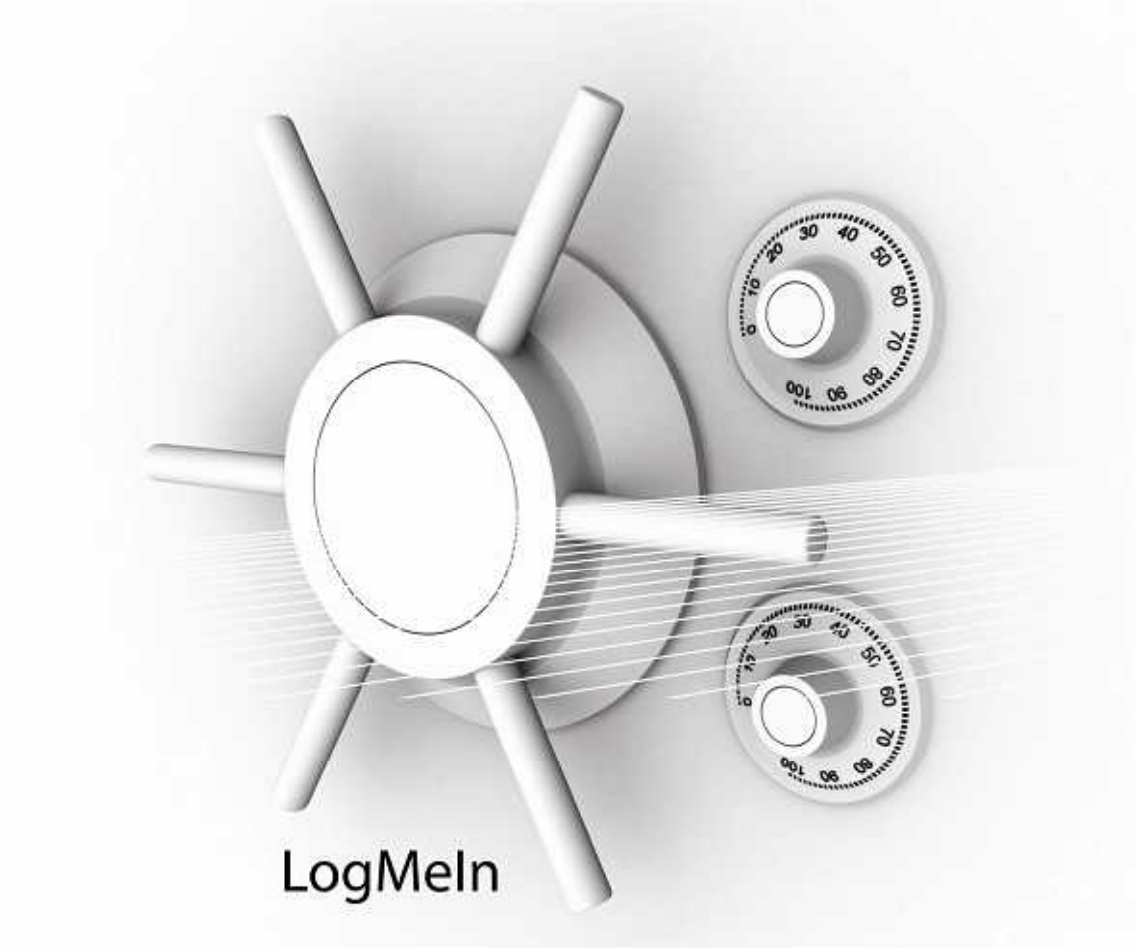

セキュリティ

詳細説明

LogMeIn®

著者

LogMeIn, Inc.のチーフ・テクニカル・オフィサーである Marton Anka がこの文書の主な著者です。

要約

この文書では、LogMeIn.com のリモートアクセス製品である LogMeIn のセキュリティ機能の詳細が述べられています。私共 LogMeIn.com では、明確に説明されていないセキュリティというものを信用していませんし、私共のお客様が、例えば終端間暗号化 (end-to-end 暗号化) のような重要なセキュリティ上の機能に対する私共の言い分を、盲目的に信用するというようなことも望んでおりません。私共の製品において、セキュリティメカニズムがどのように働き、相互に作用しているのかという詳細を公開することにより、皆様に私共の努力を評価していただければと思います。

読者

この文書は技術文書であり、ネットワークエンジニアやネットワーク設計者の方々向けに書かれています。このホワイトペーパーを読み、ここで新たに得られた情報を各自のネットワークに関する既存の知識と組み合わせることにより、私共の製品を利用する前に、必要なリスク分析を行うことができるでしょう。

用語

LogMeIn アーキテクチャでは、リモートアクセスのセッションに関わるエンティティが 3 つあります。「クライアント」または「ユーザ」とは、リモートリソースにアクセスする人、またはソフトウェアのことです。「ホスト」または「サーバ」とは、アクセスされるコンピュータ、またはこのコンピュータ上の LogMeIn ホストソフトウェアのことです。「ゲートウェイ」とは、クライアントとホスト間のトラフィックを仲介する LogMeIn のサービスのことです。

設計原理

LogMeIn は、信頼できないネットワーク上の重要なリソースに対して、安全なリモートアクセスを可能にするように設計されています。このソフトウェアの開発中、セキュリティに対する配慮は常にユーザビリティより優先されました。以下に、重要なものから順番にセキュリティ設計目標をあげていますが、意思決定プロセス(decision-making process)においては、これが指針となりました。

- セキュリティ、および攻撃の軽減
- 接続サーバに対するユーザの認証と利用許可
- ユーザに対する接続サーバ認証
- データの機密性
- 接続サーバ内におけるユーザの認証と利用許可

リモートアクセス原理

すべてがターゲットである

ブロードバンドのインターネット接続が浸透していくにつれて、ますます多くのコンピュータが年中無休でオンライン状態にあるようになりました。こういったコンピュータのほとんどはホームユーザによって操作されているコンピュータで、そこにはパッチを当てられていない脆弱性や、パスワードが適切でないといったセキュリティホールがぽっかりと口をあけています。

しかし、最大の弱点はユーザ自身なのです。インターネット利用者のセキュリティ対策の未熟さをついた攻撃の例として、電子メールによるウイルスの急速な感染ほどわかりやすいものはありません。このメールに添付されたファイル、これは「トロイの木馬」として有名ですが、信用できないコンテンツをインストールしてはならないという最重要ルールを、ユーザがいとも簡単に破ってしまったために、こんなに急速に広まってしまったのです。ユーザ自身の責任で自分のコンピュータをトロイの木馬に感染させてしまったのだとすれば、そのようなユーザがじかに攻撃された時、自分のシステムをきちんと守ることなど期待できるでしょうか？

優秀なネットワーク管理者ですら、パッチの 1 つや 2 つ、うっかりインストールし忘れることがあり得るのです。最悪のシナリオを想定した場合、このミスのせいで、攻撃者の任意のコードがシステム上で実行されてしまうかもしれません。これを立証するものとして、2003 年に Microsoft SQL サーバを狙ったワームが急速に広まったことほどわかりやすい例はないでしょう。MSBlaster と Slammer の両方が大量のコンピュータを汚染し、汚染されたホストが急激に増えたことも手伝って、過度のネットワークトラフィックが発生しました。

そのためユーザは、汚染されていないネットワークにおいてさえ、インターネットアクセスの低速化を確認することができたほどです。

MSBlaster や Slammer といったワームは雑に作られており、広がっていくスピードは決して速くはなく、データ損失やデータの盗難を引き起こすものでもありませんでした。このワームを作った人は、広く知られた Microsoft SQL サーバの脆弱性を狙ったのです。修正プログラムは、最初にワーム攻撃が行われた時より何週間も前に入手可能になっていました。つまり、これは実に悪趣味ないたずらなのです。たくさん問題を引き起こしたことは間違いありませんが、その影響は破滅的なものにはなりませんでした。（しかし、もしかしたら本当に破滅的なことになっていたかもしれません）

悪意を持った本当のハッカー達がお金になるターゲットを見つけた時、一体どんなことをしでかすのか、私たちはただ想像するしかありません。

信用できないリソースは武器になる

インターネット上にあるほとんどのシステムは、真っ先にハッカーによる攻撃対象となります。攻撃の理由は経済的利益である場合がほとんどですが、純粋に悪事を働くことを楽しむために行われることもあります。

2003 年の秋、Valve Software では一台のコンピュータで情報漏洩が起きました。[WRD20031004]ハッカーは最初に 1 つのシステムにアクセスしました。伝えられるところによると、パッチを当てられていない Microsoft Outlook の脆弱性を利用したようです。それからキーロガーがインストールされ、該当コンピュータ上で入力されたネットワークのパスワードが盗まれました。それからは簡単です。Valve 社のコンピュータゲーム、Half Life 2 は開発の最終段階にあり、発売が待たれていましたが、そのソースコードが盗まれてしまったのです。

2003 年 2 月、名前は公表されていませんが、あるインターネットサイト（おそらく一番可能性が高いのは、クレジットカード処理業者でしょう）から、何百万ものクレジットカード番号が盗まれました。[CNN20030218]その結果、何百万ドルに相当する不正取引が生じた恐れがあります。関係者は当然のことながら固く口を閉ざしており、詳細についてはわかっていません。

リモートアクセスとセキュリティ

インターネットに接続されているコンピュータは、たとえリモートアクセス製品をインストールしていなくても、非常に攻撃を受けやすいということは明らかです。リモートアク

セス製品はリスクを高める要因になると思われていますが、それは主に心理的な理由によるものです。ユーザが最初にリモートアクセス・ソリューションが動いているのを見た時、最初にネガティブな反応をするのは、たいていセキュリティのことを心配しているからです。これはいたって普通のことですし、実際、私共はそうあって欲しいとすら思っています。本当に問題なのは、ユーザがほかのネットワークを介して稼動するアプリケーション、例えば電子メールクライアントや web サーバ、または OS 自体がさらされている脅威にすぐに気がつかないということなのです。

多くの OS は、何らかの種類のリモートアクセス・ソリューションをデフォルトで備えています。例えば Windows 2000 Server および Windows 2003 Server には、単純なリモート管理インタフェースである Microsoft Remote Desktop が出荷時に備えられています。現在入手可能な OS の中で、最も安全だと見なされている Unix の変形である OpenBSD でさえ、SSH を備えています。これもまた、ネットワーク接続を介して、コマンドラインからリモートコンピュータへのアクセスを可能にする単純で安全なアプリケーションなのです。

基本的に、正しく選ばれ、正しく構成されたリモートアクセス・ソリューションは、リスクを増やすようなことはありません。RemotelyAnywhere や LogMeIn など、信頼のおけるリモートアクセス・ソフトウェアパッケージを使いつつ、ネットワーク管理者がネットワークを安全に保つことができれば、ネットワークのセキュリティの影響を受けることなく生産性を向上させ、コストを減らすことができるでしょう。

LogMeIn のアーキテクチャ

LogMeIn によって採用されているセキュリティの正確なメカニズムを説明する前に、LogMeIn によって設計されたネットワークアーキテクチャについて述べます。

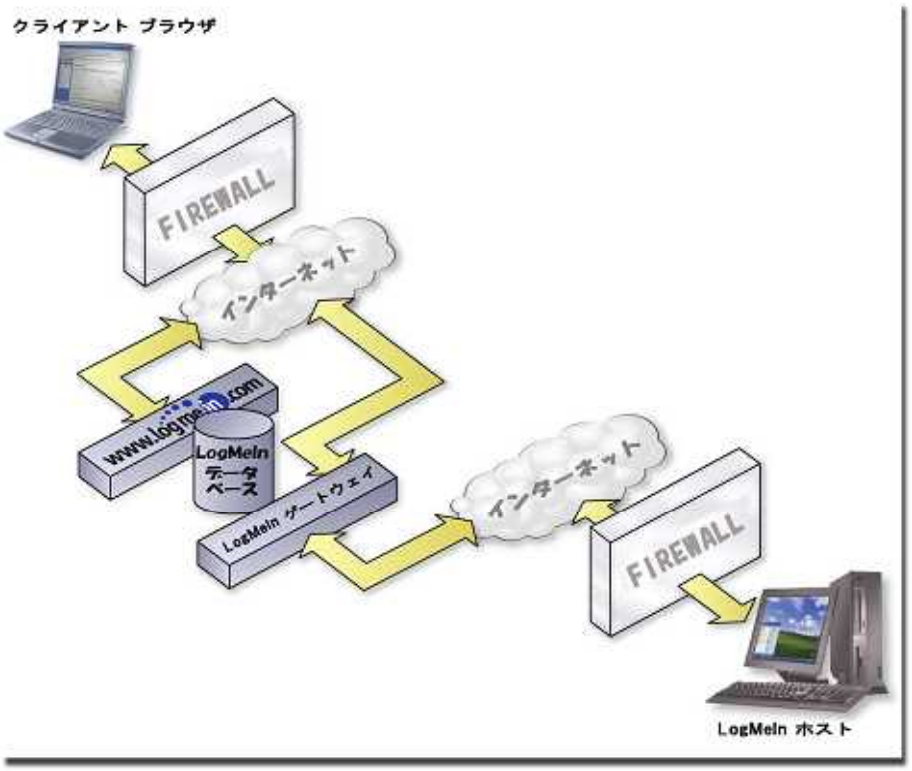

図 1：LogMeIn アーキテクチャ

すべてのリモートアクセス・セッションには、3 つのキー・コンポーネントがあります。クライアントとホストの役割はわかりやすいと思います。3 つ目のコンポーネントは LogMeIn **ゲートウェイです。**

上の図の中の LogMeIn **ホストは、物理的に安全な私共のデータセンターにある** 1 **つの** LogMeIn **ゲートウェイサーバと** SSL **で安全に接続されています。この接続はホストによって開始され外部への接続扱いになるので、ファイアウォールは安全な** web **ブラウザのトラフィックと同様に扱います。**

クライアントのブラウザは www.logmein.com **への接続を確立し、認証を得ます。そうしてはじめてゲートウェイは、その後の暗号化されたトラフィックをクライアントとホスト間でやり取りするようになります。クライアントが、さらにホストに対して認証を必要とするという点に注目して欲しいと思います。ゲートウェイは** 2 **つのエンティティ間のトラフィックを仲介しますが、ホストがクライアントを何の根拠もなく信頼することを求めません。**

クライアントとホスト間のダイレクトな接続を確立する代わりに、ゲートウェイを利用す

ることの利点は明白です。クライアントまたはホストのいずれか、または両方、または後者のエンティティをファイアウォールで保護することができるようになるからです。LogMeIn ではゲートウェイを活用することによって、ファイアウォール設定に関してユーザが心配しなくて済むようにします。

LogMeIn のセキュリティメカニズム

ユーザがインターネットデータのセキュリティについて考える時、一般的にはデータの暗号化のことを気にかけると思います。ただしそれは、使用される暗号キーの長さによってセキュリティを評価する程度のレベルにすぎません。暗号化と復号化は、どちらも非常に重要ではありますが、システムを安全に設計しようとする設計者が直面するほかの問題に比べれば、小さなタスクにすぎないのです。ご承知の通り、データの暗号化は LogMeIn の設計者が掲げた 5 つの大きな目標のうちの 1 つにすぎません。

ユーザに対する接続サーバ認証

何よりもまず、ユーザが LogMeIn をインストールした「サーバ」に接続する時、データをやり取りしようとしている相手のコンピュータが、本当に自分が接続したいと思っているコンピュータであるという確信を 100%持っている必要があります。

攻撃者が、ユーザに対してサーバのフリをしている、または、サーバに対してユーザのフリをしていると仮定してみましょう。この場合、攻撃者は両者の間にいて、通信されるデータを読んだり、そしておそらくは改ざんしたりすることができます。これは「Man In The Middle（中間者）」、すなわち MITM（中間者）攻撃として知られており、この攻撃に対処するのは特に難しいとされています。

LogMeIn は SSL/TLS 証明を活用してサーバの身元を検証し、それによって MITM 攻撃を防ぎます。接続が試みられる時、サーバの証明書が検証されます。この証明書が、ユーザが信頼できるとして認めた認証局（certifying authority）が発行したものでない場合には、警告が提示されます。証明書が、信頼できる認証局から発行されたものであっても、URL 中のホスト名が証明書に記載されているホスト名と一致しない場合には、別の警告が提示されます。

サーバがこれらの検証にパスしてはじめて、ユーザのブラウザは「Pre-Master Secret」すなわち PMS を生成し、証明書に含まれているサーバの公開鍵で暗号化した上で、それをサーバに送ります。公開鍵暗号を使うことによって守られているので、対応する秘密鍵を持っているサーバのみが PMS を解読することができます。その後、ユーザとサーバの両方が Master Secret を導き出すのに PMS が使われ、今度はそれが、安全なセッションの間中、初

期化ベクタとセッションキーを作るために使われます。

つまり上記の処理によって、ユーザが不正な第 3 者ではなく、正しいサーバとの接続を確立しているということが保証されます。万一 MITM 攻撃が試みられた場合でも、セキュリティ警告がトリガーされるか、もしくは PMS が MITM（攻撃者）にはわからないため、効果的に攻撃を防ぐことができます。

この件に関する詳細は、SSL を参照してください。

ゲートウェイに対するユーザ認証

クライアントは、ゲートウェイとホストの両方から承認されなければなりません。クライアントが LogMeIn ウェブサイトにログオンすると、簡単な電子メールアドレスとパスワードの承認が実行されます。ぜひ、LogMeIn が提供している複数のセキュリティ追加オプションを有効にしておくよう、お勧めします。

追加オプションの 1 つとして、印刷したワンタイム・パスワード（OTP）のシートがあります。この OTP オプションを有効にすると、ユーザはゲートウェイによって生成されたランダムな 9 文字のパスワードの一覧をプリントアウトするように要求されます。これ以降 LogMeIn.com サイトにログインする時は、印刷されたシート上にあるパスワードのうち、まだ使われていないものを 1 つ入力するよう要求されます。ユーザは、OTP が足りなくなる前にもう一枚シートを追加で印刷するよう要求され、その時点で前のシートにあったパスワードのうち、使われなかったものは無効になります。

もう 1 つ、もっと簡単な方法は、LogMeIn の Wireless Password テクノロジを使うことです。このオプションが有効になっていると、ユーザはワイヤレス端末で利用する電子メールアドレスを入力することを要求されます。このアドレス宛に送られるメッセージの中には、短期間のみ有効な一回限りのパスワードが含まれています。この技術を利用するためには、電子メールを受信するデバイスが携帯電話やポケットベルのような、ワイヤレスのデバイスでなければならなりません。唯一要求されるのは、このデバイスへの電子メール受信がほとんどリアルタイムでなければならないということです。この機能がオンになっていて、LogMeIn ゲートウェイに対するユーザの電子メールアドレスとパスワードが承認されると、パスワードが生成され、ワイヤレスの電子メールアドレスに送信されます。ユーザはこの電子メールを受け取り、中に含まれているコードをゲートウェイが提示するフォームに入力しなければなりません。そのため、ユーザは該当デバイスを携帯していることが前提になります。このパスワードは、生成されてから数分後、または使用された時のどちらか早い方の時点で無効になってしまいます。

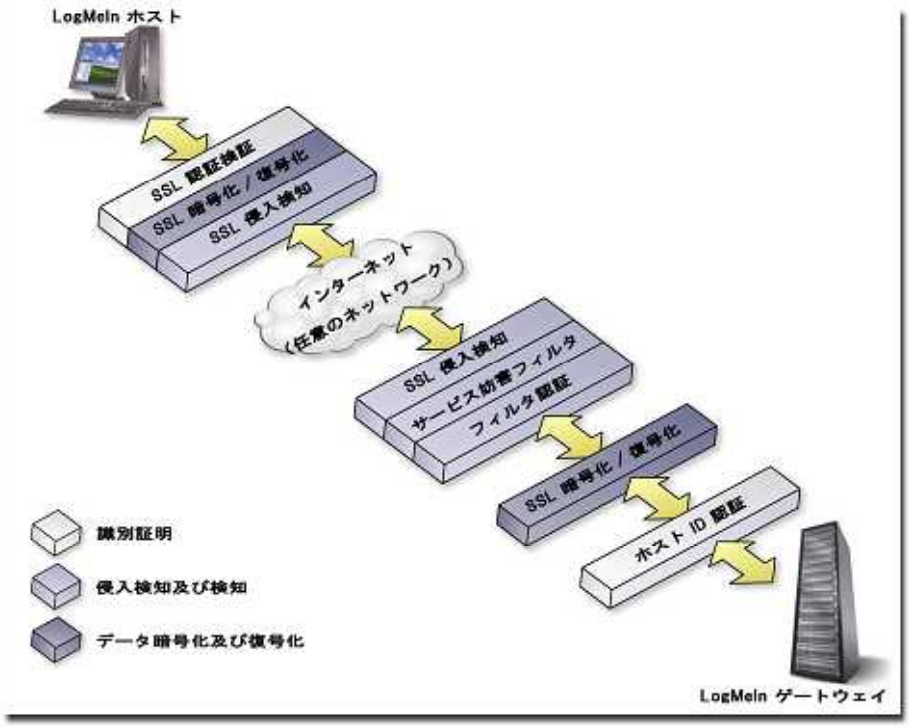

図 2：ユーザとゲートウェイ間の認証

ホストに対するゲートウェイ認証

ゲートウェイは、アクセスコードによって承認を得る前に、ホストに対して自分の身元を証明する必要があります。ホストは、ゲートウェイへの接続を試みる際に SSL 証明書をチェックし、確かに LogMeIn サーバの 1 つに接続しているということを確認します。このプロセスの詳細は、上記で説明されたステップと非常によく似ています。

ゲートウェイに対するホスト認証

ゲートウェイは接続を受け付けると、ホストの身元を検証します。検証には、ホストが最初の接続を行った時にゲートウェイによって発行された、2 つのエンティティ間で秘かに共有している長い固有の識別文字列が使われます。この固有の識別文字列は、ホストがゲートウェイの身元を検証してはじめて、SSL で守られたチャネルのみを介して通信されます。

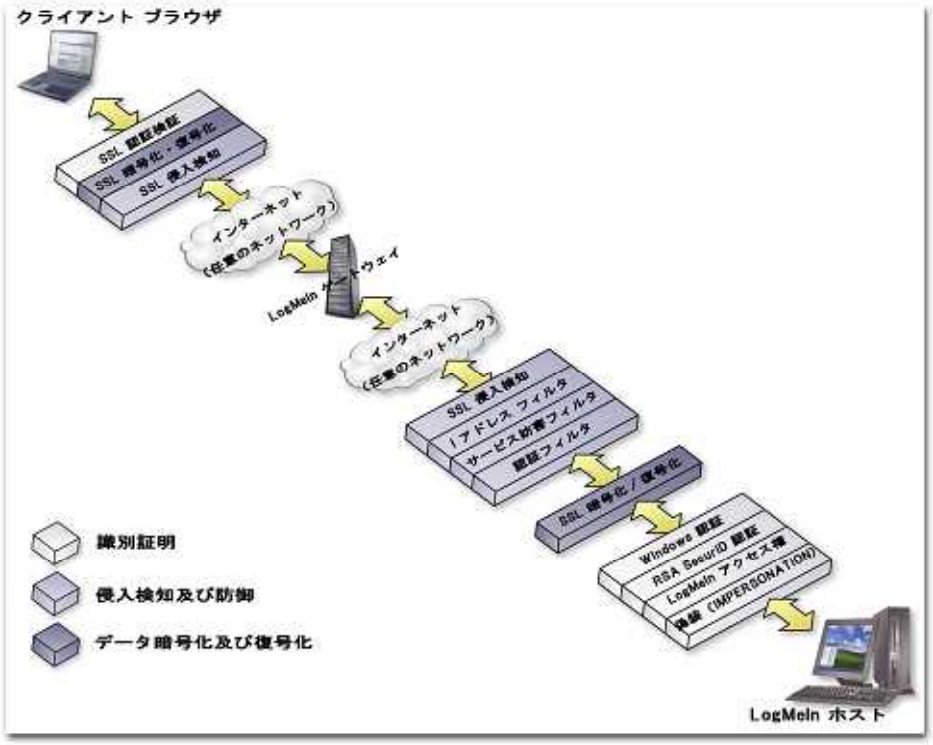

図 3：ホストとゲートウェイの認証

上の図は、ホストがユーザにアクセス可能になる前に、ホストとゲートウェイがどのようにお互いを認証するかを示しています。

データの暗号化

SSL/TLS **標準では、**RC4 **や** 3DES **など、幅広い暗号スイートが定義されており、**AES **をも含んだより進化したスイートを提供している実装もあります。**RC4 **は** 128 **ビットの鍵で動作し、**3DES **は** 168 **ビットの鍵を使います。**AES **は** 128 **または** 256 **ビットの鍵を使うことができます。おそらくユーザとサーバは、可能な限り最も強力な暗号を使うことで合意すると思いますが、これは、ユーザが使いたい暗号の一覧をサーバに送り、サーバが一覧の中から好きなものを** 1 **つ選ぶことによって可能になります。**SSL/TLS **標準では、サーバがどのようにして最終的な暗号を選ぶべきか、ということまでは定義していません。**LogMeIn **では、サーバは単純に、クライアントが提示した暗号のうちで最も強力な暗号スイートを選びます。**

このメソッドによって、クライアントとサーバの両方が、特定のデータ暗号化アルゴリズムに依存しないことになります。ということは、アルゴリズムが破られているとか安全でないという調査結果が出るようなことがあったとしても、双方のコンポーネントをアップデートする必要もないということになります。

侵入探知

LogMeIn では、侵入の試みを探知する 2 つの段階が提供されています。
一つ目の段階は SSL/TLS によって提供されており、データが伝送中に変えられていないことを保証するものです。

これは様々な技術を使うことによって可能になっています。

Record Sequence Numbering（レコードの通し番号）とは、SSL/TLS レコードが送信者によって番号をつけられ、その順番が受信者によってチェックされるという仕組みです。これにより、攻撃者が任意のレコードをデータストリームの中に挿入したり、削除したりすることができなくなります。Message Authentication Codes (MACs)がすべての SSL/TLS レコードに付加されます。これは（通信している 2 者間でのみ知られている）セッションキーと、レコードに含まれるデータから生成されます。もし MAC 照合が失敗した場合は、データが通信中に改ざんされた疑いがあります。LogMeIn が良く使う暗号スイートとして、Cipher Block Chaining (CBC モード)が活用されています。これは、すべての SSL/TLS レコードが前のレコードの内容に依存するというものです。このモードでは、現暗号文は現平文ブロックだけでなく前の暗号文ブロックも用いて計算されます。これによってさらに、パケットがデータストリームから削除されたり、挿入されたりしていないことが確実になります。
SSL/TLS 侵入探知についての詳細は、SSL を参照してください。

2 つ目の段階は LogMeIn 自身によって提供されており、3 つのフィルタから成っています。

1. IP アドレスフィルタ
   LogMeIn がユーザからの接続リクエストを受け取ると、まず信頼できる IP アドレスと信頼できない IP アドレスのリストをチェックし、場合によっては接続を拒否します。管理者は、良いとわかっている IP アドレスと、悪いとわかっている IP アドレスのリストを LogMeIn 内に作ることができます。例えば、社内ネットワークや別の管理者のホーム IP アドレスを良いアドレスとして指定することができます。

2. DoS フィルタ
   リクエスト送信元の IP アドレスが、認証なしに大量のリクエストを監視期間中に送った場合、DoS フィルタが接続を拒否します。
   これは例えば、ログインページへの自動的かつ頻繁なアクセスに対して(ブルートフォース攻撃)、ホストコンピュータに負荷をかけすぎるのを防ぐために行われます。

3. **認証フィルタ**

   ユーザのログインの試みがあまりに何度も失敗すると、認証フィルタは接続を拒否します。認証フィルタは、アカウント名とパスワードを推測することによって侵入しようとする攻撃者を防ぎます。

接続サーバに対するユーザの認証と利用許可

前の段階でユーザがアクセスを認められると、今度はユーザ自身の身元をサーバに認めてもらう番です。これは Windows 必須の認証ステップによって行われます。

Windows **認証（詳細情報については**[WINAUTH]**を参照）では、ユーザは通常の** Windows **のユーザ名とパスワードを使い、サーバへのアクセスを承認されることが求められます。サーバは通常、このリクエストを関連ドメインのコントローラに伝えます。このステップによって、ユーザの身元が検証できるだけでなく、ある特定のサーバに誰が、いつログインできるのかをネットワーク管理者が管理できるようになります。**

Windows **によって要求されるこの単純なユーザ名／パスワード認証にさらなるセキュリティレイヤーを追加するために、システム管理者は、**LogMeIn **でも** RSA SecurID **認証が要求されるように設定することができます。**SecurID **は** 2 **つの要素からなる認証方法で、ユーザは、自分と** ACE **認証サーバだけが共有する秘密情報を提示し、かつ「オーセンティケータ」と呼ばれる独自のデバイスを保持していることを証明しなければなりません。**

LogMeIn **によって使われている技術の先駆けとなった** RemotelyAnywhere **は、**2003 **年に公式に** RSA Security **によって**"SecurID Ready"**であると認定されました。それ以来、**LogMeIn **は** RSA **テクノロジに準拠した高いレベルのセキュリティを保ち続けています。**RSA SecurID **製品についての詳細は、**[RSASECURID]**を参照してください。**

LogMeIn **ホストにおけるもう** 1 **つのセキュリティオプションは、**Personal Password **です。ユーザはホストに対して** Personal Password **を割り当てることができます。このパスワードは** Windows **のパスワード同様、ゲートウェイによって保存されたり検証されたりしません。しかし、**Windows **のパスワードと** Personal Password **の興味深い違いは、ホストは決して全ての** Personal Password **をいっぺんには要求しないため、単独の認証セッションにおいては、ユーザによって** Personal Password **全てが完全な形で入力されることは決してないということです。ユーザは通常、**Windows **の認証が成功したあと、この** Personal Password **のランダムな** 3 **つの数字を入力するようホストから要求されます。ユーザが正しい数字（例えば、**1 **番目、**4 **番目、**7 **番目）を入力すると、アクセスが認められます。**

ホスト内におけるユーザの認証と利用許可

LogMeIn が上記のメソッドを使ってユーザの身元を確認すると、今度は内部のユーザデータベースをチェックして、そのユーザがどの内部モジュールについてアクセス権を持っているかを調べます。

システム管理者は LogMeIn を設定することによって、特定の役割を持つユーザが、LogMeIn によって提供されているツールの一部だけにアクセスできるようにすることができます。例えば、ヘルプデスク部門はコンピュータの画面と性能データを見ることだけはできるけれども、実際にマウスとキーボードを使ったり、システム設定を変えたりすることはできないように設定することもできます。また反対に営業部門では、自分たちのコンピュータに対するリモートコントロールアクセス権はあるけれども、性能監視やリモート管理といった機能を使うことはできない、という設定にすることもできます。

LogMeIn は、ユーザが認証された時に取得した Windows のアクセストークンを使い、ユーザの代わりに OS に対するアクションを代行します。これによって、LogMeIn は Windows のセキュリティモデルに忠実に従う一方、ユーザは自分のコンピュータの前に座っている時と同じファイルやネットワークのリソースにアクセスすることが可能になります。ユーザが Windows において利用できないリソースは、LogMeIn においても利用できません。

監視とロギング

LogMeIn では、豊富なロギング機能が提供されています。ソフトウェア内で起こったイベントの詳細なログがインストール・ディレクトリに保管されています。最も重要なイベントは、Windows の NT アプリケーション・イベントログにも保管されています。その中には、例えばログオンやログオフのアクションが含まれています。また、あらかじめ決められた ODBC リンクを使って、詳細なログを中央の SYSLOG サーバやリレーショナル・データベースへ送ることもできます。

データ転送

ゲートウェイはホストとクライアント間で暗号化されたデータをやり取りすることで、終端間暗号化を実現しています。もし読者が、SSL がどのように機能するかということをよく知っているなら、これは一見不可能なことのように思われるかもしれません。「結局、クライアントがゲートウェイと通信していることが確実である以上、クライアントから送られたデータを解読できるのはゲートウェイしかいない、という論理だろう？」と。確かにその通りではあるのですが、LogMeIn は、ホストとゲートウェイ間の SSL セッション処理方法にさらに重要な変更を加えたのです。実質的には、SSL ネゴシエーションの最初の部分は

確かにゲートウェイとクライアントの間で実行されます。が、そのあとゲートウェイはやり取りされる情報をホストに伝え、ホストは SSL セッションを再ネゴシエートし、新しいセッションキーについてクライアントと同意します。こうして本当の意味での終端間（end-to-end）セッションを提供しているのです。

クライアントがゲートウェイの証明書を認証したという事実 ー Pre-Master Secret を生成するのに使われる情報を暗号化するのに、実際に RSA 公開鍵を使ったという事実、さらに、ホストもゲートウェイの証明書について同じことを行ったという事実により、man-in-the-middle（中間者）攻撃は事実上不可能になります。

UDP NAT トラバーサル

UDP NAT トラバーサルが全体像の中でどのような位置にあるかを読者に説明するのは重要なことだと思われます。なにしろ UDP 自体が安全性の低いことで悪名高いのですから。これは全くの誤解というわけでもありません。UDP が通信メディアとして使用されている場合には、セキュリティ面では非常に深刻な問題を持つことになります。UDP データグラムは偽造しやすく、送信者の IP アドレスは簡単に成りすましが可能だからです。

UDP NAT トラバーサル接続でこのような問題が起こらないようにするために、LogMeIn.com は UDP を通信手段そのものとしては使いません。UDP は、ISO/OSI Network Model で定義されているようにネットワーク層に分類され、フロー制御、動的帯域の拡張、パケットの通し番号機能を備えた、TCP のようなトランスポート層がその上にのります。LogMeIn.com が IP パケットの代わりに UDP を使うこと（そのため TCP のようなトランスポート層を効果的に再実装すること）に決めた理由は、ほとんどのファイアウォールと NAT デバイスは、それがセキュリティの防御の範囲内から開始されている限り、UDP トランスポートを通してシームレスな双方向通信を可能にするからですが、それには TCP と IP パケットの重要な再設定が必要になります。

UDPパケットを使用し、TCPのような信頼性のあるストリームを実現させた後、さらにそのストリームをSSL層によって保護することにより、完全な暗号化、完全性保護、エンドポイント検証を可能にします。

UDP NAT トラバーサル接続を確立するには、クライアントとホストの両方が暗号化された複数の UDP パケットをゲートウェイに送ります。これらのパケットはゲートウェイとその相手によって共有された秘密情報を使って暗号化されていて、既存の SSL 接続を通して通信されます。したがって、成りすましは不可能です。ゲートウェイはこれらのパケットを使って、2 つのエンティティの外部の（インターネット）IP アドレスを決定します。また、

新しい UDP パケットが送られた時に、どのファイアウォールポートが通信に使われるか予測しようとします。ゲートウェイがその情報を相手に伝えると、相手はダイレクトな接続を確立しようとします。ゲートウェイがポートの順番を正しく予測していていれば、接続は成功します。そして相手は、ゲートウェイから取得したもう 1 つ別の共有の秘密情報を使ってお互いを確認し、SSL セッションを確立したあと、ダイレクトに通信します。

ダイレクトな接続を確立できない場合は、相手は再び TCP を介してゲートウェイに接続し、転送された終端間暗号化済みセッションが使われることをリクエストします。

このプロセス全体はわずか数秒しかかからず、ユーザに気づかれることもありません。ダイレクトな接続が使われている場合に唯一変わる点といえば、パフォーマンスが向上することと、待ち時間が減ることです。

ソフトウェアのアップデートとゲートウェイセキュリティ

LogMeIn は半自動的、または自動的にユーザのコンピュータ上でアップデートされます。ホストソフトウェアは定期的に LogMeIn.com ウェブサイトをチェックし、ソフトウェアの新しいバージョンがないかどうかを確認します。新しいバージョンがあれば、自動的にダウンロードされます（最大でも利用可能な帯域幅の 50%のみをダウンロードのプロセスに使うようにし、ネットワーキング・アプリケーションを邪魔しないように配慮します）。ユーザが実際にコンピュータの前に座っている場合には、メッセージが表示され、ユーザはアップデートの実行許可を与えることができます。

こういったソフトウェアのアップデートは LogMeIn.com によってデジタル署名されます。署名の際には、インターネットに接続されていない別システム上にて、安全に保管されている証明書を使って行っています。したがって、たとえ弊社のデータセンターのサーバが攻撃者によって乗っ取られたとしても、攻撃者によって仕組まれた不正なアップデートが取り込まれてユーザのコンピュータ上で任意のコードを実行できるようにすることはできません。このような攻撃はほとんどあり得ないことではありますが、もしそれが成功したとしても、せいぜいユーザのコンピュータ上で LogMeIn のログオン画面にアクセスできるというところまでです。もし巧みにゲートウェイのセキュリティメカニズムをすり抜けたとしても、Windows のユーザ名とパスワードの組み合わせを正しく入力しなければ、コンピュータへのアクセスは得られません。パスワードのブルートフォース攻撃は不可能です。つまり、何度かパスワードをミスタイプすると、デフォルトで認証フィルタが作動するようになっているからです。ユーザが LogMeIn ゲートウェイのパスワードと同じパスワードを Windows コンピュータに使っている場合、私共は実際の LogMeIn パスワードを弊社のデータベースに保存しません。その代わりに、現在のコンピュータ技術ではパスワードの推

測が不可能なレベルのハッシュ関数とソルト値を使っています。

結論

きちんと設計されたリモートアクセス・ソリューションは生産性を飛躍的に向上させるため、すぐに投資を回収できるでしょう。LogMeIn がきちんと正しく配備され、セキュリティ機能のオプションが活用された時、その利益はリスクをはるかに上回ります。

参考文献

SSL
Eric A. Rescorla: SSL and TLS: Designing and Building Secure Systems
Addison-Wesley Pub Co, October 13, 2000
ISBN: 0201615983

SSH
D. J. Barrett, SSH, The Secure Shell:
R. Silverman: The Definitive Guide
O'Reilly & Associates, February 15, 2001
ISBN: 0596000111

WINAUTH
Windows 2000 Security Technical Overview
http://www.microsoft.com/technet/prodtechnol/windows2000serv/deploy/confeat/sectech.mspx

RSASECURID
RSA Security's SecurID Product
http://www.rsa.com/node.aspx?id=1156

CNN20030218
Hacker accesses 5.6 million credit cards.
http://www.cnn.com/2003/TECH/02/17/creditcard.hack/

WRD20031004 Game Biz Mystified by Code Theft
http://www.wired.com/news/games/0,2101,60701,00.html

Product Information: info@LogMeIn.com
Sales Inquiries: sales@LogMeIn.com
(800) 993-1790
Press: press@LogMeIn.com
Partner Information: partners@logmein.com

LogMeIn®
500 Unicorn Park Drive, Suite 103 Woburn, MA, MA 01801