分散システムとは、存在さえ知らなかったコンピューターの障害により、自分のコンピューターが使用できなくなる可能性があるシステムです。  
  
 私たちが他の名前でバラと呼ぶものは、甘い香りがします–ウィリアムシェイクスピア  
  
あらゆるサイズの堅牢な分散システムを構築するには、認証、アクセス制御、暗号化以外にも多くのものが必要です。  
オンラインサービスプロバイダーを使用してアドレス帳を更新すると、更新が別のデバイスで1秒後に表示される場合や、おそらく数時間後に表示される場合があることに、誰もが気付いているはずです。  
私たちは確かな理論と、多くの苦労して得た経験を持っています。  
同時実行障害として発生するプロトコルへの攻撃についてはすでに説明しました。  
最後に、ネーミングは厄介な問題になる可能性があります。  
多くの組織は、IDカードを使用して市民を追跡する場合でも、デバイスIDを使用してオブジェクトを追跡する場合でも、より大きくよりフラットな名前空間を構築しようとしていますが、実際に実行できることには制限があります。  
  
プロセスは、同時に実行できる場合は同時実行と呼ばれ、これはパフォーマンスにとって不可欠です。最近のコンピューターには多くのコアがあり、多くのユーザーのために一度に多くのプログラムを実行します。  
プロセスは古いデータを使用する場合があります。一貫性のない更新を行う可能性があります。更新の順序は重要な場合とそうでない場合があります。システムがデッドロックする可能性があります。異なるシステムのデータが一貫した値に収束することはありません。そして、物事を正しい順序で行うことが重要である場合、または正確な時間を知ることさえ重要である場合、これはあなたが考えるよりもトリッキーになる可能性があります。  
さまざまな理由により、システムの並行性はますます高まっています。  
2つ目は、デバイスの複雑さです。高級車には現在、数十から数百の異なるプロセッサを搭載できます。  
各CPUの奥深くでは、命令が並列に実行され、この複雑さが、アクセス制御の章で説明したSpectre攻撃につながります。  
次に、相互作用の複雑さがあります。アプリケーションレイヤーに行くと、レンタカーの予約などの日常のトランザクションが他のシステムを呼び出して、クレジットカード、信用照会機関のスコア、保険金請求履歴などをチェックしますが、これらのシステムは順番に他人に依存することがあります。  
コンピュータサイエンティストはアムダールの法則を教えられています。並列化できる比率がpであり、sが追加リソースによるスピードアップである場合、全体的なスピードアップは（1ffip + p / s）です。  
したがって、プログラムの4分の3は並列化できるが、残りの4分の1は並列化できない場合、得られる最大の速度向上は4倍です。そして、8つのコアを投げると、実用的なスピードアップは3倍にはなりません1。  
アクセス制御と同様に、ユーザーが偶然または故意に干渉することを防ぐ必要があります。  
以下では、いくつかの具体的な例を示します。それらは決して完全なものではありません。  
7.2.1古いデータの使用と支払いによる状態の伝播  
例として、Unixの「mkdir」の脆弱性について言及しました。この場合、2つのフェーズで実行される特権命令は、それが作用するオブジェクトの名前を変更することにより、途中で攻撃される可能性があります。  
ffi1 =（0.25 + 0.09375）  
ffi1 = 2.909 1960年代。最初のマルチユーザーオペレーティングシステムの1つであるIBMのOS / 360では、ファイルを開こうとすると、ファイルが読み取られ、権限がチェックされました。ユーザーがアクセスを許可されている場合は、再度読み取られました。  
これらは、チェック時間から使用時間（TOCTTOU）の例です。  
ファイルシステムでこのような攻撃を見つける体系的な方法がありますが[251]、攻撃は依然として仮想化環境でのシステムコールなどの低レベルとビジネスロジックなどの高レベルの両方で発生します。  
良いケーススタディはカード詐欺です。  
、そしてカードネットワークが国際化したため、問題は1980年代に着実に悪化しました。  
 また、他の場所では許容できないコストと遅延を課すことになります。  
マーチャント端末は、特定の制限（フロア制限）までのトランザクションを処理できます  
実際には、使用の直前にチェックされるトランザクションは、ローカルまたは大規模なトランザクションのみです。  
世界のATMトランザクションの約半分は、誰かがATMで盗まれたカードを使用しようとしたとき、またはPINを間違って推測したときに、加入銀行からアラートを受け取るサービスを使用しています。ほとんどの泥棒はこれを理解せず、ただ捨てるだけです。  
レッスンは他の場所でも学びました。たとえば、米国国防総省は1999〜2005年に1600万の証明書を軍事要員に発行しました。これは彼らが砲撃されたとき[1299]。  
Google、Facebook、Microsoftなどの大手サービス会社は、何十億ものユーザーの資格情報を維持する必要があるため、他のWebサイトへのサービスとしてログオンを提供しています。  
ただし、一元化によってコストを削減できますが、中央サービスの妥協は混乱を招く可能性があります。  
7月9日、彼らは偽の証明書を生成し、イランの活動家のGmailに対して中間者攻撃を行いました。  
ハッキングは29日に公開され、Googleは9月3日にChromeからすべてのDiginotar証明書を削除し、Mozillaにも同様に対応させることで対応しました。  
  
ドキュメントで同時に作業している場合、バージョン管理システムを使用して、一度に1人のユーザーのみがドキュメントの特定の部分に書き込みアクセスできるようにするか、少なくとも競合を警告し、一貫性のない編集にフラグを立てます。  
別のアプローチはコールバックです。サーバーは、セキュリティ状態をサーバーに依存するすべてのクライアントのリストを保持し、状態が変化したときに通知する場合があります。  
ホテルを所有していて、顧客がチェックイン時にクレジットカードを提示する場合は、カード会社に事前承認を依頼します。事前承認には、近いうちに借方に記入することを記録しています。 「最大500ドル」の請求を登録する場合があります。  
障害モードの処理は注意が必要です。  
これは、分散システムで堅牢な承認を行う方法のpublish-register-notifyモデルの例です（[152]でより一般的な説明があります）。  
ただし、コールバックメカニズムは汎用的なソリューションを提供しません。  
別の例としてパスポートを考えてみましょう。  
実際、AadharバイオメトリックIDシステムをますます多くのトランザクションで使用するというインド政府の要求に対する頻繁な異議の1つは、すべての重要なトランザクションで市民の指紋または虹彩コードをチェックすると、彼らが行ったすべての場所の監査証跡が作成されることです。ビジネス、それは役人とそれらを買収することを気にする人に利用可能です。  
詐欺師がストアを偽装し、カードを使用して複数の大きな購入を行うことができるように認証をキャンセルする方法を考え出したときも、しばらくの間攻撃がありました。  
これらすべての詳細は、それ自体が本の章になります。クレジットカードなどの発行者に何らかの義務を生じさせるクレデンシャルと、パスポートなどのその他のクレデンシャルの間には、一般的な違いがあります。  
これは、更新が行われる順序が重要かどうかに関連しています。  
7.2.3更新の順序  
しかし、彼らが私の銀行口座に到着した場合、注文は結果に大きな影響を与えます。実際、トランザクションが適用される順序を決定する問題には、明確な解決策がありません。  
従来の銀行アルゴリズムでは、トランザクションを一晩でバッチ処理し、すべての借方を適用する前に、各口座にすべてのクレジットを適用していました。  
バウンスした支払いは取り消す必要があります。また、現金がすでになくなったATMおよびデビットトランザクションの場合、最終的に顧客が許可なしにお金を借りることになります。  
近年、国ごとにリアルタイムグロス決済（RTGS）が導入されています  
いくつかの微妙な欠点があります。  
第2に、結果はトランザクションの順序に依存する可能性があり、それは人間、システム、およびネットワークの変化に依存する可能性があり、金融​​機関間で非常に多くの支払いが行われる場合に問題になる可能性があります。  
2010年代後半、暗号通貨への関心の波により、一部の起業家は、ブロックチェーンが一貫性のない更新の問題を解決し、サプライチェーン管理などのアプリケーションを簡素化できると信じるようになりました。  
 私たちは待って見なければなりません。  
経験から、おそらく技術に非常に優れた少数の非常に大規模な銀行を持つことを除いて、一般的なケースで機能する魔法の解決策はないことが示唆されています。他のシステムでは、トランザクションが到着する順序はそれほど重要ではありません。  
パスポート発行者は、ビザが押される順序ではなく、作成日と有効期限のみを心配します3。  
7.2.4デッドロック  
Edsger Dijkstraは、この問題とその解決策について、食事哲学者の問題を通じて有名に説明しました。  
したがって、すべての人が一度に食べようとして、それぞれが右側の箸をつかむと、動けなくなります[560]。  
 [151]。  
プロセスが手動の場合、問題を回避するために多少の誤りが見つかるかもしれませんが、すべてがソフトウェアになると、このオプションは利用できなくなる可能性があります。  
昔は問題が解決し、会社が法廷に出た場合にのみ問題が解決する可能性がありました。それでも、一方の会社の条件はアメリカの裁判所を指定し、もう一方の会社はイギリスの裁判所を指定する場合があります。  
企業は、物事が失敗することを確実にするためにますます努力しています。  
  
分散システムの状態を更新するプロトコルを設計する場合、「マザーフッドとアップルパイ」はACIDです。そのトランザクションは、アトミックで、一貫性があり、分離され、永続的である必要があります。  
本のバランスを保つ必要があるなど、何らかの不変条件が保存されている場合は一貫しています。  
。  
これらのプロパティは、多すぎるか、不十分であるか、またはその両方です。  
4ジョセフヘラーの1961年のその小説は、第二次世界大戦の軍事官僚制度における一貫性のないクレイジーなルールの複数の事例を説明しています。  
これは、トランザクション量が減少した場合、最終的には一貫した状態になることを意味します[1353]。  
実生活では、問題が発生して完全に回復できないものを生き残る方法も必要です。  
 常に上昇し、時には単に説明できないことがあります。  
時々彼らは間違って推測します。そして、時には人々は、（もしあれば）ずっと後で理解されない脆弱性を見て悪用します  
最終的に、修正係数を追加し、それを一定の年間しきい値未満に維持するための目標を設定することにより、状況が悪化する可能性があります。  
フィッシング攻撃とキーロギング攻撃の出現により、銀行口座の一部の割合がいつでも犯罪者の制御下に置かれるようになりました。お金はそれらからそしてそれらを通して移動されます。  
フィッシャーマンは当然、取引の逆転を行わない、またはせいぜいゆっくりと不本意ながらそれを行う機関または管轄区域を通じて資金を移動しようとします[75]。  
  
セキュリティエンジニアが特に関心を寄せる最終的な同時実行性の問題は、正確な時間の提供です。  
1つの可能性はシンデレラ攻撃です。ファイアウォールなどのセキュリティが重要なプログラムにタイムロック付きのライセンスがある場合、攻撃者は時計を早送りして「ファイアウォールをカボチャに変える」可能性があります。  
時間は見た目よりもはるかに困難です。原子時計があっても、うるう秒は予測できませんが、何らかの方法でブロードキャストする必要があります。数分は61秒、さらには62秒です。奇妙な時間の影響はセキュリティの問題になる可能性があります6;そして世界の多くはグレゴリオ暦を使っていません。  
手形が盗まれたとしても、1882年手形法は、手形を誠実に購入した人々に良い称号を与えました。  
電子決済の場合、銀行はカルテルとして機能し、カードネットワークルールを通じて、および決済サービス指令を通じてヨーロッパの機関にロビー活動を行うことによって、より迅速に決済を完了しました。  
6一部のATMは、2000年後の数日間、顧客の残高を確認しなかったため、単語が一巡すると、承認されない当座貸越につながりました。  
絶対時間を放棄して、代わりにランポート時間を使用することができます。この場合、気になるのは、日付ではなくイベントAがイベントBの前に発生したかどうかです[1122]。  
多くのアプリケーションでは、ネットワークタイムプロトコル（NTP）を使用することになります。  
これは適度な量の保護を備えており、クロックサーバーとタイムサーバーの認証により、多くの目的で十分に信頼できます。  
たとえば、Netgearは、ウィスコンシン大学マディソン校にあるNTPサーバーを使用するようにホームルーターをハードワイヤードしました。 Netgearは、3年間のタイムサービスを維持するために375,000ドルを支払う必要がありました。第2に、2016年以降、NTPサーバーを強制的な乗数として使用するサービス拒否攻撃が発生しています。何百万ものサーバーが悪用可能であることが判明したため、多くのISPやIXPでさえもそれらをブロックし始めました。  
  
多くの場合、障害回復はセキュリティエンジニアリングの最も重要な側面ですが、最も無視されていることの1つです。  
しかし、現代の情報ビジネスの実際の支出（銀行であろうと検索エンジンであろうと）は逆です。  
この本を読むと、盗難警報から電子戦、企業をDDoS攻撃から保護することまで、他の多くのアプリケーションが基本的に可用性に関するものであることがわかります。  
従来のフォールトトレランスは通常、冗長性に基づいており、ログやロックなどのメカニズムを使用して強化されており、これらのメカニズムに対する悪意のある攻撃に耐えなければならない場合は非常に複雑です。  
私は次の定義を使用します。障害はエラーを引き起こす可能性があり、これは誤った状態です。これは、システムの指定された動作からの逸脱である障害につながる可能性があります。  
故障前の平均時間（MTBF）の意味  
 明らかなはずです。  
7.3.1故障モデル  
これの多くは、システムの動作環境に固有の脅威の分析からもたらされますが、いくつかの一般的な問題が言及されています。  
7.3.1.1ビザンチン障害  
ビザンチンの失敗は、ビザンチウムを擁護するn人の将軍がいて、そのうちtが攻撃するトルコ人に賄賂を渡されてできるだけ多くの混乱を引き起こしているという考えに触発されています。  
。  
 重要な観察は、私たちに将軍が3人しかいない場合、たとえば、アンソニー、バジル、キャラランポスであり、アンソニーが裏切り者である場合、彼はバジルに「攻撃しよう」とチャランポスに「撤退しよう」と伝えることができるということです。  
それはバジルと同じくらい簡単にできたでしょう。アンソニーはどちらにも「撤退しよう」と言ったかもしれないが、バジルが「アンソニーは攻撃しよう」と言ったときに嘘をついた。  
もちろん、将軍がメッセージに署名できる場合、2人の同僚に異なることをあえて言う将軍はいない。  
現在、ビザンチンフォールトトレランスに関する重要な文献があります。この種の障害に耐えることができるシステムの詳細な設計。たとえば、Miguel CastroとBarbara Liskov [394]によるアルゴリズムを参照してください。  
 答えがないだけでなく間違った答えを出すと、それを使用して弾力性のあるシステムを構築するのがはるかに難しくなります。  
  
したがって、さまざまな方法で故障率を抑制することができます。  
後者はデータとともにエラー訂正情報を処理し、不整合が検出されると停止します。たとえば、銀行取引の処理は、通常、処理タスクの後に不均衡状態が検出された場合に停止します。  
フォールトトレラントマルチプロセッサ（FTMP）に関する先駆的な作業がありました  
このような研究は、さまざまな潜水艦や宇宙船で使用されているフォールトトレラントプロセッサ、およびボーイングとエアバスで使用されているアーキテクチャの設計を推進することになった。  
Stratusには2つのディスク、2つのバス、さらには2つのCPUがあり、それぞれがエラーを検出すると停止します。フェイルストップCPUは、同じカードに2つのCPUチップを搭載し、それらの出力を比較することによって構築されました。  
代わりのカードがポストに到着します。あなたはそれを機械室に降ろし、カード5が点滅する赤いライトを持っていることに気づき、それを引き出して新しいものと交換します–機械が毎秒数十のトランザクションを処理している間。  
Googleは関連するソフトウェアスタックの開発におけるリーダーであり、2000年代初頭に、専門のベンダーからこれまでよりも大きなサーバーを購入するよりも、市販のPCとスマートソフトウェアで大規模なシステムを構築する方がはるかに安価であることを発見しました。  
まず、より複雑なソフトウェアスタックとツールチェーンを処理する必要があります。  
第2に、バックアップデータのあるサイトが複数ある場合、そのうちの1つが危険にさらされると機密性が失われる可能性があります7。破棄する義務があるデータがある場合、複数のバックアップテープからデータをパージすることは頭痛の種になる可能性があります。  
不注意のための他のトラップがあります。  
彼はそのカードが本物であることを強く主張した。  
何が起こったのか、そしてそれを再構築できるのはこれでした。  
前者はエラーを検出するために存在し、後者は偽造を検出するために存在します。暗号チェックサムが失敗しました。  
冗長性は、機械システムで処理するのに十分なほど困難です。  
初心者のパイロットは、実際には、単一のエンジンよりもマルチエンジンの飛行機でエンジンの故障によって殺される可能性が高くなります。最寄りのフィールドに着陸することは、突然の非対称推力に対​​処するよりも危険が少ないです。  
航空機は、多くの最新の情報システムよりもはるかに単純ですが、パイロットが安全性を維持することになっている冗長性の管理にパイロットが失敗すると、依然として航空機事故が発生します。  
多くの場合、システム設計者は複数の保護メカニズムを導入し、その結果を十分に注意深く検討していません。  
  
システムに冗長性やその他の回復力メカニズムを導入する場合、それらが何のためにあるのか、そしてさまざまな主体が直面しているインセンティブを理解する必要があります。  
最初のケースでは、複製をサーバーの内部機能にして、サーバーをより信頼できるものにすることができます。  
。  
これらは現在、クラウドサービスアーキテクチャの標準コンポーネントです。数十万のサーバーを運用している企業では、非常に多くの障害が発生するため、リカバリを大幅に自動化する必要があります。  
サービスは、クライアントの一部がそれをだまそうとしていると想定する必要があり、完全に正確なものではないいくつかのサービスに依存する必要がある場合もあります。  
信頼の決定には複雑なロジックが含まれる可能性がありますが、電子戦で使用されているどの入力が妨害されているかを解明するために使用されるシステムとはまったく異なります。  
 不信感の方向はプロトコル設計に影響を与えます。  
誰もが信頼できず、すべてが相互に疑わしい現実世界向けのシステムを設計することは困難です。  
ここでの明らかな例は銀行カードです。銀行は、マグストリップからチップにアップグレードする場合でも、安価なチップからより高度なチップにアップグレードする場合でも、カードの新しいバージョンを郵送することでセキュリティを時々アップグレードできます。また、影響を受けた顧客にサイクル外のカードを郵送することにより、侵害から回復できます。  
  
システムは、さまざまなレベルでのエラー、攻撃、および機器の障害に対する耐性を持たせることができます。  
前述のStratusシステムやRAIDディスクなど、一部のコンピュータはハードウェアレベルで冗長性を備えて構築されています。  
次のレベルでは、プロセスグループの冗長性があります。  
これにより、機械的な破壊によるか、不正なソフトウェアの挿入によるかを問わず、対戦相手がマシンに物理的にアクセスして破壊するような攻撃を阻止できます。  
次のレベルはバックアップで、通常はシステムのコピー（チェックポイント）を取得します  
コピーは通常、書き込み禁止のテープや特別なソフトウェアを使用したディスクなど、上書きできないメディアに保存されます。  
詳細、バックアップ、および回復メカニズムは、物理的な資産の破壊からの回復を可能にするだけでなく、論理レベルで攻撃を受けた場合でも、回復の見込みがあることを保証します。  
銀行や小売業者などの重要なサービス要件を持つ企業には、長年バックアップデータセンターがありました。  
そのような設備を維持することは、典型的な銀行の情報セキュリティ予算のほとんどを吸収しました。  
フォールバックシステムは、通常、メインシステムが使用できないときに処理が戻る能力の低いシステムです。  
フォールバックシステムは、アプリケーションレイヤーの冗長性の例です。これは、私たちが配置できる最上位のレイヤーです。  
冗長ディスクは、すべてのアカウントファイルを削除する悪意のあるプログラマから保護することはできません。また、ファイルを削除するだけでなく、エラーをゆっくり挿入するコードを作成する場合でも、バックアップは彼を止めることはありません8。  
一方、世界最高の暗号化は、データ処理センターが焼損した場合には役立ちません。  
冗長性の難しさ、およびそれを適切に計画およびトレーニングする絶対的な必要性について先に述べた発言は、システムバックアップに適用されます。  
1990年代初頭までに、英国のスーパーマーケットであるテスコは、実地訓練まで進んでいました。彼らは、オペレーターに警告せずにメイン処理センターのプラグを1年に1回抜き、バックアップが40秒以内に確実に来るようにしました。  
2019年までに、大規模なサービス会社は、これを必要としないほどの規模になっています。したがって、今日では、お金を支払うだけでクラウドサービスプロバイダーは多くの詳細について心配することになります。  
主要なプロバイダーの標準サービスレベル契約により、サービスを月に数時間中断することができます。小規模なクラウドサービス（政府のクラウドも含む）を使用する場合  
依存しているサービスのうち、直接のサプライチェーンの外にあるものを検討することは価値があります。  
政府は医師や看護師にガソリン配給を割り当てていましたが、学校の教師には割り当てていませんでした。  
これはストライカーがトニーブレア首相を倒すのに役立ちました。彼は着実に増加する燃料義務の彼の署名の環境政策を放棄しました。  
8今日、本当に真面目なランサムウェアオペレーターがシステムをハッキングし、ファイルの暗号化を不正に追加し、侵入するまで待機します。これにより、現在のデータだけでなく数週間のバックアップも人質にとどめます。  
セキュリティサービスをフォールトトレラントにしたい理由の1つは、サービス拒否攻撃の魅力を弱める、効果を下げる、またはその両方を行うことです。  
たとえば、セキュリティサーバーを停止して、他のサーバーに資格情報のキャッシュコピーを使用させたり、Webサーバーを一時停止して一時的にオフラインにしたり、別のマシンに被害者がダウンロードしようとしたページを提供させたりすることができます。  
プリンシパルが匿名の場合（ロードバランサーの背後に同等のサービスがいくつかあり、対戦相手がどのサービスを攻撃するのかわからない）場合は、効果がない可能性があります。  
これが不可能で、敵が攻撃する場所を知っている場合、冗長性と回復力のメカニズムによって阻止できるサービス拒否攻撃と、そうでない他のタイプのサービス拒否攻撃があります。  
この種の攻撃に対する防御には、専門のパケット洗浄ハードウェアを備えたより優れたホスティングサービスにサイトを移動すること、または犯人を追跡して逮捕することが含まれます。  
 攻撃は、1996年にニューヨークのISPであるPanixを数日間停止させるために使用されたときに、公に知らされました。  
2001年に、私はこの本の最初の版でそれらについて言及しました。  
典型的なターゲットはオンラインのブックメーカーであり、通常1万ドルから50,000ドルは放っておくために要求され、典型的なブックメーカーはこれが最初に発生したときに支払いました。  
 彼らは平均的なボットネットが彼らに投げかけることができるものをすくめることができるということ。  
2018年までに一周し、約50人の悪意のある人々がサービスとしてDDoS-as-a-a-serviceを運用していました。これは主に、対戦相手のチームスピークサーバーを停止したいゲーマー向けです。  
また、サービスオペレーターはそれらを、よりユーフェミスティックに「ストレッチャー」と呼んでいます。これを使用して、独自のWebサイトの堅牢性をテストできます。  
最後に、より脆弱なフォールバックシステムが存在する場合、一般的な手法は、サービス拒否攻撃を使用して被害者を強制的にフォールバックモードにすることです。  
スマートカードは一般に磁気ストリップカードよりも偽造が困難ですが、静電気と接触部の摩耗により、スマートカードは毎年1％失敗する可能性があります。  
そのため、ほとんどのカード決済システムには、磁気ストリップを使用するフォールバックモードがまだあります。  
  
命名は、通常の分散システムの面倒な面ではマイナーですが、セキュリティエンジニアリングでは驚くほど難しくなります。  
「Ross Andersonという名前の人はマシンXの管理を許可されている」とだけ言っている証明書はほとんど役に立ちません。  
私はまた、何十もの異なるシステムで異なる名前で知られています。  
概念的には、名前空間は階層的またはフラットにすることができます。  
しかし、これらは同じ種類のものではなく、それらをリンクすると、あらゆる種類の問題が発生します。  
「これがマシンXを管理するための鍵です」と単純に述べている公開鍵証明書は、金属製のドア鍵と同じように、無記名トークンです。ルートパスワードが銀行の金庫の封筒にあるかのように、その証明書の秘密鍵を管理するのはだれでも管理者です。  
私のパスポートが侵害された場合、その結果は広範囲に及ぶ可能性があり、政府の1人に私の名前で偽のパスポートを発行するインセンティブを政府に与えたくありません。  
たとえば英国では、チケットを購入したクレジットカードだけを使用して国内線に搭乗することはできません。パスポートまたは運転免許証を提出する必要があります。また、支店で銀行振込を1000ポンド以上で注文し、アパートを借りたり、弁護士を雇ったり、仕事を得たりする必要もあります。世界がより大きくよりフラットな名前空間に向かって移動している2つ目の理由は、オンライン認証における大規模なサービス企業の優位性の高まりです。  
これらの企業は、認証の章で説明した技術的な外部性と経済学の章で説明するビジネスの外部性の恩恵を受けているだけでなく、ネーミングの問題のいくつかを解決しました。  
したがって、分散システムでのネーミングについて、何世代ものコンピュータサイエンス研究者が学んだことを振り返ることは有益です。  
7.4.1ニーダムの命名原則  
名前をアドレスにバインドするために使用される基本的なアルゴリズムはランデブーと呼ばれます。名前をエクスポートするプリンシパルはどこかにそれをアドバタイズし、インポートして使用しようとするプリンシパルはそれを検索します。  
分散システムを構築している人々は、ネーミングがすぐに複雑になることにすぐに気付き、その教訓はニーダムによる古典的な記事[1424]に示されています。  
1。  
これは引き続き保持されます。私の銀行口座番号は、先週預金した情報を、今週出金しようとしている出納係と共有するために存在します。  
デポジットした金額とまったく同じ金額を引き出したいだけの場合は、無記名預金証書が適しています。  
2。  
これは復讐で成り立つ。  
したがって、各システムは他のシステムに依存しており、1つのシステムへの攻撃は他のシステムに影響を与える可能性があります。  
標準的な攻撃は、詐欺師が電話会社にあなたであると主張し、電話の紛失を報告することです。  
電話会社はそれを止めることができますが、それが失うすべての状態は、わずかな通信時間であり、限界費用がゼロであるので、認証についてあまり気にしません。  
Googleは、AndroidプラットフォームをAppleと同じくらい厳しくロックダウンすることでそれを止めることができましたが、そうするインセンティブがありません。  
それほど多くの名前が必要になると想定するのは悪いことです。  
、十分に議論されています。  
発行者は当初、13桁で十分であると想定していましたが、システムは数万の銀行（多くは数十の商品）で終わったため、6桁の銀行識別番号が必要でした。  
また、エラーを検出するためのチェックデジットもあります。  
グローバルな名前はあなたが思っているよりもあなたを買いません。  
ただし、私たちがビジネスを行うには、私の側のローカル名をこの一意の名前に解決し、最後のローカル名に戻す必要があります。  
。  
 私たちが保護しようとしているシステムとして。  
名前を追加するか、より複雑な名前を採用すると、追加のコストと障害モードが追加される可能性があります。  
名前はコミットメントを意味するので、組織の変更に対処するのに十分柔軟なスキームを維持してください。  
そこで、プリンシパルの秘密鍵はメールアドレスから生成されます。  
6。  
この例は、第2章と第3章ですでに確認しています。  
たとえば、ノルウェーはかつて市民のID番号を公開していると見なしていましたが、結局のところ、多くのアプリケーションで一種のパスワードとして使用されており、非公開にする必要がありました。  
。  
名前がパスワードとして使用され始めたときに問題が発生する例をいくつか示しました。  
私が大学で使用していた駐車場に入るには、自分の姓と駐車バッジの番号をバリアのマイクに向かって話さなければなりませんでした。  
 実際、それは「アンダーソン」でした。誰でも駐車場を歩いて、フロントガラスの駐車許可証から有効なバッジ番号を書き留めることができるからです。  
間違った名前が明らかな場合、物事ははるかに簡単になります。  
支払いシステムでは、クレジットカード番号が有効であるように見える限り（つまり、最後の桁が最初の15桁の適切なチェックディジットである限り）、端末がオフラインのときにクレジットカード番号が受け入れられていました。  
最新のチップカードの証明書は、同じ基本概念の高品質な実装を提供します。暗号化やセキュリティ印刷などの認証メカニズムは、名前をなりすましに対して回復力のあるものにするという追加の利益を与えることができます。  
8。  
ディレクトリーが複製されている場合、使用可能なディレクトリーが多すぎるか少なすぎるかによって、読み取りまたは書き込みができない場合があります。  
これは理論的には簡単ですが、製品ごとに一意の数値コードを使用していますが、実際には、製造業者、販売業者、小売業者によって、データベースのバーコードにまったく異なる説明が付けられています。この問題を修正する提案は、驚くほど複雑になる可能性があります[914]。  
製品がリコールされたかどうかなど、適時性の問題もあります。  
賢くなりすぎないでください。  
初期の安全なメッセージングシステム（PGPから政府システムまで）は、キーを電子メールアドレスにリンクしようとしましたが、これらは人々の仕事が変わると変わります。  
同じように、SETなどのプロトコルで銀行口座番号とクレジットカード番号を公開鍵証明書に置き換えるという初期の試みは失敗しましたが、ケニアのM-Pesaなどの一部のモバイル決済システムでは、電話番号に置き換えられています。  
 10。  
賢明なプログラマは通常、絶対アドレスまたはファイル名をコーディングすることを避けます。これは、マシンのアップグレードまたは交換を困難にするためです。  
しかし、ぎりぎりの解決に使用されるサードパーティのサービスが攻撃のポイントになる可能性があるため、安全なシステムは多くの場合、安定した説明可能な名前を必要とします。  
  
ニーダムの原則は、システム所有者の都合に応じてネーミングシステムを課すことができる1990年代初頭の世界のために作成されました。  
 グローバルな規模で運営されているサービス産業では、さらに追加する必要があることがわかりました。  
実際、これは古典的なトリレンマです。これらの属性は2つしか持てません（Zookoの三角形）。  
これまでエンジニアは、PGPの公開鍵やAndroidのアプリ名として機能する自己署名証明書のように、URLのように一意で意味のある、または一意で分散型のネーミングシステムを採用していました。  
前述のとおり、最初の検索エンジンが登場するとすぐに、ロスアンダーソンと呼ばれる他の何十人もの人々をすぐに見つけることができましたが、それよりもさらに悪いことです。半ダースは、ソフトウェアエンジニアリングや配電など、私がこれまでに取り組んだ分野でも働いていました。  
ソーシャルコンテキストを使用して、分散型で意味のあるシステムを構築できます。これは、私たちの脳が対処するために進化したものです。  
これをどうやって理解し、人々をつまずかせるのに使わないようにできるでしょうか？  
  
まず、セキュリティプロトコルのプリンシパルは通常、銀行口座番号、会社の登録番号、個人名に加えて生年月日または住所、電話番号、パスポート番号、健康状態など、さまざまな種類の名前で知られていますサービス患者番号、またはコンピューターシステムのユーザーID。  
IDは、2つの異なる名前（または同じ名前のインスタンス）の場合です  
。  
家を売ろうとする人は、購入したときとは異なる名前を使用することがよくあります。結婚や性転換の際に名前を変更したり、代わりにミドルネームを使用したりした可能性があります。  
侵害につながるIDの失敗には2つのタイプがあります。私が誰かになりすまして喜んでいる場合と、特定の個人になりすまそうとしている場合です。  
。  
 公共料金の請求書など、簡単な2つの住所証明を求めてください。  
これは、銀行とパスポートオフィスの2つの別々のシステム間のシンボリックリンクと見なすことができます。  
一般に、名前には再帰のいくつかのステップが含まれる場合があり、攻撃者はターゲットを選択できます。  
事後分析アプリケーションも一般的です。  
今では、Web検索により、ラッパーのTupac Amaru Shakurなど、多くの人々のそのようなデータがすぐに表示されます。確かに、Linden LabsはShakur氏のライセンス番号を受け入れました。ライセンスが失効していて、彼は死んでいます。  
たとえば、アラブ首長国連邦は、売春犯のためにパキスタンに強制送還された女性が数週間後に別の名前の本物のパキスタンのパスポートを持ち、別の「夫」を伴って出頭した後、すべての訪問者の虹彩スキャンを開始しました'。  
破損に加えて、広範囲にわたる障害は元のレコードの損失です。  
デジタル化がこの問題を修正していると思うかもしれませんが、デジタルレコードの長期保存は、豊かな国でさえ難しい問題です。ドキュメント形式が変更され、ソフトウェアとハ​​ードウェアが古くなり、古いマシンをエミュレートするか、古いデータを変換する必要がありますが、どちらも理想的ではありません。  
賢明な先進国は、紙の原本を長期の記録文書としてまだ保持しています。名前の根底にある仮定は、国によって異なります。  
しかし、一部の国ではエイリアスの使用が禁止されており、他の国ではエイリアスの登録が必要です。  
そして、国間の互換性のない規則は、移民、観光客、そして実際に海外の顧客を持つ企業にとって実際の問題を引き起こします。  
そのため、私の研究室の同僚である故カレン・スパールク・ジョーンズ教授は、大学から毎年何も発表しなかった理由を尋ねる手紙を受け取りました（彼女はカレン・ニーダムとして給与を受け取りました）  
出版物追跡システムは、人事システムが知っているすべてに対応できませんでした。  
たとえば、カレンは大英図書館の管財人でもあり、所有者の自宅の大学図書館カードの名前を使用して独自の入場券を発行し始めたときは感銘を受けませんでした。  
複数の名前を持つこれらの問題は現在主流です。複数の名前を使用するのは、教授、ミュージシャン、小説家だけではありません。  
グローバルに拡大することを望むシステムを構築している場合、最終的にはそれらすべてに対処する必要があります。  
中国人が香港出身の場合、英語と中国語の両方の名前が付けられ、英語の名前が前に来て、中国の名前が姓の後に来ます。  
インドの慣習では、2つのイニシャルを追加することになっています–出生地と父親の名前です。  
南インドから米国への移民の間で共通の戦術は、パトロニミック（ここでは、クマール）を使用することです  
ロシア人は、姓、愛称、姓で知られています。  
昔、「マリアトロスタッタッティル」が米国の入国管理局に到着し、警官が「トロスタッタティティ」が姓でも常習者でもないことを知ったとき、彼らの標準的な慣習は、彼女を姓として愛国者として採用させることでした。 （彼女の父親がカールと呼ばれた場合、「カールソン」と言います）  
アメリカのインディアンの多くは同様の問題を抱えており、そのすべてが不必要な犯罪を引き起こしています。  
もう1つの文化的格差は、プライバシーの観点からIDカードが受け入れられない英語圏の国と、IDカードが標準であるナポレオンまたはソビエトによって征服された国との間のものであると考えられています。  
IDのローカル履歴は、あらゆる種類の仮定を条件付けます。  
。  
ただし、アイスランドのIDカード番号は静的です。これは、市民の生年月日とさらに2桁の数字です。  
これらはおそらく、プライベートIDとパブリックIDの番号付けの両極端です。  
支配部族は、自分たちの領土で出生を登録することを困難にしたり、IDカードを取得することを不便にしたりすることにより、他の人々の権利を奪うことを求めるかもしれません。  
カードは、営業許可や福祉の支払いに結び付けることができます。賄賂を抽出するために遅延を使用できます。  
 州レベルと連邦レベルで別々の登録システムを持っているが、他のもの（マラウイなど）  
出身国の国籍の外で生まれた難民の子供や、宗教的または思想的な理由で無国籍になったグループなど、除外されたグループはたくさんあります。  
これらはあらゆる種類の複雑な方法で政府と相互作用し、これを研究する研究コミュニティ全体が存在します[88]。  
つまり、政府と人々の名前の関係についての隠された仮定は、システムの設計を制約し、仮定が国境を越えて運ばれるときに予期しない失敗を引き起こす方法で異なります。  
政府は永遠に2つを絡ませようとしていますが、これはあらゆる種類の痛みにつながります。  
7.4.2.3名前の意味的内容  
銀行は、口座番号による顧客データの保管から、名前と住所による保管に移行した後に訴訟を起こされました。  
ある顧客への影響は深刻でした。愛人のために保管していた口座の銀行取引明細書は、彼と離婚した妻に送られました。  
多くの交通機関では、チケットや通行票は現金で購入できるため、プライバシーの懸念はありませんが、銀行口座にリンクする方が便利であり、これらのリンクは時間の経過とともに蓄積されます。  
実際、私はかつて、ランダムなアカウント番号（信用調査なし）のハードウェアストアのポイントカードを入手しました  
店がスーパーマーケットに引き継がれ、スーパーマーケットが銀行を始めた後、これを銀行カードに変更する機会が与えられました。7.4.2.4名前の一意性  
私たちは名前だけで始めましたが、中世後期までに旅行の増加により政府は人々をいじめて姓を採用するようになりました。  
インターネットへの人々、企業、行政の大規模な移動は、社会的適応には速すぎます。  
 私たちが慣れているよりもオンラインで。  
 上で説明したように、Facebookは、ジョンスミスに人間のように、友人のグループとクラスタリングし、写真を追加することで、それぞれを区別します。  
名前は、公開キーまたは名前が付けられているオブジェクトの他の安定した属性のハッシュです。  
。  
暗号通貨とブロックチェーンの世界では、ハッシュベースの識別子を多く利用しています。  
これは完全に新しいものではありません。なぜなら、トランザクション処理では、すべてとすべての人に番号を与えるだけで長い間一般的であったからです。  
たとえば、英国の銀行は、取引のキャプチャに使用されるステーショナリーに印刷することにより、トランザクションに一意のシーケンス番号を割り当てました。  
1000万ポンドの2回目の支払いが注文されましたが、これにより書類から同じトランザクションシーケンス番号が取得されました。  
  
多くの名前にはある種のアドレスが含まれていますが、アドレスは変更されています。  
1990年代後半に、暗号化された電子メールを使用する人々のディレクトリとそのキーを作成することを試みたとき、エントリの変更の主な原因は電子メールアドレスの変更であることがわかりました[103]。  
 物事はおそらく今より安定しています。  
GoogleやMicrosoftのような大手サービスプロバイダーは、通常、同じメールアドレスを2回発行することはありませんが、他の企業はまだ発行しています。  
ただし、階層ネーミングシステムには、複数の抽象化レイヤーが含まれ、各レイヤーのアドレス情報の一部が、上のレイヤーの名前の一部を形成します。  
多くの場合、人々は部門レベルと組織レベルで異なる名前になります（私の場合、rja14 @ cam.ac.ukとross.anderson@cl.cam.ac.ukなど）。  
したがって、名前とアドレスを明確に区別できるとは限りません。  
 アドレスのプロパティの。  
同様の問題が、名前が複合しているシステムを悩ませています。  
これは明らかに悪い習慣です。  
仮名を使用するのには十分な理由があります。  
22歳の大学を卒業する最初の仕事、またはCEOの45歳の仕事に行くとき、検索で10代の怒りをすべて起こしたくはありません。  
一方、一部の警察やその他の機関では、人々が偽名を使用しないことを望んでいるため、オンラインでのトレーサビリティの問題全体に私たちを連れていきます。これについては、パートIIで説明します。  
7.4.2.6名前の使用に関する制限  
これは、米国の他の一部の国における社会保障番号およびそれに相当するものと同様に、法律によって規定される場合があります。  
制限付きネーミングシステムは、予期しない方法で相互作用します。  
これは、健康維持組織の合併、またはポリシーの変更により、病院に統一された名前を導入するよう強制する場合に問題を引き起こします。  
最後に、私たちが法律と政策に来るとき、名前の定義は新しくて予期しない落とし穴を投げかけます。  
これにより、検索エンジンに渡されるパラメーターなどのデータを含むURLのステータスが乱れます。  
q =大麻+栽培;同様に、多くの人々は、そのような大規模なトロールをプライバシーの許容できない侵略であると考えています。  
’  
技術的なものから組織的なもの、政治的なものまで、すべてのレベルでネーミングが複雑であるだけでなく、本当に邪悪な問題のいくつかはレベルを超えています。  
だけでなく、組織、役割（「時計の役員」）にも  
論理積はしばしば暗黙のアクセスルールを表現します：「ブランチマネージャとして機能するアリスとブランチアカウンタントのグループのメンバーとしてのボブ」。  
名前はサービス（NFSや公開鍵インフラストラクチャなど）にも適用されます  
。  
セキュリティに関する文献で使用されている通常の抽象化は、それらを異なるプリンシパルとして扱うことです。  
多くの組織は、「アリス・イン・パーソナル」、「アリスの自宅のラップトップ上で実行されるプログラムとしてのアリス」、「コーポレートクラウド上のアリスの代わりに実行されるプログラム」を慎重に区別し始めており、この章で考えられるメカニズムのいくつかについて説明しましたアクセス制御。  
企業は主に給与を受け取りたいのに対し、政府は人々を一意に識別したいと考えています。インセンティブに基づく分析は、ネーミングシステムがオープンまたはクローズ、ローカルまたはグローバル、ステートフルまたはステートレスのいずれであるか、およびそれを維持する人々が失敗のコストを支払う人々と同じであるかどうかを示すことがあります（経済学は、信頼性の重要な問題であり、次の章の主題です）  
最後に、名前を付ける際の問題の多くを人々に関して説明しましたが、それにより問題がより迅速で説得力のあるものになるため、同じ問題の多くが暗号化キー、一意の製品コード、ドキュメントIDのさまざまな方法で現れます、ファイル名、URLなど。  
（パート2では、ネットワークセキュリティのより技術的な側面について説明します。）  
ネーミングのスコープを慎重に検討し、誰が依存する名前を制御するかを理解し、どれほど滑りやすいかを理解し、制限にもかかわらず信頼できるシステムを設計する必要があります。  
7.5まとめ  
 分散システム。  
多くのセキュリティ違反は、なんらかの同時実行性の障害です。システムは古いデータを使用したり、更新を不整合にしたり、間違った順序で行ったり、データが一貫していない場合や不可能な場合でも一貫性があると想定します。  
フォールトトレランスと障害回復は重要です。  
より技術的なレベルでは、保護メカニズムと回復力メカニズムの間に重要な相互作用があります。  
冗長性にはさまざまなフレーバーがあり、適切な組み合わせを使用する必要があります。  
多くの問題は、名前を過度に使用させようとしたり、特定のシステム、文化、または管轄外では保持されない仮定を行うことからも発生します。  
最も簡単な解決策は、多くの場合、銀行口座番号やシステムログオン名など、他の目的では使用されない一意の識別子を各プリンシパルに割り当てることです。  
時々これは偶然に起こることさえあります。  
しかし、おそらく最も重要な研究の問題は、悪意に直面しても復元力があり、優雅に低下し、攻撃が過ぎるとセキュリティを回復できるシステムを設計する方法を見つけることです。  
しかし、これらの魔法の弾丸はゴブリンを殺すようには見えません。  
現在、英国政府の検証スキームなどの失敗したIDシステムが増えています。これは、2019年に廃止された公共サービス用のフェデレーションログオンシステムを作成する試みです[1392]  
そして、ブロックチェーンが当初の約束を果たすことができなかったのです。これについては、この本のパート2で説明します。  
マルウェアとフィッシング攻撃は、常に小さな（ただしゼロではない）ことを意味します  
しかし、銀行システムは継続します。  
古典的なコンピュータサイエンス理論では、分散システムでの収束は本質的に技術的な問題であり、その解決策は技術的な特性に依存していました（あるレベルでは、原子性、一貫性、分離、耐久性。別のレベルでは、デジタル署名、二重制御および監査）。  
多分私達は私達がどのように世界の状態について十分な合意を得て、技術的回復力メカニズムと保護技術だけでなく詐欺の犠牲者であった人々が救済を得るメカニズムも組み込んでいるかを尋ねるより高いレベルの見方が必要です。  
参考資料この章の内容に不慣れな場合は、数学/暗号のバックグラウンド、チップ/エンジニアリング、または法律/ポリシーの問題に直面している可能性があります。  
長年にわたって私たちが生徒たちに勧めてきた他の本には、タネンバウムとヴァンスティーン[1860]とマレンダー[1353]があります。  
 「プライバシーのレンズによる認証」では、認証とプライバシーのトレードオフ、およびそれらがどのように不十分に拡張される傾向があるかについて説明します[1039]。