコンテキストごとにユーザーアクセスを制御できます。これは、ユーザーに関するクレームとして表示されるActive Directoryのアカウント属性、またはドメインに関するクレームとしてKerberosチケットで行われます。  
 そのようなプリンシパルを表す。  
Windows 8.1は、Microsoftアカウント（以前のLiveID）を追加することで、クラウドコンピューティングへの移行にも備えています  
資格情報がローカルに保存されている場合、仮想化を使用して資格情報を保護します。  
 古いセキュアアテンションシーケンスctrl-alt-delの廃止（これはタッチスクリーンデバイスでは実行が困難であり、ユーザーはとにかく理解していませんでした）  
要約すると、Windowsは進化して、以前はマスマーケットで販売されていたどのシステムよりも豊富で柔軟性のあるアクセス制御ツールセットを提供します。（このようなコントロールが実際にどのように設計されているかは、銀行業務と簿記に関する章で取り上げます。）  
大きなWindowsショップのアクセス制御を設定することは、非常に熟練した仕事です。  
1980年代以降、規模と複雑さの増大により、アクセス制御が他のレベルで（またはそれとともに）他のレベルで行われるようになりました。  
たとえば、簿記システムは、オペレーティングシステムを1つの大きなファイルと見なすOracleなどのデータベース製品の上で実行されることがよくあります。  
そして1990年代以降、クライアント側での作業の多くはWebブラウザによって行われてきました。  
しかし、企業は現在、トランザクションをデータベースに直接渡すWebサーバーの前に、在庫、発送、eコマースを処理するための重要なデータベースがあります。  
Oracle、DB2、MySQLなどのデータベース製品には、オペレーティングシステムメカニズムに基づいてモデル化された独自のアクセス制御メカニズムがあり、通常、ユーザーとオブジェクトの両方で使用できる特権を備えています（そのため、メカニズムはアクセス制御リストと機能の混合です）  
ただし、一般的なデータベースアクセス制御アーキテクチャは、Windowsと同等の複雑さです。最新のデータベースは、それらがサポートするものと同様に、本質的に複雑です。通常、ファイルやドメインよりも高いレベルの抽象化を伴うビジネスプロセスです。  
第2部では、章全体を推論制御のトピックの調査に充てます。  
私が助言した企業では、オペレーティングシステムとデータベースのアクセス制御は、互いに話し合わない異なる部門によって管理されています。多くの場合、IT部門は、さまざまなアクセス制御システムが1つのように機能しているように見せるために粗雑なハックを行わなければなりませんが、深刻な穴を開けます。  
たとえば、Oracleには両方のオペレーティングシステムアカウントがあります（これらのユーザーはプラットフォームで外部から認証を受ける必要があります）。  
。  
多くのインストールでは、データベースは外部から直接アクセスできます。また、Webサービスのフロントエンドによってシールドされている場合でも、SQLコードをデータベースに挿入できる抜け穴が含まれていることがよくあります。  
2003年のSlammerワームは、Microsoft SQL Server 2000に対するスタックオーバーフローエクスプロイトを使用して感染を拡大し、侵入されたマシンがランダムなIPアドレスに大量の攻撃パケットを送信したため、大量のトラフィックが発生しました。  
あなたが1つをロックする必要がある場合、または何が起こっているのかさえ理解する必要がある場合は、[1174]などの専門の教科書を読むか、専門家に相談することをお勧めします。  
メインのアクセスコントロールルールは、同一生成元ポリシーです。これにより、Webページ上のJavaScriptまたはその他のアクティブコンテンツは、元のIPアドレスとのみ通信できます。このようなコードは、次のセクションで説明するように、ホストシステムを変更しないようにサンドボックスで実行されます。  
この本の以前の版では、Webセキュリティはサーバーの構成方法の問題であり、これがクロスサイトの脆弱性につながるかどうかを考慮していました。  
amount = 10000USD＆recipient = thiefこれは、mybank.comにログインしているユーザーがこれをクリックすると、有効なセッションCookieがあるため、トランザクションが実行されるリスクがある可能性があるという考えです。  
、およびReferer：ヘッダーを確認します。  
あなたが生計のためにウェブページをデザインするならば、あなたはこれらすべてのメカニズムをかなり詳細に理解したほうがよいでしょう（例[119]を参照）  
たとえば、私が2020年に書いているように、Amazon Alexaはクロスオリジンリソースシェアリングに関するポリシーの構成が間違っていることが判明しました。つまり、別のAmazonサブドメインを侵害した人は、ターゲットAlexaのスキルを悪意のあるものに置き換えることができます[1481] 。  
結局のところ、ブラウザーはラップトップ上の場所であり、信頼したくない人や、場合によっては悪意のある人によって書かれたコードを実行した場合です。前に説明したように、携帯電話のオペレーティングシステムは、より強力な保護を提供するために、異なるユーザーとして異なるアプリを実行します。  
（Chromeは、各タブを個別のオペレーティングシステムプロセスで実行することにより、これを確認しようとします。）  
多くのWebページには、複数の広告ネットワークとデータブローカーによって提供されるトラッカーやその他の悪いものがたくさんあり、それらは同じ生成元ポリシーの背後にある意図をあざけっています。賢明な人は、URLを直接入力するか、ブックマークを使用して、銀行のWebサイトにアクセスします。残念ながら、マーケティング業界はメールのリンクをクリックするように全員を訓練しています。  
モデルは、ユーザーがアプレットとしてダウンロードしたコードを実行したいが、アプレットがアドレス帳を盗んだり、マーケティング会社に郵送したり、CPUを独占したりするなど、何か厄介なことをする可能性があることを懸念しています。バッテリーを使い切る。  
、およびそれが発生したホストとの通信のみが許可されます（同一生成元ポリシー）  
これは、インタープリター、つまりJava仮想マシン（JVM）によってコードが実行されることによって実施されます。  
このアイデアは、Webページで使用される主要なスクリプト言語であるJavaScriptに適合しましたが、実際には別の言語です。その他のアクティブコンテンツも。  
6.2.12仮想化仮想化はクラウドコンピューティングを強化するものです。 1台のマシンで複数のマシンを個別にエミュレートできるため、仮想マシン（VM）をレンタルできます  
仮想化は1960年代にIBMによって発明されました[496]。単一のマシンは、VM / 370を使用して複数の仮想マシンに分割できます。  
仮想マシンモニター（VMM）を実行するだけでは不十分  
稼働中のVMMは、VMware ESX Serverを搭載したIntelプラットフォーム用に2003年に（特に）登場しました。  
Intelが2006年以降VT-xで提供しているプロセッササポートを使用すると、よりクリーンに作業を行うことができます。詳細については、以下で説明します。  
クライアントエンドでは、仮想化により、ホスト上でゲストオペレーティングシステムを実行できます（たとえば、macOS上のWindows）  
たとえば、従業員がラップトップでWindowsの2つのコピーを実行している場合があります。オフィス環境でロックダウンされたバージョンと自宅で使用するバージョンです。  
しかし、仮想化を使用してクライアントのセキュリティドメインを分離することは、見かけよりも困難です。  
安全なデータ共有は決して簡単なことではありません。  
これにより、企業は、サポートされていない古いソフトウェアを適切に安全に使用できます。  
 ブラウザはファイルをVMからホストのハードディスクにエクスポートし、「信頼できない」とマークします。そのため、ユーザーがファイルを開こうとすると、そのドキュメントとOfficeだけを保持する新しいVMが与えられます。  
アプリに統合されたネットワークサービスを使用すると、事態はさらに複雑になります。どのサイトがどのCookieにアクセスできるかについてのルールは複雑であり、シングルサインオンや、複数のドメインにまたがるHPが所有するワークフロー5Nowに対処することは困難です。  
ルールの多くは時々変更され、ハードで検証可能なアクセスロジックというよりもヒューリスティックです。  
実際には、仮想化のために構築されていない既存のOSやアプリに仮想化をレトロフィットします。  
それらはクラウドコンピューティングの仮想化に代わる軽量なものとして進化し、特にマーケティング担当者によってそれと混同されることがよくあります。  
コンテナの実装では、プロセス識別子、プロセス間通信、名前空間などのオペレーティングシステムメカニズムのサブセットを仮想化することで、プロセスのグループを分離します。また、サンドボックスやシステムコールフィルタリングなどの手法も使用します。  
他の新しいテクノロジーと同様に、経験よりも熱意のあるスタートアップがたくさんあります。  
クラウドシステムを使用する場合は、ツールの選択に真剣に注意を払い、さらにAmazon AWS Identity and Access Managementなどのサービスプロバイダーが提供するアクセス制御メカニズムのセットを学習する必要があります（わたし）  
これにより、別の複雑なレイヤーが追加されます。  
ただし、クラウドシステムを適切に拘束したとしても、分離メカニズムが達成できることにはハードウェアの制限があります。  
コンテナーを使用して支払い処理を展開する銀行は、少なくとも暗黙のうちに、AmazonまたはGoogleのサイズのクラウドでコンテナーをターゲットにすることが難しいことに依存しています。  
  
ほとんどのアクセス制御システムは、ユーザーが実行できることを制御するだけでなく、プログラムが実行できることも制限することを目指しています。  
あるプロセスが別のプロセスを妨害しないようにすることは、保護の問題です。  
それぞれにいくつかのフレーバーがあります。  
これは、商用オペレーティングシステムがやろうとしていることです。サンドボックステクニック（一般的なプログラミング環境には制限が多すぎる）を使用しない限り  
プロセス間通信を可能にするために共有されるメモリ領域がある場合があります。ただし、プログラムは偶発的または故意の変更から保護する必要があり、同様に保護されているメモリにアクセスできる必要があります。  
古典的なメカニズムは、セグメントアドレス指定です。  
セグメントレジスタはオペレーティングシステムによって制御され、多くの場合、アクセス制御メカニズムをハードウェアとリンクする参照モニターと呼ばれるオペレーティングシステムのコンポーネントによって制御されます。  
初期のIBMメインフレームには2つの状態のCPUがありました。マシンは許可された状態かそうでないかのどちらかでした。  
許可されたプログラムは、許可されたライブラリーからロードされたプログラムです。  
 認可されたライブラリから。  
1960年代にMITで開発され、Unixに影響を与えたオペレーティングシステムであるMulticsは、さまざまなレベルの特権を表す保護リングを導入しました。リング0プログラムはディスクに完全にアクセスでき、スーパーバイザステートはリング2で実行され、ユーザーコードはさまざまな低い特権で実行されましたレベル[1684]。  
ハードウェアとソフトウェアのセキュリティメカニズムのインターフェースには、多くの一般的な問題があります。  
デバイスドライバー）  
これを行うメカニズムは慎重に設計する必要があります。そうしないと、セキュリティバグが予想されます。  
  
初期のPCで使用されていたIntel 8088/8086プロセッサは、システムモードとユーザーモードを区別していなかったため、実行中のプログラムはマシン全体を制御していました6。  
80386には組み込みの仮想メモリと十分な大きさのメモリセグメント（4 Gb）がありました  
486およびPentiumシリーズのチップにより、パフォーマンスが向上しました（キャッシュ、順不同実行、MMXなどの追加の命令）  
保護リングは多くのメカニズムによってサポートされています。  
。  
2006年から、Intelは、Intel VTと呼ばれるx86仮想化のハードウェアサポートを追加しました。これにより、クラウドコンピューティングの採用が促進されました。  
ただし、ネイティブのIntel命令セットには機密性の高いユーザーモード命令が含まれており、アプリケーションコードの書き換えやホストされているオペレーティングシステムへのパッチなどの面倒な回避策が必要です。  
機密性の高いオペコードをトラップする必要はありますが、システムコールはVMMの介入を自動的に必要とせず、変更されていないオペレーティングシステムを実行でき、処理が高速になり、システムは通常より堅牢になります。  
 BIOS用。  
2015年、インテルはSoftware Guard eXtensions（SGX）をリリースしました  
同社は、Trusted Computingイニシアチブの初期にこのようなアーキテクチャに取り組んできましたが、TrustZoneと競合するためにエンクレーブアーキテクチャが必要になるまで、次のセクションで説明します。  
、SGXでは新しい命令とメモリアクセスチェックも導入され、非エンクレーブプロセスがエンクレーブメモリ（ルートプロセスでさえも）にアクセスできないように  
SGXは、DRMとセキュリティで保護されたクラウドVM、特に暗号化キー、資格情報、または機密の個人情報を含むも​​のに対して推奨されています。これはSpectreや類似の攻撃の脅威にさらされています。サイドチャネルの章で詳しく説明します。  
時間と空間の両方で。  
会社は署名を委任しました（悪意のある人々がコードに署名を取得できるようにするため）  
したがって、人々はSGXマルウェアを実験しています。SGXマルウェアは、ウイルス対策ソフトウェアでは検出されないままになる可能性があります。  
 ホストアプリのエンクレーブからの攻撃。彼らは、問題は飛び地が何をすることになっているのかについての明確さの欠如であり、あらゆる合理的な脅威モデルは信頼できない飛び地を含まなければならない、と主張している[1688]。  
 SGX’7内でこの脅威に対処するのに役立つはずです。  
これは、コールドブートおよびDMA攻撃に対処し、信頼できないハイパーバイザーからコードを保護することを目的としました。また、次世代の飛び地のスペースとパフォーマンスの制限を引き上げる可能性もあります。  
[2010]。  
アクセス制御の脆弱性だけでなく、暗号の問題もあります。これについては、高度な暗号工学の章で説明します。  
6.3.2 Armプロセッサ  
オリジナルのアーム（Acorn Risc Machineを意味していました）  
1991年、Armは独立した企業となり、Intelとは異なり、ファブを所有または運営していません。チップ設計者が製品に組み込んだ一連のプロセッサコアのライセンスを取得しています。  
高速割り込みと通常の割り込み、リセット時に入力されるシステムモード、およびさまざまな種類の例外処理を処理する複数の監視モードがあります。、およびその他のメモリ管理ユニット（MMU）  
Armは2011年に、64ビット処理をサポートし、複数の32ビットオペレーティングシステムを仮想化できるバージョン8を発売しました。  
コアは、ダースステージが12段を超えるパイプラインを備えた大きな64ビットスーパースカラープロセッサから、安価な組み込みデバイス用の小さなコアまで、すべてのサイズで提供されます。  
上記の; 2004年に携帯電話メーカーが利用できるようになった[44]。  
TrustZoneは、通常のオペレーティングシステムと汎用アプリケーションのオープンワールドに加えて、暗号化や重要なI / O（携帯電話では、SIMカードと指紋リーダー）  
プロセッサがセキュアな状態か非セキュアな状態かは、ユーザーモードまたはスーパーバイザモードのどちらかと直交しています（ただし、セキュアモードとハイパーバイザモードのどちらかを選択する必要があります）。  
閉じた世界は単一の信頼できる実行環境（TEE）をホストします  
TrustZoneは2004年にリリースされましたが、2015年まで閉鎖されました。 OEMは自分の利益を保護するためにそれを使用し、時々NDAの下を除いて、アプリ開発者にそれを公開しませんでした。  
繰り返しになりますが、社内関係者は「待って見る」とつぶやきます。間違いありません。  
現在、Chromeなどのブラウザはタブを異なるプロセスに配置しているため、スクリプトの実行速度が遅い場合でも、あるWebページが他のタブの速度を落とすことはありません。  
CHERIを使用すると、サブスレッドを生成するプロセスが特定の範囲のメモリへの読み取りおよび書き込みアクセスを割り当てることができるため、複数のサンドボックスを同じプロセスで実行できます。  
このテクノロジーの長期的な約束は、それがWindows、Android、iOSなどのオペレーティングシステムで完全に使用された場合、近年のほとんどのゼロデイ攻撃を防ぐことができるということです。  
  
Android、Linux、Windowsなどの一般的なオペレーティングシステムは非常に大規模で複雑であり、その機能は非常に多様な状況下で何十億ものユーザーによって毎日テストされています。  
発見後、バグはCERTまたはベンダーに報告されます。パッチが出荷されます。パッチはリバースエンジニアリングされ、エクスプロイトが生成される可能性があります。そして、パッチを時間内に適用しなかった人々8完全な開示：これは、ケンブリッジの同僚のチームによって開発され、ロバートワトソンが率いました。  
少数のケースでは、脆弱性は報告されるのではなく一度に悪用されます-攻撃が脆弱性の既知の存在のゼロ日から発生するため、ゼロデイエクスプロイトと呼ばれます。  
攻撃者の従来の目標は、システムの通常のアカウントを取得してシステム管理者になることで、システムを完全に乗っ取ることができました。  
21世紀には、ユーザーとルートの区別は2つの理由でそれほど重要ではなくなりました。  
 そのため、それらは最も一般的な攻撃対象でした。また、管理者として多くのアプリケーションを実行していたため、侵害される可能性のあるすべてのアプリケーションが管理者アクセスを許可しました。  
これも、管理者アクセスを必要としません。  
ただし、ボットネットの遊牧民は、その名前が示すようにルートとして実行されるルートキットをインストールすることを好みます。それらはリモートアクセストロイの木馬またはRATとしても知られています。  
別の違いは、エクスプロイトがワーマブルであるかどうか、つまり、人間の介入なしにマルウェアを1つのマシンから別のマシンにオンラインですばやく拡散するために使用できるかどうかです。  
第2章でWannacryとNotPetyaについて説明しました。これらは、NSAによって開発された脆弱性を使用し、その後、他の国家主体に漏洩しました。  
執筆時点で最も厄介なワーム可能なエクスプロイトは、既知のルートパスワードを使用するIoTデバイスを乗っ取るために使用されるワームであるMiraiの亜種です。  
ワーム可能なエクスプロイトはルートアクセスをよく使用しますが、そうする必要はありません。エクスプロイトが自動で先に送信できることで十分です9。  
9まれに、人間が感染してもマルウェアが急速に広がる可能性があります。たとえば、ILoveYouワームは2000年に件名が電子メールで広がり、それを開くのに十分な数の人が開き、スクリプトを実行して全員に送信しました新しい被害者のアドレス帳。  
6.4.1スタックを破壊する  
メモリの安全性の侵害を伴う攻撃は、1990年代後半から2000年代初頭にかけてオペレーティングシステムに対するエクスプロイトの半分以上を占めました[487]。Morrisワームで使用されている典型的な例は、Unix fingerコマンドの脆弱​​性でした。  
攻撃者がより長い引数でコマンドを使用すると、引数の末尾のバイトがスタックを上書きし、システムによって実行されることになります。  
 コマンド、またはコントロールフローを変更しなかった他のレジスタコマンド、およびそのいずれかを実行した場合にプロセッサをキャッチするタスク。  
。  
1960年代初頭のタイムシェアリングシステムのほとんどがこの脆弱性の影響を受け、修正されました[804]。  
Intelの80286プロセッサは、1982年に明示的なパラメータチェック命令（読み取りの検証、書き込みの検証、および長さの検証）を導入しましたが、アーキテクチャの依存関係を防ぐために、ほとんどのソフトウェア設計者によって回避されました。  
  
多くの脆弱性は同じ一般的なテーマのバリエーションであり、文法Aのデータが文法Bのコードであると解釈されるときに発生します。  
URL）  
これらはタイプセーフティの障害です。  
解放後のタイプの安全障害の使用は、リモート実行の脆弱性の最も一般的な原因であり、近年ブラウザに多くの攻撃を提供しています。  
悪意のあるチャンクが割り当てられると、ヒープ上でその場所を占め、古い無害な機能が新しい機能と呼ばれるときに、代わりに悪意のある機能が呼び出される可能性があります。  
分類法については、Gary McGrawの著書「Software Security [1266]」を参照してください。  
多くの場合、ゲームはエラーメッセージによって表示されます。エラーメッセージから、有能でやる気のあるユーザーが攻撃を仕掛けるのに十分な推測をすることができます。  
通常の対策は、すべてのユーザー入力を疑わしいものとして扱い、それを検証することです。  
可能な場合は、そのような攻撃を設計することにより、安全な状況でのみユーザー入力に対処する必要があります。特定のエクスプロイトをブラックリストに登録する必要がある場合、メカニズムを適切に維持する必要があります。  
これらは、トランザクションが2つ以上の段階で実行され、最初の段階でアクセス権が検証され、2番目の段階で機密情報が実行されるときに発生します。  
UNIXの初期のバージョンでは、古典的な例が発生しました。ここでは、「mkdir」というディレクトリを作成するコマンドが、ストレージの割り当てと所有権のユーザーへの移行という2つのステップで機能していました。  
その後、元のプロセスが再開し、パスワードファイルの所有権がユーザーに変更されます。  
これらのラッパーは、カーネルのアドレス空間で実行され、すべてのシステムコールの開始状態と終了状態を検査し、セキュリティロジックのみをカプセル化します。  
システムコールはお互いに関してアトミックではありません。 2つのシステムコールが共有メモリにアクセスするために互いに競合する可能性が多く、これにより、チェックタイムから使用時間（TOCTTOU）が発生します。  
初期（2007）  
それ以来他にもあり、時間の経過とともに各CPUチップで出荷されるプロセッサが増え、コンテナがアプリケーションを展開するためのより一般的な方法になるにつれて、この種の攻撃はますます問題になる可能性があります。  
異なるタイプのタイミング攻撃は、バックアップおよびリカバリシステムから発生する可能性があります。  
 取り消されたアクセス権を再取得して、さらに巧妙なトリックをプレイできます。  
 [1708]。  
 違う！攻撃者はガジェットを探すことができます。ガジェットとは、有益な効果をもたらし、最後に戻る一連の命令です。  
次に、コールスタックの制御をつかむだけです。  
最新の攻撃バリアント、ブロック指向プログラミング（BOP）  
この攻撃と防御の共進化は間違いなく続くでしょう。  
攻撃テクノロジの最新の主要な革新は、CPUパイプラインの動作を対象としています。  
基本的な考え方は、現在のプロセスが実行を許可されていない命令であっても、大規模な最新のCPUのパイプラインは非常に長く複雑であるため、先を見越して次の数十の命令を予測するというものです（アクセスチェックが2つの命令であると想定してください）。将来的には、それが禁止する読み取り操作は、その後の2つの命令です）  
取られなかったパスは、情報をキャッシュにロードし、遅延の形で情報を漏らす可能性があります。  
スペクターとメルトダウンについては、この本の第2部のサイドチャネルに関する章で詳しく説明します。  
この章の冒頭にあるロジャーニーダムの言葉を思い出します。このような攻撃は、コンテナーとVMだけでなく、TrustZoneやSGXなどのエンクレーブメカニズムによっても提供できる保護を制限します。  
  
要塞を攻撃する一般的な方法は、警備員をだましてあなたを助けさせることです。オペレーティングシステムも例外ではありません。  
プレイヤーは、プレーヤーがシステム管理者であるかどうかをチェックするゲームを作成し、そうであれば既知のパスワードで別の管理者アカウントを作成します。  
 正規のユーティリティを呼び出す前。  
彼らがディレクトリに入り、そこに何があるかを確認するためにlsと入力すると、ダメージが与えられます。  
この特定のケースでの修正は簡単でした。管理者の「PATH」変数（コマンドが呼び出されたときに適切な名前のプログラムが検索されるディレクトリのリスト）  
。  
ただし、これは、アクセス制御を堅牢にするために多くの小さな詳細を正しく取得する方法の例であり、これらの詳細は常に事前に明確であるとは限りません。  
ソフトウェアのインストールを管理者に制限することで、銀行や政府機関などのMicrosoftの大企業の顧客がシステムをロックダウンして、スタッフがゲームやその他の不正なソフトウェアを実行できないようにすることができました。  
そのため、何億人もの人々が、それを必要としないはずの管理者特権を持っていて、Webサイトが何かをするように指示するボックスをポップアップしただけで、悪意のあるコードをインストールしました。  
Windows Vistaはこれから離れ始めましたが、マルウェアのエコシステムがPCの世界で確立され、企業がWebサイトを使用する代わりにアプリをインストールするように人々に圧力をかけ、アプリがアクセスを要求するようになり、Androidエコシステムに根付き始めています彼らが本当に持ってはならないあらゆる種類のデータとサービスに。  
  
ソフトウェアのセキュリティは、すべての運命と悲観ではありません。 2000年代には状況は大幅に改善しました。  
いくつかの違いがありました。  
1つ目は、特定の防御で構成されています。  
スタックが上書きされると、高い確率でカナリアが変更されます[487]。  
 すべてのメモリをデータまたはコードとしてマークし、前者が実行されるのを防ぎます。 2003年にWindows XPで登場しました。  
 同時に到着しました。システムの各インスタンスでメモリレイアウトを異なるものにすることにより、攻撃者がターゲットアドレスを予測することを困難にします。  
制御フローの整合性メカニズムには、コンパイル時に可能な制御フローグラフを分析し、間接的な制御フロー転送を検証することで実行時にこれを強制することが含まれます。これは2005年に登場し、その後の10年間でさまざまな製品に組み込まれました[348]。  
2。  
Coverityなどの静的分析プログラムは、多数の潜在的なソフトウェアバグを見つけ、コードがベストプラクティスから逸脱している方法を強調することができます。プロジェクトの最初から使用すると、大きな違いが生まれます。  
 根本的な解決策は、より優れた言語を使用することです。私の同僚は、CまたはC ++ 10ではなく、Rustでシステムコードを書いています。  
3つ目はより良いトレーニングです。  
（このために彼らが作成した 『Writing Secure Code』 [927]は、まだ一読する価値があります。）  
4。  
アジャイル開発方法論が拡張され、パッチの非常に迅速な展開とインシデントへの対応が可能になります。これにより、設計、コーディング、およびテストに費やした労力を最も緊急の問題に向けることができます。  
プログラムは2010年にMozillaの研究から10Rustが出現しただけで、Firefoxの再開発に使用されています。 Stack Overflowの2016年〜2019年の年次調査で、お気に入りの言語に投票されました。  
また、ソフトウェアは、デフォルトの構成、一般的には何かを行う最も簡単な方法が安全になるように設計する必要があります。  
ただし、多くのシステムには危険なデフォルトと厄介なコードが同梱されており、あらゆる種類のインターフェースがSQLインジェクションのような攻撃にさらされてはなりません。  
  
環境の変化によってセキュリティモデルが損なわれると、多くのセキュリティ障害が発生します。  
アクセス制御メカニズムも例外ではありません。  
。  
この環境では、セキュリティメカニズムの機能はほとんど間違いを含むことです。あるユーザーの入力エラーやプログラムのクラッシュによって別のユーザーのファイルが削除または上書きされるのを防ぐため。しかし、Unixのセキュリティは古典的な「成功した災難」になりました。  
バークレーのバージョンは、単一のマシンから、すべて1つのLAN上にあり、すべて1つの管理下にあるマシンのネットワークへの拡張を想定しています。  
 もともとは、安全なネットワーク上のメインフレーム用に作成されました。  
したがって、Berkeleyモデルが本当に必要とするリモート認証は、単にサポートされていませんでした。  
世界を結びつけるために、Kerberos、TLS、SSHなどのプロトコルをダクトテープとして改造する必要がありました。  
、セキュリティエンジニアが追いつくためにこれまでより速く実行している。  
初期の仮定の一部はまだ部分的に適用されていますが、グローバルに適用されるものはありません。  
競争する企業があります。お互いを軽んじる政治集団、そして互いに戦争状態にある国民国家。  
以前は単にバグのあるコードでしたが、今では悪意のあるコードがたくさん出回っています。  
  
アクセス制御メカニズムは、ハードウェアからオペレーティングシステム、ブラウザなどのミドルウェア、アプリケーションまで、システムのさまざまなレベルで動作します。  
アクセス制御の主な機能は、特定のグループ、ユーザー、およびプログラムがエラーまたは悪意を介して行うことができる損害を制限することです。  
基本的なメカニズム（およびその問題）  
ほとんどの攻撃は、日和見的なバグの悪用を伴います。複雑な製品、広く使用されている製品、またはその両方の製品は、特に脆弱性が発見されて悪用される可能性があります。  
研究の問題アクセス制御の問題のほとんどは、1960年代または1970年代初頭に特定され、Multics [1684]やCAP [2020]などの実験システムで解決されました。  
最近の研究スレッドには、飛び地と、きめの細かいアクセス制御を追加するためのCHERIメカニズムが含まれています。  
 第2版​​では、「今後数年間の有用な研究トピックは、堅牢であるだけでなく、使用可能でもあるプログラマーとエンドユーザーの両方がアクセス制御メカニズムを設計する方法になるだろう」と予測しました。YaseminAcar氏などによる最近の研究は、それをピックアップして、セキュリティ研究の分野で最も急速に成長している分野の1つに発展させました[11]。  
暗号化の章では、暗号APIがECBモードで長いメッセージを暗号化するなど、本当に安全でないデフォルトを使用するように誘導する方法についてすでに触れました。 WindowsシステムやIntelまたはArm CPUのアクセス制御メカニズムに冷たくなった人が見つかるように、アクセス制御も同様に悪いです。  
技術レベル（ハードウェア、OS、アプリなど）でアクセス制御について知っていることを組織レベルに拡張できますか？  
興味深い技術的類似物ができた今、深いアウトソーシングと仮想組織の世界のためにこれを再訪する時が来ましたか？  
クラウドネイティブセキュリティファウンデーションは、クラウドネイティブソフトウェアの導入と管理のためのコンテナーやその他のテクノロジーを中心に、より優れたオープンソースプラクティスに向けて人々を動かそうとしています。  
カールランドヴェールは、1960年代から1980年代のオペレーティングシステムに見られる多くのフローについて有用なリファレンスを提供しています[1129]。  
 1970年のウィリスウェア[1986]による。バトラーランプソンの拘束問題に関する独創的な論文は1970年代に登場し[1125]、3年後、ジェリーソルツァーとマイクシュローダーによって影響力のある別の初期の論文が書かれました[1639]。  
IntelのSGXおよび実際のCPUセキュリティアーキテクチャに関する標準リファレンスは、Victor CostanおよびSrini Devadasによるものです[479]。  
 ある年から次の年へ。  
それ以来、ROP攻撃、スペクター、その他多くの攻撃がありました。短くても便利なアップデートは、Matthias Payerのソフトウェアセキュリティ[1504]です。  
現在の攻撃に関する詳細は、おそらくGoogleのProject Zeroブログにあります。たとえば、必須のアクセス制御コンポーネントを使用した最新のオペレーティングシステムのハッキングに関係するものについての洞察については、実際に見つかったiPhoneに対する攻撃の分析を参照してください[204]。