セキュリティシステムはバックファイアという厄介な習慣があり、それらが大きな問題を引き起こすことは間違いありません。  
–ロジャー・ニーダム6。  
 あなたは一組のカードで列を作り、ジョブを実行し、プリントアウトして去りました。  
 次に、同時に複数のプログラムを実行するcamemachineと、1つのプログラムが別のプログラムに干渉するのを防ぐ保護問題に沿って。  
 そして、多くの信頼性の問題は、アプリケーションがお互いに誤解し合ったり、お互いに争ったりしていることに起因しています。  
メールクライアントとブラウザ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、メール内のURLだけをクリックすることはできませんでしたが、こ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るか（ユーザーが読み取り可能なファイル、実行可能なプログラム、他のプリンシパルとデータを共有する方法など）。  
 最初に、Armのアーキテクチャリファレンスマニュアルの7000以上のページ、または2056と同じように複雑な配置を確認してください。  
 はじめにO / Sレベルのウィンドウの場合、最初の反応は「代わりに音楽を勉強したいです！」かもしれません。この章では、すべてを理解できるように支援します。  
 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、スーパーバイザー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スマートフォンは、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、お互いのデータを保護します。  
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間の分離は、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、それほど完全ではありません）。  
 それは非常に厄介になる可能性があり、それを理解するために、基礎となる原則、共通アーキテクチャ、およびそれらがどのように進化したかを理解する必要があります。  
 これらは歴史的に最初に登場し、1960年代に最初のタイムシェアリングシステムとともに発明されました。それらは、多くの上位層メカニズムが構築される基盤を維持し、同様に上位層で同様のメカニズムを刺激します。  
2。  
 ソフトウェア攻撃とそれらを防御するための手法（MAC、ASLR、サンドボックス、仮想化、およびハードウェアで何ができるか）を紹介します。  
 これらは、DOSやWin95 / 98などの古いシングルユーザーPCオペレーティングシステムの有毒な遺産を積み重ねるのに役立ち、プロセスがデータを変更し、管理者権限で実行していると思わせない限り、実行できない多くのアプリケーションを制限します。 。  
2オペレーティングシステムのアクセス制御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許可します。  
 図6に示すように、読み取りの許可にはr、書き込みの許可にはw、プログラムの実行許可にはxを、アクセスしないには-を書き込みます。  
OperatingAccountsAccountingAuditSystemProgramDataTrailSamrwxrwxrwrAlicexxrw–Bobrxrrr図6。  
マネージャーであるアリスは、オペレーティングシステムとアプリケーションを実行する必要がありますが、承認されたインターフェースを介してのみ実行する必要があります。彼女はそれらを改ざんする能力を持っていてはなりません。  
 監査人であるボブはすべてを読むことができます。  
 取引が適切な形式であること、つまり各借方が他のどこかでクレジットとバランスが取れていることを確認したいので、Aliceにアカウントファイルへの書き込みアクセスを禁止しないでください。  
 したがって、アカウンティングデータファイルへの書き込みアクセスは、アカウンティングプログラムを介してのみ可能であることが望ましいです。  
2：セキュリティエンジニアリング207ロスアンダーソン6。  
 オペレーティングシステムのアクセス制御UserOperatingAccountsAccountingAuditSystemProgramDataTrailSamrwxrwxrrAlicerxx––Accounts programrxrxrwwBobrxrrr図6。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のようなプログラムには関心がありません。  
 しかし、それらは十分に拡張できません。50,000のステーションと300のアプリケーションを持つ銀行は、15,000,000のエントリのマトリックスを持ち、パフォーマンスのオーバーヘッドを課すだけでなく、管理者のミスに対して脆弱になる可能性があります。  
 最初に、グループまたはロールを使用して多数のユーザーのセットを同時に管理できます。2つ目では、列（アクセス制御リスト）または行（機能、プロトコルエンジニアへの「チケット」とも呼ばれます）ごとにアクセス制御マトリックスを保存できます。携帯電話の「許可」）[1639、2020]。  
2。  
 銀行には、40または50のテラー、コールセンターオペレーター、ローンオフィスなどがあります。  
。  
）個人用にカスタマイズされたアクセス権が必要になります。  
 一部のベンダー（Microsoftなど）は、グループと役割をほとんど同じ意味で使用していますが、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、役割は、1人以上のプリンシパルが一定の期間想定す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るだけで、時計が変わったときに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代理の役割を引き受けることで表される特権を持つことができます。  
セキュリティエンジニアリング208ロスアンダーソン6。  
 オペレーティングシステムのアクセス制御この区別に注意する必要があるかどうかは、アプリケーションの問題です。  
 銀行では、「1千万ドルを超える送金は、少なくともマネージャーがランクで、もう1人がアシスタントの会計士であるランクの2つの従業員によって承認されなければならない」というポリシーがあるかもしれません。  
6。  
2アクセス制御リストアクセス権の管理を簡素化する従来の方法は、アクセス制御マトリックスを一度に1列ずつ、その列が参照するリソースとともに格納することです。  
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は、図6に示すようになります。  
UserAccountingDataSamrwAlicerwBobr図6。  
 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注意が必要です。通常のオペレーティングシステムは、起動後にどのファイルがアクセスを許可されているかではなく、どのユーザーが特定のプログラムを実行しているか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もできますが、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ACLstooを確認するのを忘れたため、多くの企業データが漏洩しました。  
）そして、従業員のセキュリティエンジニアリング209ロスアンダーソンのアクセス権を取り消す6。  
 操作されたばかりのシステムアクセス制御は、通常、パスワードまたは認証トークンをキャンセルすることによって実行する必要があります。  
6。  
3Unixオペレーティングシステムのセキュリティ従来のUnixシステムでは、ファイルに任意のアクセス制御リストを設定することはできませんが、ファイルの読み取り、書き込み、実行を可能にするrwx属性のみを使用できます。  
 すべてのフラグが設定されたディレクトリには、ACL：drwxrwxrwx Alice Accountsがあります。図6の最初の例では、  
マシンの起動時に制御を取得するプログラム（オペレーティングシステムカーネル）はスーパーバイザとして実行され、マシン全体への無制限のアクセス権を持っています。  
 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。  
 したがって、rootは好きなことを実行できます。つまり、任意のファイルにアクセスしたり、任意のユーザーになったりすることができます。  
 rootユーザーIDは、通常、随意アクセス制御を備えたシステムのシステム管理者が利用できます。  
 私たちの例では、サムはアカウントをいじくりまわして、いじくり回しであると誤って非難された場合、自分自身を守ることが困難になる可能性があります。さらに、管理者になるように管理されたハッカーは、侵入のすべての証拠を削除できます。昔は、ロックされた部屋のプリンターにシステムログを送信する必要がありました。最近では、他のマシンやサードパーティのサービスに送信することを意味します。  
次に、ACLにはユーザーの名前のみが含まれ、プログラムの名前は含まれません。したがって、（ユーザー、プログラム、ファイル）のアクセストリプルを実装する簡単な方法はありません。  
プログラムの所有者は、そのプログラムを表すファイルにsuidのマークを付けることができます。これにより、プログラムを呼び出したユーザーの特権ではなく、その所有者の特権で実行できるようになります。  
2。  
 この特別なユーザーには、accountsプログラムに必要なアクセス権を付与できます。  
 プログラマーはしばしば怠惰であるか、厳しい締め切りに直面しています。したがって、アプリケーションをsuid rootにするだけで、何でも実行できます。  
 アクセス制御の決定を行う責任は、オペレーティングシステム環境からアプリケーションプログラムに移され、ほとんどのプログラマーは、必要なすべてのものをチェックするために不慣れな経験を持っています。  
第三に、ACLは変更可能な状態の表現があまり得意ではありません。  
 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のが難しく、stu canがスタックする可能性があるため、これらのユーザー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。  
 リソースが複数で使用され、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のメカニズムを使用して、目的の効果を達成できます。  
 そのため、人々は自分のコードを設計して、厳密に持っているべきよりもはるかに多くの特権を必要とします。  
2。  
 これらは機能と呼ばれ、図6の例では、  
4ここ：UserOperatingAccountsAccountingAuditSystemProgramDataTrailBobrxrrr図6。  
 ランタイムセキュリティチェックの方が効率的で、難しいことなく権利を委任することができます。ボブは、「ここに私の能力があります。午前9時から午後1時までファイル4を読む権利をDavidに委任します。セキュリティエンジニアリング211ロスアンダーソン6。  
 オペレーティングシステムのアクセス制御署名済みボブ。  
 インシデントを調査したり証拠を準備したりする必要がある場合、これは面倒な場合があります。  
 分散Unixでは、Kerberosチケットの存続期間中、アクセスが持続する場合があります。  
 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、病院には「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」などのアクセス規則がある場合があります。  
 多くの病棟または病院の診療科に関連するファイルにアクセスする資格を与える証明書を看護師のIDカードに与えることにより、問題が修正されました[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もたらされ、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手間が減り、コストも節約されます。  
2。  
 これは、随意アクセス制御（DAC）の背後にあるモデルです。コンピューターをスーパーバイザーモードで起動してから、管理者として、信頼度の低いタスク（権限のない会社が作成したアプリを実行するなど）に特権の少ないアカウントを使用できるようにすることができます。完全に信頼する、または他のユーザーにリモートログオンアクセスを許可する。  
 このようなマルチレベルセキュア（MLS）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任されています。  
2。  
 スーパーバイザ、または必要に応じてrootアクセスはリモートコントロール下にあります。  
1980年代から、安全エンジニアは安全度水準のアイデアにも取り組みました。大まかに言えば、信頼性の高いシステムは、信頼性の低いシステムに依存してはなりません。  
 軍事システムの人々はまた、保護メカニズム自体の耐タンパー性が重要であることに気づきました。  
 これは必須のアクセス制御の一種でもあります。サブスクライバーとサブスクライバー以外で曲を共有するサブスクライバーを停止することは、多くの場合、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するトップシークレットユーザーを停止することと同じです。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によるイニシアチブでした。  
MicrosoftはDRMから始めて、ドキュメントの権利管理を提供することも、顧客をWindowsとOceにしっかりと固定する方法の1つになる可能性があることを認識しました。  
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した。これは、独立したプロセッサであるTrusted Platform Module（TPM）、つまり、信頼できる起動とハードディスク暗号化をサポートするPCマザーボードに搭載されたスマートカードチップによって実現されています。  
 システムの実際のスーパーバイザーは、マシンの所有者ではなく、オペレーティングシステムのベンダーです。  
 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によって採用されました。現在、Googleの製品に基づくオープンソースTPMチップ、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。そのプロジェクトパラディウムは、レガシーソフトウェアの通常のアプリケーションに加えて、権利管理アプリに新たな、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には複雑すぎることが判明し、2つの個別の開発の結果、放棄されましたが、ビジョンは2004年からArmのTrustZoneで持続しました。  
2。  
2。  
 BSDlayerはメモリ保護を提供します。高度な権限で実行しない限り、アプリケーションはシステムメモリ（または他のメモリ）にアクセスできません。  
 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スコンポーネントがあり、Aquaユーザーインターフェイスは、表面上で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、システムを管理できるユーザーはrootにsuできるグループ「ホイール」に属しています。  
 バージョン10以降。  
6。  
7iOS2008年以来、AppleはiPhoneでスマートフォン革命をリードしており、これは（iPadのような他のデバイスと共に）iOSオペレーティングシステムを使用しています。これは現在（2020年に）2番目に人気があります。  
たとえば、標準的なUnixでは、ファイル名に複数のパス名を含めることができます。これにより、オペレーティングシステムで認識されるファイルオブジェクトを表すiノードが作成されます。 iOSではこれが単純化されているため、ファイルには一意のパス名があり、これがファイルレベルのアクセス制御の対象になります。  
Appleがこれを導入したのは、アプリがiPhoneをブリックするのではないかと心配して、保証の主張につながったためです。  
これは、ユーザーが同意した場合に許可されます2。  
 サイドチャネルについては、その主題に関する章のパート2で説明します。  
 J2MEは、trust-on-installなどもサポートしています。  
 いずれの場合も、アプリストアを持つことは重要なイノベーションでした。ノキアは、これが両面市場を成功させるために重要であることを理解できませんでした。  
 これはAppleの場合はハードパワーであり、Androidの場合はソフトパワーです。これについては、電話の章で説明します。  
2。  
 これにより、同社は収益の一部を抽出し、マルウェアやその他の望ましくない動作（サイドチャネルの悪用によるアクセス制御の無効化など）についてアプリをスクリーニングすることもできます。  
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はこのような機密データを使用してiOSもTrustZoneも信頼しないことを決定しました。  
iOSはシステムパーティションにアクセスできますが、ユーザーの個人データは暗号化され、キーはSEによって管理されます。  
 デバイスの電源がオンになると、ユーザーはパスコードの入力を10回試行します。その後、マスターキーからファイルキーが導出され、使用可能になります4。  
 SEはアップグレードも管理し、ロールバックを防止します。  
モバイルデバイスのセキュリティはかなり複雑な問題であり、アクセス制御と改ざん防止だけでなく、SIMカードの提供からアプリストアの運用まで、ユーザーがデバイスを使用する文化、企業がデバイスを操作する方法、どのように政府機関がそれらをスパイするか。  
6。  
8AndroidAndroidは、世界で最も広く使用されているオペレーティングシステムで、2を搭載しています。  
 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、これは効果的な機能です。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。apkファイル、およびiOSアプリはAppleによって署名されていますが、Androidの検証キーは自己署名証明書で提供され、開発者の名前として機能します。  
 各パッケージには、一連の権限を要求するマニフェストが含まれており、ユーザーは「危険な」ものを承認する必要があります。これは、お金を費やしたり、個人データを危険にさらしたりできるものです。  
 しかし経験から、ほとんどのユーザーは何でもクリックするだけでインストールプロセスを完了でき、フラッシュライトアプリでさえアドレス帳へのアクセスを要求することがわかりました。  
4改ざん防止の章でヒュージブルリンクについて説明し、監視とプライバシーの章でiPhone PINの再試行を無効にします。  
2。  
Android 5以降、SELinuxは必須のアクセス制御でオペレーティングシステムを強化するために使用されており、コアシステム機能を攻撃から保護するだけでなく、プロセスを強力に分離して違反をログに記録しています。  
 哲学は、ユーザー、開発者、プラットフォームの3つの当事者の同意を必要とするアクションです。  
 新しい電話エコシステムは、古いPCエコシステムとは十分に異なりますが、パートIIの電話に関する章で個別の議論に値するほど、古い有線電話システムの特性を十分に継承しています。  
6。  
9Windows現在のバージョンのWindows（Windows 10）は3番目に人気のあるオペレーティングシステムであり、2020年3月（2016年まではWindowsがリーダーでした）に毎月10億のアクティブデバイスを達成しています。  
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。  
 まず、読み取り、書き込み、実行だけでなく、所有権の取得、アクセス許可の変更、削除など、より柔軟な委任をサポートするための個別の属性がありました。  
 属性は、Unixのように単純にオンまたはオフになっているのではなく、複数の値があります。AccessDenied、AccessAllowed、またはSystemAuditを設定できます。  
 より豊富な構文を使用すると、プリンターのインストールなどの日常の構成タスクで完全な管理者特権を必要としないように、問題を調整できます。  
 典型的な大企業では、すべてのユーザーをHRが管理する担当者ドメインに配置し、サーバーやプリンターなどの資産を部門の管理下にあるリソースドメインにすることができます。個々のワークステーションは、ユーザーが管理することもできます。  
 個々のワークステーションは、次に部門を信頼し（逆は不可）、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します。  
 これらすべてを管理するために使用されるデータ構造と、hideSecurity Engineering216Ross Anderson6。  
 オペレーティングシステムのアクセス制御ユーザーインターフェイスからのACLの詳細は、レジストリと呼ばれます。  
 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をオーバーライドまたは補完できる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。  
 以前は、非常に多くの日常的なタスクが管理者特権を必要としていたため、多くの企業がすべてのユーザーを管理者にしていたため、マルウェアを含めることが困難でした。そして、多くの開発者は、すべてにアクセスできるという前提でソフトウェアを作成しました（恥のホールについては、[？]を参照してください）。  
そこで、昇格された特権を必要とするアプリケーションを起動し、仮想化を使用してそれらを含めるアプリケーション情報サービスを追加しました。たとえば、レジ​​ストリを変更した場合、「実際の」レジストリは変更せず、表示できるバージョンのみを変更します。  
 タスクに管理者特権が必要な場合、ユーザーは管理者パスワードを要求する昇​​格プロンプトを受け取ります。）管理ユーザーは悪意のあるソフトウェアをインストールするようにだまされることが多いため、Vistaはファイルの整合性レベルの形式で必須のアクセス制御を追加しました。  
2012年、Windows 8は、セキュリティエンジニアリング217ロスアンダーソン6を制御できる動的アクセス制御を追加しました。  
 オペレーティングシステムのアクセス制御ユーザーの仕事用PCと自宅のPCおよび電話などのコンテキストによるユーザーアクセス。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。  
1は、セキュリティコンテキストまたはそのようなプリンシパルが属するグループで実行されているユーザー、コンピューター、プロセス、またはスレッドであるプリンシパルと、そのようなプリンシパルを表すセキュリティ識別子（SID）を使用した、より明確なプリンシパルを追加しました。  
 ウィンドウズ8。  
 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。  
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。  
2。  
 1980年代以降、規模と複雑さの増大により、オペレーティングシステムレベルではなく（またはオペレーティングシステムレベルで）アクセス制御が他のレベルで行われるようになりました。  
 したがって、ほとんどのアクセス制御はデータベースで実行する必要があります。すべてのオペレーティングシステムが、ログオンする各ユーザーに認証済みIDを提供している可能性があります。  
6。  
10。  
 しかし、企業は現在、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を持っており、トランザクションをデータベースに直接渡すWebサーバーが前にあります。  
セキュリティエンジニアリング218ロスアンダーソン6。  
 オペレーティングシステムのアクセス制御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のメカニズムをモデルとする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が与えられます（メカニズム領域にはアクセス制御リストと機能が混在しています）。 。  
ユーザーが多すぎる顧客について学習しすぎないようにすることを目的としたアクセス制御があるかもしれません。これらはステートフルである傾向があり、単純な「はい/いいえ」アクセスルールではなく、可能な統計的推論に対処する場合があります。  
多くの場合、管理のしやすさがボトルネックになります。  
一部の製品では、開発者はオペレーティングシステムの制御をバイパスできます。  
 後者を使用すると、他の部門が行っていることと同期する手間を省くことができて便利です。  
したがって、データベースのセキュリティ障害は直接問題を引き起こす可能性があります。  
Windowsは非常に複雑であるため、安全に構成するのが難しいのと同じように、通常のデータベースシステムでも同じことが言えます。  
6。  
10。  
 メインのアクセスコントロールルールは、同じ生成元のポリシーであり、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。次のセクションで説明するように、このようなコードは、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ました。  
com / transfer。  
2。  
com、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ります。  
 Web認証メカニズムに関する問題もあります。セクション4でOAuthについて簡単に説明しました。  
4。  
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を誤って設定していることが判明しました。これは、別のAmazonサブドメインを侵害した誰もが、ターゲットAlexaのスキルを悪意のあるものに置き換える可能性があることを意味します[1481]。  
 結局のところ、ブラウザーは、信頼したくない人や、時には悪意のある人によって書かれたコードを実行しているラップトップ上の場所です。先に説明したように、携帯電話のオペレーティングシステムは異なるユーザーとして異なるアプリを実行し、より堅牢な保護を提供します。  
 （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。  
多くのウェブページはトラッカーやその他の悪いことでいっぱいであり、複数の広告ネットワークやデータブローカーによって提供されているため、同一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。  
2。  
 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の実行を望んでいるが、アプレットが自分の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を独占して実行を停止したりするなど、厄介なことをする可能性があることを懸念しているバッテリー。  
これは、限られたアクセス権しか持たないインタプリタ（Java VirtualMachine（JVM））によってコードが実行されることによって強制されます[783]。  
 Javaのバージョンも使用されていますセキュリティエンジニアリング220ロスアンダーソン6。  
 オペレーティングシステムのアクセス制御スマートカード上にあるため、異なる企業が作成したアプレットをサポートできます。  
2。  
 仮想化は1960年代にIBMによって発明されました[496]。単一のマシンをVM / 370を使用して複数の仮想マシンに分割できます。  
 ホストオペレーティングシステム上で仮想マシンモニター（VMM）を実行してから、他のオペレーティングシステムを上で実行するだけでは不十分です。絶対アドレスやプロセッサクロックなどのプロセッサ状態を明らかにする機密情報を扱う必要があります。  
 Intelが2006年以来VT-xで提供していた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。  
クライアント側では、仮想化により、ホスト（たとえば、macOS上のWindows）の上でゲストオペレーティングシステムを実行できます。これにより、柔軟性だけでなく、より優れた封じ込めの可能性がもたらされます。  
 Samsungo↵ersKnoxは、ユーザーが同じデバイスで通常のAndroidを楽しんでいる間、雇用主がリモートでロックダウンして管理できる仮想マシンを携帯電話上に作成します。  
 人々は複数のVM間でデータを共有する必要があり、USBスティックやWebメールアカウントなどのアドホックメカニズムを使用している場合、これは分離を弱体化させます。  
 たとえば、企業のPC上の特定のアプリに合わせて調整されたBromium5o↵ersVMがあるため、1つのVM forOce、1つはAcrobat Reader、1つはブラウザーなどになります。  
 では、どのようにOceドキュメントをダウンロードしますか？さて、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Oce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関するルールは複雑であり、シングルサインオンとHPSecurity Engineering221Ross Anderson6が所有するワークフロー5Nowに対処するのは困難です。  
 複数のドメインにまたがるハードウェア保護。  
 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。  
つまり、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の代替手段として進化し、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、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、経験よりも熱心なスタートアップがたくさんいます。  
 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、Amazon AWS Identity andAccess Management（IAM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別のセットも学習する必要があります。 ）。  
 たとえば、2019年に銀行と警察に生体認証サービスを提供しているセキュリティ会社は、データベース全体を無防備のままにしました.2人の研究者がElasticsearchを使用してそれを発見し、データベース上で数百万人の写真、指紋、パスワード、セキュリティクリアランスレベルを発見しました読むだけでなく書くこともできた[1864]。  
 2018年に、2つのクラスの強力なサイドチャネル攻撃が公開されました：MeltdownとSpectreです。これらは、次のセクションで説明し、サイドチャネルの章で詳しく説明します。仮想化とコンテナの進化に関する包括的な調査については、Randal [1575]を参照してください。  
3ハードウェア保護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的としています。  
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。  
3。  
 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。  
サンドボックス化手法（一般的なプログラミング環境には制限が厳しすぎる）を使用しない限り、単一のプロセッサで保護の問題を解決することは、少なくとも、あるプログラムが別のプログラムのコードやデータを上書きするのを防ぐメカニズムを備えていることを意味します。  
これは通常、ハードウェアアクセス制御をプロセッサのメモリ管理機能と統合する必要があることを意味します。  
 メモリは、メモリのセグメントを指すセグメントレジスタと、そのセグメント内の場所を指すアドレスレジスタの2つのレジスタでアドレス指定されます。  
プロセッサ自体が動作するため、実装はより複雑になっています。  
 後者の場合、プログラムはオペレーティングシステムによって割り当てられたメモリセグメントに制限されていました。前者では、自由にセグメントレジスタに書き込むことができます。  
適切な承認済みライブラリがあれば、必要なアクセス制御ポリシーをこの上に実装できますが、これは必ずしも効率的ではありません。また、システムのセキュリティは、悪意のあるコードかバグのあるコードかを問わず、不正なコードを認証済みライブラリから除外することに依存していました。  
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のプログラムはディスクに完全にアクセスでき、監視プログラムの状態はリング2で実行され、ユーザーコードはさまざまなレベルで実行されました特権レベル[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と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。  
g。  
 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。  
セキュリティエンジニアリング223ロスアンダーソン6。  
 ハードウェア保護6。  
1Intelプロセッサ初期の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、無視できるほど十分に大きなメモリセグメント（4 Gb）があり、マシンは32ビットのフラットアドレスマシンとして扱われました。  
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。  
 プロシージャは下位レベルのリング内のオブジェクトに直接アクセスすることはできませんが、異なる特権レベルでコードを実行し、複数のスタックセグメントなどのサポートするインフラストラクチャを管理できるゲートがあります。  
S / 370やPowerPCなどの一部のプロセッサアーキテクチャは仮想化が容易であり、その理論的要件は1974年にGeraldPopekとRobert Goldbergによって確立されました[1532]。それらには、未加工のプロセッサ状態を公開するすべての機密命令が特権命令であることが含まれます。  
 ハードウェアにVMMサポートを追加するということは、オペレーティングシステムを設計どおりにリング0で実行できることを意味します。 VMMには、その下に独自のメモリアーキテクチャのコピーがあります。  
 最近のIntel CPUには9つのリングがあります。通常のコードのリング0–3は、ハイパーバイザーのリング0–3 VMMルートモードの追加セットで、下部はBIOSのシステム管理モード（SMM）です。  
2015年、インテルはSoftware Guard eXtensions（SGX）をリリースしました。これにより、trustedcodeをエンクレーブ（メモリの暗号化されたセクション）で実行し、残りのコードは通常どおり実行できます。  
 暗号化はメモリー暗号化エンジン（MEE）によって実行されますが、SGXは新しい命令とメモリーアクセスチェックも導入して、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します。  
 SGXのセキュリティ境界はCPUであるため、そのソフトウェアはメインメモリで暗号化されます。これは実際のペナルティを課します。6これらはクラッシュプログラムで開発され、RISCプロセッサの出現とiAPX432の市場の失敗に続く市場シェアを節約しました。  
3。  
 もう1つの欠点は、SGXコードがIntelによって署名されなければならないことでした。Sopeople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です。  
 この単純な点は、エンクレーブアーキテクチャの再考を強いる可能性があります。 Intelは、「将来的には、Intelの制御フロー強制テクノロジ（CET）がSGX内のこの脅威への対処に役立つはずです」と述べています7。  
 これは、コールドブートおよびDMA攻撃に対処し、信頼されてい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、次世代のエンクレーブのスペースとパフォーマンスの制限も解除される可能性があります。  
 [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。  
3。  
 元のArm（Acorn Risc Machineを意味する）は最初の商用RISC設計でした。それはMIPSの直前の1985年にリリースされました。  
 初期のコアには32ビットのデータパスがあり、15個のレジスタが含まれていました。そのうちの7個は、割り込み時にコンテキストを切り替えるコストを削減するために、システムプロセス用のバンクされたレジスタによってシャドウイングされていました。  
 コアには当初メモリ管理が含まれていなかったため、Armベースの設計ではハードウェア保護を大幅にカスタマイズできます。現在、メモリ保護ユニット（MPU）を備えたバリアントと、仮想メモリを処理するメモリ管理ユニット（MMU）を備えたバリアントがあります。  
 ハイパーバイザのサポートにより、さらに別のスーパーバイザモードが追加されました。  
TrustZoneは、「2つの世界」モデルmen-7をサポートするセキュリティ拡張機能です。2019年のROP攻撃に対する最善の防御策は、レジスターに保持されているキーでポインターに署名するためのiPhone X3以降のAppleのメカニズムのようです。攻撃者がシグネチャを推測できないため、これによりROP攻撃が停止します。  
4。  
 オペレーターは補助金付きの電話をロックしたいと規制当局はRF機能を制御するベースバンドソフトウェアを改ざん防止することを望んでいたので、飛び地の「キラーアプリ」をPhoneswereしました[1239]。  
 プロセッサがセキュアな状態か非セキュアな状態かは、それがユーザーモードかスーパーバイザモード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）。  
TrustZoneは2004年にリリースされましたが、2015年まで閉鎖されていました。OEMはそれを使用して自分の利益を保護し、NDAの下での場合を除いて、アプリ開発者に開放しませんでした。  
 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。  
 現在のところ、Chromeなどのブラウザではタブが異なるプロセスに配置されているため、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に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徹底的に使用された場合、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。  
6。  
 多くのバグが見つかり、そのうちのいくつかは典型的なライフサイクルを持つ脆弱性を引き起こします。  
セキュリティエンジニアリング226ロスアンダーソン6。  
 何が問題なのか彼らのマシンが危険にさらされていることに気付くかもしれません。  
脆弱性ライフサイクルの経済学と生態学は、安全保障経済学者による集中的な研究の主題です。これについてはパートIIIで説明します。  
 最初のステップには、パスワードの推測、つまりソーシャルエンジニアリングを行い、次にオペレーティングシステムのバグを使用してユーザーからrootにエスカレートすることが含まれる場合があります[1129]。  
 まず、Windows PCが最も一般的なオンラインデバイス（Androidが追い抜かれる2017年まで）だったため、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、危険にさら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、その名前の候補として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、あるマシンから別のマシンにマルウェアを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。  
 WannacryとNotPetyaについては、第2章で触れましたが、これらはNSAによって開発された脆弱性を使用し、他の政治家に漏洩しました。  
 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使用することがよくありますが、そうする必要はありません。エクスプロイトが自動で前方に送信できることで十分です9。  
9まれなケースですが、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。例としては、ILoveYouワームが2000年にその件名の電子メールを介して自分自身を広げ、それを開くために十分な数の人々に開かせ、スクリプトを実行してそれを全員に送信させました。被害者のアドレス帳。  
4。  
4。  
 メモリの安全性違反を伴う攻撃は、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が、その割合はその後ゆっくりと低下しています。  
 Morrisワームで使用されている古典的な例は、Unix fingerコマンドに脆弱性がありました。  
 攻撃者がより長い引数でコマンドを使用すると、引数の後続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てしまいました。  
 ランディングパッドは、管理者権限を持つシェルを直接作成するような処理を行う攻撃コードにプロセッサを提供しました（図6を参照）。  
図6。  
 1960年代初頭のタイムシェアリングシステムのほとんどは、この脆弱性の影響を受け、修正されました[804]。  
 Intelの80286プロセッサーは、1982年に明示的なパラメーターチェック命令（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。  
セキュリティエンジニアリング228ロスアンダーソン6。  
 何が間違っているか6。  
2その他の技術的な攻撃多くの脆弱性は、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。  
g。  
 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、ヒープに置き換わり、古い無害な機能が新しい機能と呼ばれると、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、安全なコンテキスト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発生した古典的な例では、「mkdir」というディレクトリを作成するコマンドは、2つのステップで機能していました。ストレージが割り当てられ、所有権がユーザーに転送されました。  
 その後、元のプロセスが再開され、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。  
4。  
 システムコールは互いに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、チェックタイム対使用時間（TOCTTOU）攻撃が発生します。  
 それ以来、他のものがあり、時間が経過するにつれて各CPUチップに出荷されるプロセッサの数が増え、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っているため、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。  
最近多くの研究で注目されている攻撃の1つは、リターン指向プログラミング（ROP）[1708]です。  
十分なガジェットを収集することで、チューリングパワフルなマシンを組み立て、攻撃コードをROPガジェットのチェーンとして実装することができます。  
 これは、よく知られているガジェットを提供するために共通の共有ライブラリlibcを使用するlibcへの攻撃から発展しました。それ以来、SGXエンクレーブ内のマルウェアがホストアプリにステルス攻撃を仕掛けることを可能にする攻撃など、多くの亜種が開発されてきました[1688]。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、現代の大規模なCPUのパイプラインは非常に長く複雑であるため、現在のプロセスが実行を許可されない命令であっても、先読みして次の数十の命令を予測するというものです（アクセスチェックは将来2つの命令であり、禁止される読み取り操作は、その後の2つの命令です）。  
 狡猾な場合、あるプロセスは別のプロセスのメモリを読み取るように調整できます。  
 緩和策は発表されていますが、同じ一般的な種類のさらなる攻撃が発見され続けており、完全に制御されるまでに数年と新世代のプロセッサが必要になる場合があります。  
4。  
最適化は、機能するものを、ほとんど機能するが安価なものに置き換えることで構成されます。また、最近のCPUは非常に最適化されているため、Spectreテーマのバリエーションが増えることは間違いありません。  
 特に、慎重な会社が高価値の暗号化キーをエンクレーブに委託することを阻止し、旧式のハードウェア暗号化の耐用年数を延ばす可能性があります。  
4。  
 初期の攻撃の1つは、管理者に実行を依頼されたプログラムであるトロイの木馬でしたが、意外な驚きが含まれています。  
 バリアントは、Unixディレクトリ内のすべてのファイルを一覧表示するlsコマンドなど、一般的なシステムユーティリティと同じ名前のプログラムを記述し、正規のユーティリティを呼び出す前に管理者権限（ある場合）を悪用するように設計することでした。  
 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タイプするとき、損傷は行われます。  
 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）には、 ‘を含めないでください。  
 ModernUnixバージョンには、これがデフォルトで付属しています。  
おそらく、これまでに攻撃されたシステムの数に関して、ユーザーインターフェイス障害の最も深刻な例は、2つの事実で構成されています。1つ目は、Windowsが確認ダイアログを常にポップアップし、人々がボックスをクリックして作業を完了するように訓練することです。次に、2006年まで、ユーザーは何でもインストールするために管理者になる必要がありました。  
 しかし、ほとんどの環境では、一般の人々は仕事をするためにソフトウェアをインストールする必要があります。  
 これは、コードが怠惰であったり、本来あるべきではないデータを収集したりするためにrootとして実行されると主張した多くのアプリケーション開発者によって悪化しました。  
 セキュリティエンジニアリング231ロスアンダーソン6。  
 何が間違っているのかについては、電話機の章の後半で説明します。  
4。  
 世紀の変わり目に、脆弱性の90％がブフローバーフローでした。この本の第2版が2008年に発売されるまでに、それは半分未満でしたが、現在はさらに少なくなっています。  
1。  
 スタックカナリアは、スタック上の戻りアドレスの横にコンパイラによって挿入されるランダムな数字です。  
 データ実行防止（DEP）は、すべてのメモリをその他の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。  
 DEPをバイパスするROP攻撃を実行するツールキットがあるため、これは特に重要です。  
 しかし、分析は正確ではなく、ギャップを利用するために進化したトリックには、ブロック指向プログラミング攻撃があります[964]。  
 2つ目は、より優れた汎用ツールで構成されています。  
 （後で追加した場合、対処するのが面倒な数千のアラートがスローされる可能性があります。  
3。  
 2002年に、Microsoftはセキュリティ対策を発表しました。これには、すべてのプログラマーが安全なコードを書く方法について訓練を受けていることが含まれます。  
）他の会社も同様でした。  
 最新のアプローチはDevSecOpsで、パート3で説明します。  
アーキテクチャは重要です。制御された方法で進化するクリーンなインターフェースを持つことは、製品のセキュリティに長期的な利害関係を持つ経験豊富な誰かの鋭い眼の下で、大きな違いを生む可能性があります。  
セキュリティエンジニアリング232ロスアンダーソン6。  
 必要なだけ多くの特権を与える：最小特権の原則[1639]。安全なデフォルトを達成し、最小限の特権を使用するには、健全なアーキテクチャが重要です。  
 これらには、個人的および企業的なインセンティブの失敗、ならびに不十分な教育およびセキュリティツールの不十分なユーザビリティが含まれます。  
4。  
 初期環境で適切に機能したメカニズムは、より広いメカニズムで失敗することがよくあります。  
 たとえば、Unixは元々「シングルユーザーMultics」（したがって名前）として設計されました。  
 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。あるユーザーのスタイピング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。  
メインフレームは自律的であり、ネットワークはセキュリティプロトコルの外にあり、承認の転送はありませんでした。  
 NFSなどのSunの拡張機能がパーティーに追加されました。複数のトラステッドLANを持つ単一の企業を想定しています。  
 時にはwifiで、時にはモバイルネットワークで通信し、何百万人もの作者からのアプリを実行する数十億台の電話（そのほとんどは自己完結型で、一部は積極的に悪意のあるもの）の到来により、セキュリティエンジニアは追いつくためにこれまで以上に速く動いています。  
 初期の仮定の一部はまだ部分的に適用されますが、グローバルに適用されるものはありません。  
 競合する企業があります。お互いを軽んじる政治グループ、そして互いに戦争状態にある国家。  
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あります。  
5。  
6。  
 高レベルのメカニズムはより表現力を高めることができますが、本質的な複雑さから実装者のスキルに至るまで、さまざまな理由で攻撃に対して脆弱になる傾向があります。  
 最も広く使用されている例は、クライアント側のAndroidとWindows、およびサーバー側のLinuxです。それらは共通の系統と多くのアーキテクチャ上の類似点を持っています。  
 ほとんどの攻撃は、日和見的なバグの悪用を伴います。複雑で、広く使用されている製品、またはその両方は、脆弱性が発見されて悪用される可能性が特に高いです。  
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。  
’最近のYasemin Acar氏らの研究はそれを取り上げ、それを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、ユーザーに安全でないデフォルトをどのように誘発することが多いかについてすでに触れました。 Windowsシステムまたは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けるのと同じように、accesscontrolも同じくらい悪いです。  
 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？ 20世紀には、Bell-LaPadulaからClark-Wilsonまで、多数のセキュリティポリシーが提案されました。それらについては、greatSecurity Engineering234Ross Anderson6で説明します。  
 第2部の長さ  
 Cloud Native Secu-rity Foundationは、クラウドネイティブソフトウェアをデプロイおよび管理するためのコンテナやその他のテクノロジーを中心とした、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]。  
 アクセス制御の問題について生徒に読む教科書は、Dieter Gollmannの「ComputerSecurity」[779]です。  
ソフトウェアセキュリティの分野は急速に変化しています。攻撃は1年から次の年に大きく変化します（少なくとも詳細は）。  
 それ以来、ROP攻撃、Spectreandなどが発生しています。短くても便利なアップデートは、Matthias PayerのSoftware Secu-rity [1504]です。  
org。  
セキュリティエンジニアリング235ロスアンダーソン