Microsoft Informed Trust Management Architecture

1997 年 5 月

昨年 10 月に Authenticode（認証コード）が発表されるまで、多くの人は、インターネットのブラウジングを見物人の楽しみのように考えていました。オンライン ブックを匿名で読んでいるだけではないということには思いも及ばず、活発に情報やプログラムをやり取りしていました。その間に、サイトへのすべてのアクセスは訪問カードとして記録され、あらゆる動画はプログラムを取り出します。セキュリティ ホールの存在を明らかにして、一時だけ有名になるハッカーたちは、本当の悪者たちが裏に隠れているという認識を高めてくれます。変わったことをするのにも、価値があるというわけです。

このことから判断すると、新しい世代のインターネットにおけるセキュリティやプライバシーについて再考する必要があります。従来のネットワーク セキュリティでは、悪意のあるクライアントからサーバーの大切な資源を守ることに力を注いできました。しかし今では、"クライアント" の価値ある資源を悪意のあるサーバーから守る、ということも問題となっています。情報というのは、インターネット コンテンツの発行者と消費者の間で双方向にやり取りされるものですから、インターネット セキュリティに真の違いをもたらすために、両者の間で協力していくことになります。必然的に、お互いがある程度まで信頼し合わなければならず、逆に、信用する根拠のない相手とやり取りするときには、システムの正常な状態を危険にさらすようなことは、ある程度避けたいというのも当然のことです。

このような状況での信用というのは、直観や経験、そして情報によって生まれるものです。結局、考えられたのは、危険をもたらす可能性のある操作を避けるのではなく、すべてのアクションには危険が伴うと認識した上で、そのアクションを認めるかどうか、情報に基づいた (informed) 判断を行うためのツールをユーザーに提供するというものです。

この文書では、Microsoft の Informed Trust Management Architecture を照会し、次の 3 つにおいて実現される内容を説明します。

- Authenticode 2.0
- Java
- Internet Explorer 4.0 および Internet Explorer Adaptation Kit

背景情報

インターネットがネット サーフィンのオープンな手段であるという事実から考えて、危険な楽しみ方はなくならないでしょう。しかし、海を眺めてずっとビーチにいたのでは、つまりません。正しいアプローチは、前もってリスクを認識してから判断するという方法です。インターネット セキュリティのことばで言うと、これは "情報に基づく信用管理 (informed trust management)" です。

信用管理は、リスク管理に比較してみると一番良く分かります。あらゆる種類の保険プランは、リスクを避けるのではなく、リスクを管理するためのものです。信用というのは、身近な概念です。最も単純な形の信用は、次のものに基づいた、人間の直感的な概念構成です。

- 認識 (Recognition)
- 評判 (Reputation)
- 照会 (Referral)

認識 (Recognition) は、最も信頼できます。これは1対1で見分けられるもの、つまり、顔、声、または、既知の情報に基づく推定や慎重な順位付けによる、ブランドなどです。認識の問題点は、スケーラブルでないということです。その点、評判 (Reputation) は比較的優れています。評判は、間接的かつスケーラブルで、中には「評判は高いがお粗末な食事」というような、信頼できないものもよくありますが、企業能力の付加財産を確かに持つものです。"評判という財産" は重要であり、企業は、一貫した品質提供を基にこれを築き上げます。手に入れるのは難しく、失うのは簡単です。照会(Referral) は、信用の中で最も複雑な形です。というのは、「私の信頼する医師が、私の病状に最適の医師だと話している」といったように、常に過渡的なものだからです。評判と照会を取り混ぜて使用することによって、デジタルの世界で効果的に信用を確立できます。

このための素材は、公開鍵暗号化方式、つまり、デジタル署名、証明、暗号化などにあります。これらのものすべてによって、だれが要求を行い、要求がなにを意味するかを判断したり、プライバシーを確立したり、ダウンロード可能なコードについては "shrink wrap" の完全性を確認するための手段が提供されます。

インターネット上の信用を築く

いかなる信用提案も、「だれとやり取りしようとしているのか」という簡単な疑問から始める必要があります。専門用語ではこれを認証 (authentication) と呼び、エンティティとデータ発信元の両方に適用されます。インターネット上で確信の持てる認証は、デジタル署名およびデジタル証明に依存します。

デジタル署名 (digital signature) は、メッセージをその発信者に関連付けるための暗号化の手段方法です。さらに、デジタル署名の使用によって保証されるのは、署名された文書の受信者が、その署名が本物であるかどうか照合でき、おそらくなにも変更されていない形でそのメッセージを復元できるということです。署名の確認は、添付される証明を用いて行います。証明 (certificate) は、識別された加入者に 1 つの公開鍵がバインドされていることを証明し、特定のポリシーのもとに発行される、コンピュータによる記録です。さらに正確には、証明は認証以前の証しです。ここで言う認証とは、相手の顔を運転免許証の写真と見比べるとか、郵送された秘密文書の内容を相手が確かに受け取ったかどうかを確認するといった、従来からある手続きを意味します。証明の唯一の目的は、過渡的な信用を設定することです。たとえばある文書に関する確認のために、その文書に見知らぬ人の次のような署名があったとします。
"ここに含まれる私の作成したコードには、なんの悪意もありません。
Sam Shareware より "
このSamという人物 (このソフトウェアのユーザーにとっては未知の人) を信用できるのは、照会があった場合だけです。この照会のまずはじめに必要なのは、Sam が自分から出向いて、証明を付与されるためのプロセスに進んで申請したという事実です。 Authenticode モデルにおいては、営利 Web 発行者の証明を獲得するために、開発者は義務ではない認証テストに合

格して (これが本当にビジネスでしょうか?)、悪意のないソフトウェアのみに署名することを宣言した誓約書に同意しなければなりません。これでやっと、明示的に信用される Certificate Authority から証明が発行されます。
ここに、重要かつ相互依存性の高い 4 つの概念を挙げます。

- 信用の基準には最低限、署名と証明が含まれる。
- 署名は、それを確認できる証明がなければ適正ではない。
- 証明は、関連するポリシーがなければなんの価値もない。
- 署名した人が、証明が発行されたときのポリシーを守り、これを破ったときにその責任を負うのでない限り、全体の機構が無意味となる。

適例: IP Spoofing が簡単になると、URL 読み取りは、サイト認証の証しとしては不十分になります。SSL や IPSEC などの安全なプロトコルを使ってサイトの証明を要求すると、次レベルの確証が得られます。

権限 (Authorization) は、なんらかのアクションを行うための許可のことです。 Authenticode 1.0 を使用した場合、デジタル署名によってメッセージの完全性が確認され、実行が許可されるかどうかという、信用判断の基盤になります。

Authenticode 2.0 の使用によって、ソフトウェア 発行者は自社のコードに "日付スタンプ" を追加できます。証明は、大型の暗号鍵のペアを使用して作成されますが、この暗号鍵には、無理なく時間内に鍵が使われるであろうという予想に基づいた、実用的な予測寿命があります。有効期限を証明に適用するときは、期限までに余裕を持たせるようにします。時刻スタンプは、この暗号鍵の期限内に署名が行われ、ゆえに、その署名自体が有効であることを保証するものです。

また、Authenticode 2.0 は、Java のためによりきめの細かいセキュリティを備えています。デジタル署名では、メッセージの完全性を確認し、発行者のモーバイル コードを認証できます。それに対して、認証において判断が提示されるのは、実行許可だけです。いちかばちかです。実際的には、だれが発行したかよりも、"このモーバイル コードに必要なのはどの資源か" を知ることの方が大切な場合が多いのです。たとえば、Microsoft は信用できるかもしれませんが、自分のシステムにある任意のファイルをオープンできてしまうコードは、インターネットからダウンロードしたくはありません。

Java では最初、"サンドボックス" モデルをセキュリティのために定義しました。このモデルでは、ネットワークからロードされた Java クラスには、極端に限られた機能に対する権限が与えられ、ローカル ディスクからロードされたクラスには実質的にすべてのことができるフリー権限を付与しました。このバイナリ方式の信用モデルでは、たくさんの面白い Web アプリケーションにネットワークから実行するための書き込みができず、その一方で、無制限のローカル クラスが自ら、または悪意のある信用できないアプレットから呼び出されたときに、なんらかの悪いセキュリティ ホールを偶然にオープンしてしまうことがありました。Microsoft の Authenticode署名テクノロジと Sun の JDK 1.1 によって、ネットワークからロードされたアプレットに署名する機能が追加され、ローカル アプリケーションと同じ特権を活用できたものの、Java セキュリティのいちかばちかという性質はなくなりませんでした。

Java の信用

Netscape や Sun が発表している Java の他の拡張セキュリティ モデルと同じように 、Authenticode 2.0 ではまず、中間レベルの信用を既存の Java セキュリティ モデルに追加します。仮想マシンの管理オプションが拡張され、スクラッチ空間、ローカル ファイル、およびネットワーク接続へのアクセスといった、Java コードに権限が認められている機能に関して詳細な制御が取り入れられます。これによって、システム内のすべての機能へのアクセスを無制限に認めることなく、特定の機能をアプリケーションに追加することができるようになります。これはつまり、最高レベルの信用を持ったシステム クラスのみがモデルを拡張できるということです。

さらに大切なのは、アプレットを信用すべきかどうかの判断を、ロードする "前に" 表示します。Java の 2 つの特性 (ストロング タイピングと、仮想マシンで稼動するということ) は、セキュリティ上の利点です。開発者は、自社のコードにどの資源が必要か確認して、仮想マシンにそれを実行させることが可能です。機能サイニングを使用すると、アプレットの要求する機能のリストをそのアプレットの署名の中に含めることができます。(これは、Netscape のモデルが、Java ソース コード内にハードコードされた機能要求の埋め込みをアプレットやクラス ライブラリに強制するのとは対照的です。) アプレットのために要求される許可 (たとえば、そのアプレットが接続することを認められるホスト) は、コードとはまったく別個のものなので、そのアプレットを再コンパイルする必要はなく、イントラネットの管理者が許可を拡大したり削減することができます。また、コードの実行中、新しい機能が要求されるたびに信用ダイアログをユーザーに表示するという、ユーザー インターフェースの "ノイズ" も減ります。
新しい Package Manager は、機能サイニングを使用して、完全には信用されていないローカル クラス ライブラリをインストールできるようにします。これは、Java Beans やクラス ライブラリにとって特に重要なものです。これらのコンポーネントをローカルに常駐させておいて、いくつかの拡張機能を手に入れられるのが望ましいのですが、これらのコンポーネントに無制限の権限を与えてしまうのは適切ではありません。これとは対照的に、 Netscape や Sun のモデルでは、すべてのローカル ライブラリが完全に信用できるものでなければなりません。
したがって、これは Java にとってすばらしいことです。理解しやすい方法で表示された詳細な判断が容易に行えます。しかし、全体的に見ると、Java は 1 つのコンポーネントです。サイト自体はどうなのでしょうか?

他のものをすべて信用する

歴史的に見ると、サイトのコンテンツを基準として差別化する手段の必要性が高まったのは、多くの人にとって厚かましくも無礼なサイトが多いということが認識されたためです。W3C の PICS (World Wide Web Consortium Platform for Internet Content Selection) http://www3.org/pub/WWW/PICSは、1995 年に自発的セルフ ラベリングのインフラストラクチャを定義するために開始され、すでに 100 万以上のサイトがラベルを取得しています。
これは、明らかにまだインターネットのごく一部の活動とはいえ、大きな一歩です。同時に、はじめにサイト、次にコンテンツの順に信用判断が始まるのだ、というメッセージを強く示しています。ページ上のコンポーネントとホストが信用の点で異なる性質を持つという可能性は、ありえないとは言えませんが、ほとんどないでしょう。それよりも考えられるのは、たとえば、コンポーネントにハードドライブへの書き込みをさせたいかどうかといった特定の機能こそ、実行を許可するか拒否するかの最終的な判断の差を生むのです。

残念ながら、Java Capability、Scripting、ActiveX など、Web で利用できる機能が急増したことによって、Web も大幅に複雑になりました。「自分のマシンではなにもトラブルを起こさせないように」という以前は単純だった考えは、複雑な Web の中で相互に入り組んだ選択肢をユーザーにせまるものとなってしまいました。

以前のセキュリティ機構では、遭遇したすべての Web サイトおよび Web ページに対して一律のセキュリティの判断を行うことを、エンド ユーザーや管理者に任せていました。Java はつねにオンであるか、またはつねにオフのどちらかの状態でした。 ActiveX コントロールもつねに利用可能か、またはつねに利用不能でした。この単純な機構は、今や Web 成功の中で犠牲となっています。

以前は、すべての Web コンテンツはほぼ同じレベルのフィールドで扱われていました。
実際大したことができなかったので、すべてのコンテンツには同じレベルの機能しかなかったのです。すなわち、実質的にはなにも機能はありませんでした。

しかし、企業は最近では、Web ブラウザや HTML を組み合わせてパワフルな社内用アプリケーションを作成できることを知っています。C++ や Visual Basic で作成して、さらに社内にインストールしなければならなかったシステムが、今では、HTML ですばやく作成できてしまいます。運用保守やインストールのコストは大幅に削減され、直観的なユーザー インター

フェースも用意され、言うことはありません。

ところが、実はこれでは十分ではありません。これらのアプリケーションは、アプリケーションに似た機能 (ディスク書き込み、ネットワーク入出力) を実行する Java を使用する必要があり、あるいは、同じ機能を実行するために ActiveX コントロールを使用する場合もあります。これ以上、すべてのコンテンツにユーザーのハード ドライブへの均一なアクセスを認めるわけにはいきません。ここでの解決策は、ユーザーに尋ねるという月並みな方法です。

たとえば、ユーザーがサイト A をブラウズしにいきます。すると、ブラウザはサイト A に次のような機能があるかどうかを義務的に尋ねます。

ディスク書き込み、ネットワークオペレーションの実行、ActiveX Control の実行、ActiveX コントロールのスクリプティング、1MB 以上のメモリー使用、フレーム間操作の実行

なんと、これらは多くのユーザーが答えられるような質問でしょうか? 少なくとも、確信を持って答えることはできないはずです。これで安全な作業環境が作り出されるかというと、絶対にそうではありません。ユーザーの判断に任せるということは、ユーザーとコンテンツの間に一種の契約を締結するということです。一方の当事者がその結果がどうなるか完全に理解していないのに、正当な契約を結べるはずがありません。

信用のゾーン

少しの間、すべての質問にユーザーが絶対的な自信をもって答えられると仮定してみましょう。これによって作られるのは、無限に多くの許可レベルをもつセキュリティ モデルではありません。ユーザーは一般的に、関連するサイトをグループ化する傾向があります。よくあるのは、次のようなグループです。

- インターネット
- イントラネット
- 信用できる Web サイト
- 信用できない Web サイト

インターネット

"ブラウジング" が最も多く行われるのが、このグループです。インターネットから取り出されたサイトは、信用できるサイトと信頼できないサイトのいずれでもありません。

イントラネット

このグループは、企業のファイヤーフォール内のサイトです。社内のサイトは絶対的に信用できるで、Web アプリケーションが利用されています。これらの Web アプリケーションは、ユーザーのハードドライブにアクセスしたり、破壊につながる可能性のあるその他の操作にアクセスする必要があります。

信用できる Web サイト

これらは、ユーザーが日常的に何度もアクセスするような、よく知られているサイトです。この中には、企業の関連会社や、信用のおけるビジネス パートナーの Web サイトなども含まれます。これらのサイトへのアクセス頻度は、イントラネット上のサイトよりも少ないのですが、インターネット上で無作為にアクセスするサイトよりは多くなります。

信用できない Web サイト

これらは、ハードドライブの消去など、システムを脅かすサイトであることを、システム管理

者が新聞で読んで知っているサイトです。これらのサイトには、なんの許可も与えてはいけません。いわゆる、ブラックリストです。

関連サイトのための共通セキュリティ設定値モデルである Trust Zone は、Internet Explorer 4.0 における重要な新機能です。

Trust Zoneとは?

Trust Zone は 2 つのことを行います。

1. 複数のサイトをまとめてグループにする
2. ゾーンにセキュリティ設定を割り当てる

ActiveX コントロールのダウンロードおよびインストール、スクリプティング、クッキー管理パスワード認証、フレーム間セキュリティ、そして当然ながら Java Capability に対するすべての設定値は、サイトが属している Trust Zone を基準にして制御できます。

Trust Zone は、ユーザーや管理者がセキュリティ オプションに関して微調整コントロールできるようにすることによって、エンド ユーザーがブラウジングのセッション中に侵入してくる応答不能なダイアログに遭遇する回数を減らします。Trust Zone は、ローカル セキュリティの規則やポリシーを表し施行する 1 つの手段です。この "ポリシー" というのは、技術文書によく登場する用語ですが、特定のアクションが認められるための一連の条件を意味します。ポリシーの例としては、次のようなものがあります。

microsoft.com から入手したすべての実行可能ファイルにこのマシンで実行することを認めるが、ただし、ネットワーク接続のオープン要求があったときに、ダイアログをユーザーに対して表示する。

Aこれはまさに、Trust Zone が実行する内容です。

Trust Zone の構成

ユーザーは、各 Trust Zone のサイトおよびセキュリティ設定のリストに対して、完全な制御が可能です。IE4 は、ユーザーがゾーン設定を効果的に管理するためのオプション UI をすべて備えています。いったん設定されたゾーンは、新規サイトを追加したり、サイトを削除できます。サイトは、URL または IP アドレスで指定されます。ただし、特別なイントラネット ゾーンに対して、ファイアウォールの壁の向こうにあるすべての IP アドレスを含めるような設定も簡単に行えます。

ユーザーは、ゾーンごとにもセキュリティ設定を定義します。そのゾーン内のすべてのサイトは、ゾーンのセキュリティ設定に基づいて扱われます。セキュリティ設定には、IE3 で使い慣れた High、Medium、Low の設定値が継承されていますが、いくつかの重要な変更点があります。以前、ユーザーから High、Medium、Low のセキュリティが正確にはなにを意味するのかという質問がよくあり、また、企業からは、これら 3 つの設定値のほかに制御を追加したいという要望がありました。これからは、ユーザーが各設定値の中で有効または無効になっているセキュリティ オプションを正確に把握でき、新たな "カスタム" 設定を選択することによって、50 以上のオプションについて的確に制御できます。

IE4 では、インターネット、イントラネット、信用できる Web サイト、信用できない Web サイトという 4 つのゾーンが事前設定されており、管理者は各組織のニーズに合わせてゾーンを追加・削除できます。

これらのゾーン選択肢すべては多すぎるという企業ユーザーがいた場合は、ユーザーが IE4 をインストールする前に、管理者が Internet Explorer Administration Kit (IEAK) を使用して、これらのセキュリティ オプションをすべて事前に設定することができます。特定のオプシ

ョンを指定外にすることもできます。

さらに重要なことには、ゾーン メンバーシップおよびセキュリティ設定値のアップデートが IE4 スタート時に各ユーザーのデスクトップ マシンに自動的に送られるため、管理者は、ネットワーク上のすべてのマシンにおいて動的にセキュリティ ポリシーを管理できます。

将来のゾーン

現在、ゾーンの作成および保守は、エンド ユーザーまたは管理者に責任があります。しかし、W3C によって進行中の新しい規格作成においては、エキサイティングで便利なゾーンの機能拡張が提示されています。W3C の Digital Signature Initiative (詳細は後述します) では、Web サイト、OCX、JAR など、静的および動的ドキュメントを含む多様なコンテンツに対応するために、明確で自動化および拡張が可能なセマンティックスを標準化する作業に積極的に取り組んでいます。この取り組みで宣言されている目標は、単独の文書または文書の集合体に関する認証を宣言するために、デジタル署名付きのラベルを使用することであり、すなわち、デジタル署名へつながる情況を作ることです。

このラベル情報は、第三者の組織によって収集・検証され、ゾーンの会員情報ソースとして役立つ可能性があります。

まとめ

インターネットのセキュリティに対する真剣なアプローチは、すべて包括的である必要があります。さらには、わかりやすいものでなければなりません。Microsoft の Informed Trust Management Architecture には、コンポーネントのセキュリティを約束するだけでは不十分だという認識があります。問題はそれ以上に大きく、サイト自体の信用から始まっています。信用 (Trust) とは、共通で不可避の基準であり、あるものを信用すべきかどうかは、子細にわたる、けれどもわかりやすい判断に基づくものでなければなりません。

Trust Zone および Authenticode 2.0 は、ユーザーによる "正しい" 信用判断を可能にする 2 つの新しいツールです。ゾーン方式は、ユーザーや管理者にとって、発信元をベースにしたインターネットのセキュリティ ポリシーを設定する便利な手段です。Authenticode 2.0 は、shrink wrap されたデジタル ソフトウェア モデルに、時刻スタンプの添付と Java Capability への署名の機能を追加します。これらを用いることにより、インターネットにおいて実用性があり、拡張可能なセキュリティ機構の基礎が形成されます。

信用管理の背景

Microsoft Informed Trust Management Architecture は、1996 年に AT&T 研究所の Matt Blaze、Joan Feigenbaum、Jack Lacy によって最初に導入された信用管理 (trust management) の概念 [1] と、W3C コンソーシアムの Digital Signature Initiative の共同作業 [2] をベースとしています。

信用管理は、インターネット セキュリティ コミュニティにおいて活発に研究されている分野です。この文書の [4][5][6] で照会するように、このテーマについては、多数の優れた論文が発行され続けています。

W3C Digital Signature Initiative の概要

Web のもつあらゆる可能性を達成するためにも、エンド ユーザーが信用できる Web コンテンツを見極められる確かなメカニズムが必要です。とりわけ、企業は、一般の信用が重要事項となる 2 つの文書クラスがあると考えており、これが、企業間のワーキング グループが結成される十分な理由となりました。

- Active コンテンツ (たとえば、ActiveX コントロール、Java アプレット、プラグイン、アプリ

ケーション マクロなど)。ユーザーは、どの特権を付与すべきか判断するために、サイトに関する十分に信頼できる情報を入手する必要があります。
- 責務を伴う文書 (たとえば、価格表、プレスリリース、政治声明など)。

Web の発行者は、認証を確実に行う手段を必要とし、ユーザーは、それを検証する必要があります。両者のニーズは、オンライン文書にデジタル署名を添付することで解決されます。このデジタル署名は、文書の発行元を識別する役目を果たします。しかし、多くのユーザーにとっては、信用判断の裏付けに必要な情報がほかにもあります。これが典型的に、ユーザーが信用している当事者から承認を得るという形となって登場します。たとえば、1 つのアクティブ コードを使用するという判断が、PC Week の記事によって影響されることもあれば、サイト セキュリティ管理者の承認によってのみこのコードの実行が許可される場合もあります。

市場のパワーに動かされて、様々なソフトウェア ベンダーが第一の問題の一部にだけ対応した初期ソリューションを配備しましたが、これらのソリューションは、ユーザーの Web 信用判断を助けるためのもっと大きなニーズに対しては、完全には満たしてくれません。業界の会議を二度行った結果、W3C は、次のような成果物をもたらす短期集中プロジェクトを開始しました。

- 上記の 2 つの問題点をいずれも解決し、W3C のすべてのメンバーに承認される、フレームワーク、プロトコル、フォーマットの仕様 ("W3C 勧告")
- このフレームワークを中核として業界全体で使用される、実装サンプル
- 以上のすべてを維持・拡大していくための運用プロセス (例、W3C Editorial Review Board、または、IETF や ANSI へ提案)

このプロジェクトの目標は、インターネットの一般的な信用問題に対する相互運用可能なソリューションを作成することによって、ユーザーに明確な利益をもたらすことです。

Digital Signature Initiative に関する情報は、次のサイトにあります。
http://www.w3.org/pub/WWW/Security/Dsig/
現在作成されている DSig 1.0 Signature Labels ドラフトは、次のサイトにあります。
http://www.w3.org/pub/WWW/TR/WD-DSIG-label.html

参考資料

[1] M.Blaze、J Feigenbaum、J Lacy、"Decentralized Trust Management"、Proceedings of the 1996 IEEE Symposium on Security and Privacy または、DIMACS Technical Report に掲載。
[2] Y Chu、J Feigenbaum、B LaMacchia、P Resnick、M Strauss、"Referee: Trust Management for Web Applications"、Proceedings of the 1997 World Wide Web Conference (WWW6) に掲載。

-→ [3] - [6]