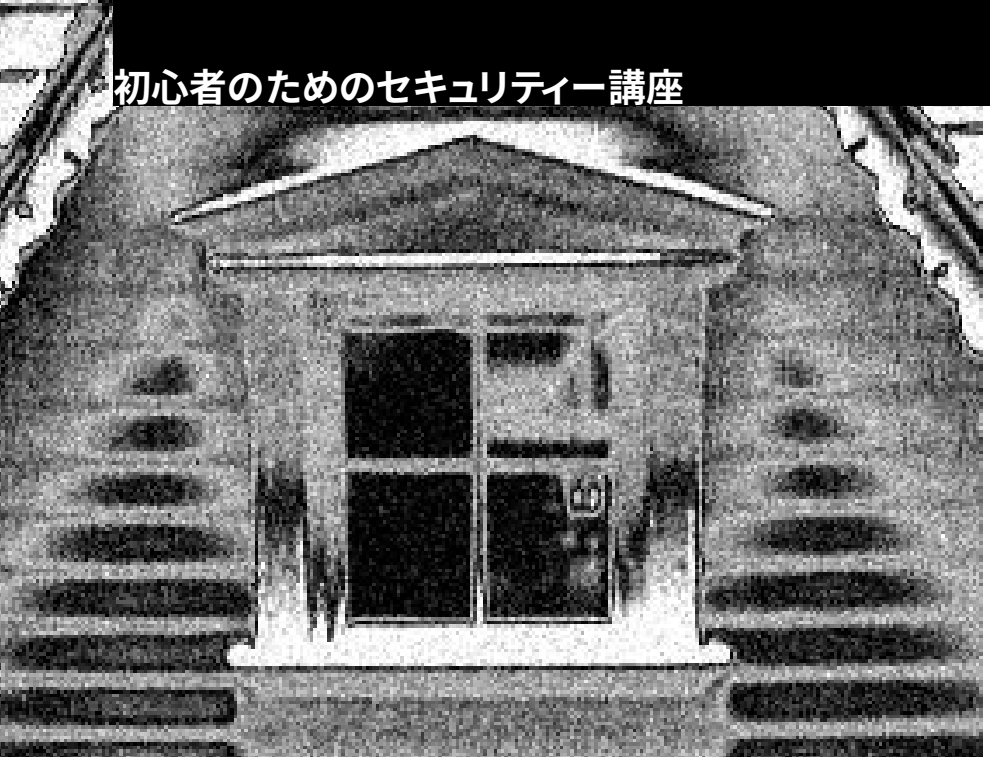
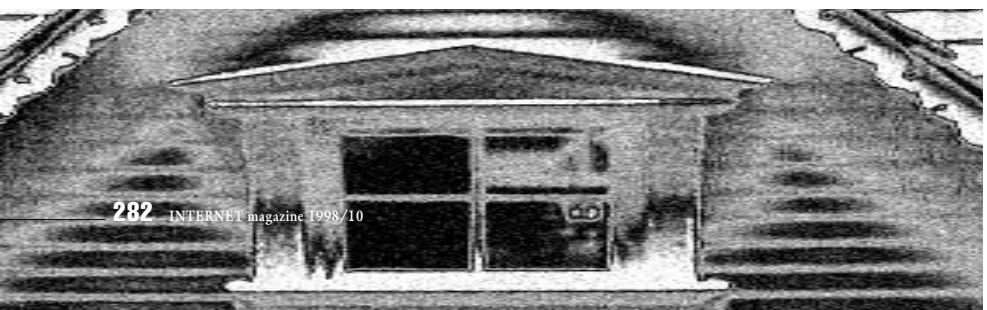


インターネットでの不正行為 その傾向と対策

先月からこの連載ではSOHO環境のセキュリティーについて解説しています。前回はSOHO環境におけるセキュリティーの基本的な考え方について解説しました。今回は具体的なSOHOモデルを例にとりながら、実際の設定について解説します。設定についてのセキュリティーポリシーを知ることで、どんな状況にでも応用できます。今回紹介する内容はあくまでも一例ですので、自分の環境に合わせて考えてみましょう。

第13回 SOHO環境のネットワークセキュリティー その2

JPCERT/CC（コンピュータ緊急対応センター）
URL http://www.jpcert.or.jp/



セキュリティー対策に一歩踏み出そう

前回は、SOHO環境ではシンプルなポリシーが適用できるという所で終わりました。今回は具体的なSOHO環境でのセキュリティーポリシーを考えてみましょう。

1つ1つのサイトで利用しているユーザーの目的もいろいろと違うでしょうし、もちろんスキルにもいろいろなレベルがあると思います。ここでは、ベーシックな利用モデルを考えてみます。あくまでも一例ですので、自分の管理するサイトに対しては、自分自身で考えることが求められます。

何かをまねるのも1つの手ではありますが、基本的な部分の理解がないまま設定だけをまねていては、正しくセキュリティーポリシーを守ることができているかの評価は難しいでしょうし、また、どうしてもセキュリティーホールができやすくなってしまうことでしょう。

このようなことを言うと、多くの読者の方がためらってしまいそうです。しかし、インターネットに接続された莫大な数のサイトを自動的にスキャンするような不正アクセスの手法が行われている現在では、まったくセキュリティーについて考慮していないサイトは許されない時代になりつつあります。まねであれ、何であれ、まずはセキュリティー対策を施すことに一歩踏み出す必要があります。

SOHO環境を使うユーザーについて考えてみよう

ネットワークに接続されたSOHO環境をどのようなユーザーが何のために使っているかを明確にしましょう。

❶ユーザーが誰か、何の目的でマシンを使っているかを明確にできるか
❷ユーザーが外部のネットワークにアクセスする目的は何か
❸ユーザーがネットワーク内部のほかのマシンにアクセスする目的は何か

まず最初の「ユーザーが誰かを明確にできるか」というのは、いったい誰が何のためにサイト内のマシンを使っているかを明確にすることです。誰が何のためにマシンを使っているのかが不明では、防御する対象が明確にできません。「事務担当Aさんが事務処理のために使っている」とか「デザイナーBさんがDTPデザインのために使っている」といった内容でもかまいませんし、あるいは「家族が共有して、電子メールやWWWブラウザー、ワープロなどに使っている」でも構いません。

次に、「ユーザーが外部のネットワークにアクセスする目的は何か」ということを明確にしましょう。たとえば「IRCを楽しむ」とか「特に目的はないがウェブページの閲覧を楽しむ」といったものでもいいですし、「外部から必要なソフトウェアやデータを入手するためにFTPを使う」とか「業務に必要な情報収集のため特定のウェブページを常に参照する必要がある」というのもあるでしょう。

最後に「ユーザーがネットワーク内部のほかのマシンにアクセスする目的は何か」を明確にしましょう。主にユーザーの使うクライアントマシンから内部に用意されたサーバーへのアクセスになると思います。「内部にあるウェブサーバーに新しく用意したHTMLファイルを置く」とか「メールサーバーに配送されているメールを取り出す」などがあるでしょう。

さて、ここでモデルケースと図1のようなものを考えてみます。このサイトでは、外部のネットワークからユーザーの使用しているコンピュータに直接アクセスする必要がないので、外部からのアクセスはサーバー以外へは一切許さないこととします。

サーバーの構成を考えてみよう

外部とのサービスのやりとりを集中して処理するために、ネットワーク内に専用サーバーを用意します。これは、セキュリティーの問題を局所化する戦略でもあります。後に説明しますが、ルーターの機能を活かして外部からは唯一このサーバーのみにアクセスできるようにします。

外部からのアクセスをこのサーバーのみに許すようなセキュリティーポリシーにすることによって、ほかのユーザーのマシンの防御をより確実なものにします。このような役目のサーバーを要塞ホストとも呼びます。本文では単にサーバーとだけ呼びます。

このサーバーは、外部ネットワークへの情報の提供、内部ネットワークへの情報の提供を同時に行なう重要なサーバーであるとともに、外部からの攻撃を受けるときの前線に立ちます。

ここではサーバーが1台であることで話を進めます。本来は、サーバー1台で複数のサービスを提供することは望ましいことではありません。「1つのバスケットに全部の卵を入れるな」という古い言葉のように、1つのサーバーが侵略されたらすべてのサービスが影響を受けるという危険があります。

しかし、現実的にはSOHO環境で何台もサーバーを用意するというのは、非常に難しいことです。もちろん、サーバーは完全にサーバーとしてのみ使われているものとします。サーバーであると同時にユーザーのマシンとしても使うようなことは絶対に避けてください。

ISPのサービスを活用してみよう

さて、このサーバーで何をサービスするかを書き出してみましょう。

ごく基本的なサービスを図2に羅列してみました。ただし、常時接続でもインターネッ

図1　SOHO環境のモデルケース

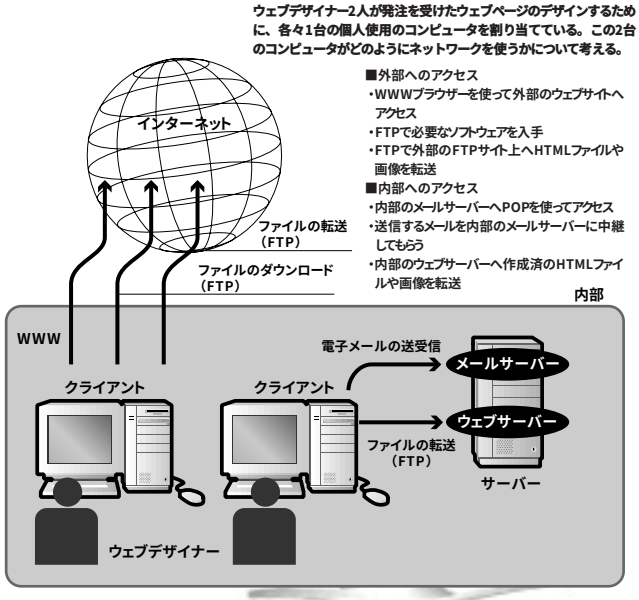
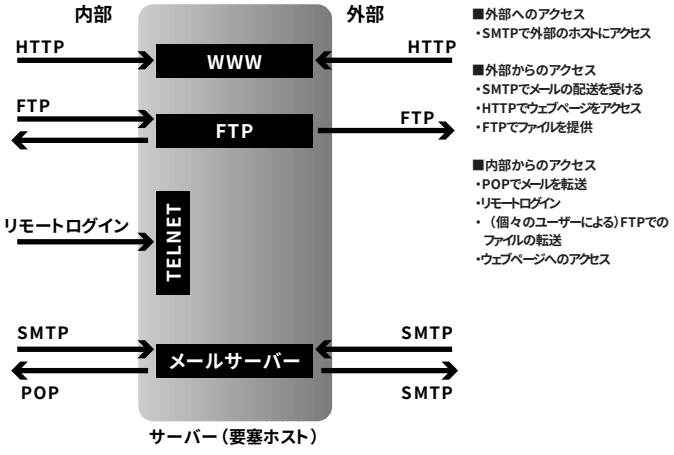


図2　サーバーでのサービス



トサービスプロバイダー（ISP）によって提供されている各種のサービスを使うと条件が変わってきます。

たとえば、図2のサービスの中にはDNS（ドメインネームサーバー）は含まれていません。DNSはホスト名とIPアドレスを相互に変換するサーバーなので、常時稼働している必要があります。SOHOレベルの常時接続サービスでは、ISP側でDNSを用意している場合が多いので、それを使うことにして、なるべく自分のサーバーの管理を軽減することも可能です。

また、ISPによってはDNSの機能だけではなく、メールサーバーを用意している場合もあります。これらは、セキュリティーを考えた場合、むしろ自分のサイト内で管理するよりもISPのサービスを使ったほうが楽な場合もあるので、十分に考慮する価値があると思います。

さて、上記のようなサービスを提供するために用いるソフトウェアをユーザーのときと同様に書き出して確認しましょう。チェックポイントは、安全性が確認されたバージョンであるかという点です。

ソフトウェアの脆弱性に関してのレポートは、CERTやJPCERTといった各国のIRT（Incident Response Teamの略で不正アクセスに対応するための組織）から出ているので、ウェブサイトなどを参考に確認してください（主要IRTのドキュメントも下記サイトに置かれています）。

URL http://www.jpcert.or.jp/

サーバーのソフトウェアで注意すべき点

本文で想定しているようなSOHO環境のサーバーでよく見られるトラブルは、ウェブサーバーのCGI-BINプログラム（以下CGIとします）です。CGIはユーザーが簡単に追加できるインターフェイスプログラムであるという特徴があります。しかし、先程サーバーで使用するソフトウェアとして列挙したものと同格に扱う必要のあるソフトウェアです。なぜならば、外部から直接CGIにアクセスされるということは、CGIプログラムにセキュリティーホールが存在した場合、サーバー全体に対する脅威に発展する可能性があるからです。

CGIプログラムは、ウェブサーバーとの間に簡便なアプリケーションインターフェイスを持っているので、簡単なものであればすぐに作成できます。そのためか、セキュリティーを考慮せずに作成されているものがあり、トラブルを起こすケースが多く見られます。

安全なネットワーク構成を考えよう

今まで調べたことをふまえて、安全なSOHOのネットワーク環境の構成を考えましょう。規模の大きいネットワーク環境とは違い、少ない資源で効果的なセキュリティーを保つことが、SOHO環境では特に強く求められます。

複数のルーターを用意できるような所は稀でしょう。SOHO用ルーターを1台使用しているのがほとんどであると思われます。そのような環境では、スクリーンドホスト構造と呼ばれるものが効果的です（図3）。ルーターが厳密に外部と内部の通信の必要なものだけを取り出します（フィルタリング）。

現在では、SOHO用ルーターはIPフィルタリングやNATなどの機能が標準的に用意されています。その機能を活用することによって、セキュリティーを高めることができます。NATは内部ネットワークで使われているプライベートなIPアドレスを、インターネット側で割り当てられているグローバルなIPアドレスに割

図3　スクリーンドホストの構造

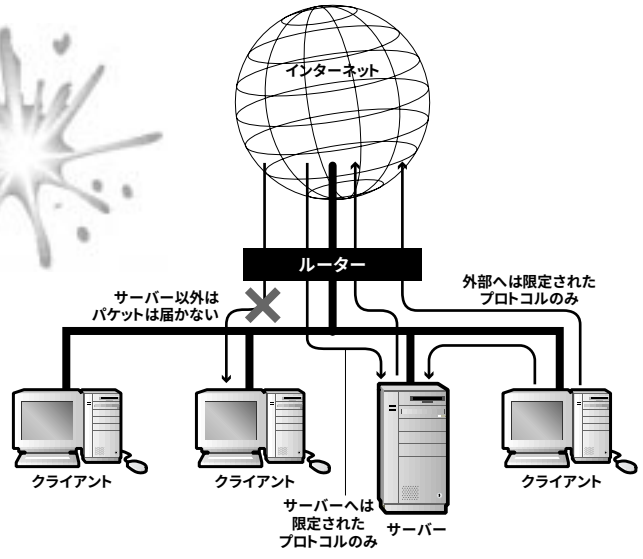


図4　SOHOルーターの設定例

```
ISPから割り当てられるグローバルIPアドレス .....133.XXX.96.8/29
サーバーのグローバルIPアドレス ..............133.XXX.96.9
サーバーのプライベートIPアドレス .............192.168.0.2

■機種A

ip nat 1 192.168.0.2/*/* 133.XXX.96.9

■機種B

nat address global 133.XXX.96.9-133.XXX.96.
14 133.XXX.96.9=192.168.0.2
```

り当てる機能です。NAT機能を使うことによって、内部のマシンは外部に接続できても、外部からは内部のマシンにアクセスできない状態を作り出せます。

この構造では、内部ネットワークのマシンで外部からのアクセスを許すのは、唯一サーバーのみです。このサーバーのみがグローバルなIPアドレスに対応してアクセスできるようにします。ルーターのNATのルールを記録するテーブルに、サーバーのグローバルなIPアドレスとプライベートなIPアドレスの対応を登録します。このテーブルに登録されていないサーバー以外のネットワーク内の機材は、割り当て対応のルールがないので、外部からアクセスできません。これらのルールの記述は、使っているルーターの機種によって異なりますので、詳しくはマニュアルをよく読んでください（図4）。

フィルタリングには「All Deny」を適用しよう

ルールを決める前に、確実で最も安全である初期設定を行います。まず最初に外部から内部へと、内部から外部へのどちらに対しても、すべての送信元、すべての送信先、すべてのプロトコル、すべての送信先ポートに対して通信を拒絶する設定を行います。これは「All Denyのルール」と呼ばれる設定です。こうしておけば、万が一設定を忘れても、すべての通信が閉ざされているため、大きな被害に結び付くことはありません。

次に、サーバーに関するルールを設定します。

・外部から SMTP、HTTP、FTP
・外部へ　SMTP

内部から外部へ攻撃するわけではないので、内部からは自由にアクセスできてかまわないように思われるでしょう。内部から外部へのアクセスを限定するのは、内部になんらかの形で送り込まれて内部の情報を勝手に持ち出すようなトロイの木馬タイプの攻撃に対する対応です。サーバー内にあるトロイの木馬タイプのソフトが、サーバーから外部へネットワーク接続を行うようなコマンドが実行された場合にサーバーを守ることができます。

次に、ユーザーが使うマシンに関するルールを設定します。

・外部へ HTTP、FTP

二重三重の防御を施してミスを防ごう

ルーターでフィルタリングすると同時にサーバー自身にも防御を施します。ルーターのフィルターが無効になった場合でも、二重三重の防御を施すことによってさらに安全性が高まります。人間が操作を行う以上、オペレーションミスは付きものです。新しい機材が入ったり、あるいはソフトウェアが増えたためにルーターのルールを書き換えようとしたりするとき、うっかりミスで無効にしてしまう場合がままあります。同じミスでも、接続できなくなった場合はユーザーがネットワーク資源を使えなくなるような不具合が発生するので問題は発見できます。しかし、「接続できる」場合は、通常ユーザーにはなんの不具合も発生しないので、長い間気が付かないままになってしまうことがあります。ミスとは発生しやすいものですし、簡単に見逃してしまうものです。したがって、二重三重の防御を施したほうが確実になります。

セキュリティーのためのソフトを使おう

まず、そのままではパスワードを見ることができないシャドウパスワードという仕組みが用意されているシステムであれば、これを使います。

できることなら、UNIXのリモートログインのサービスであるtelnetdやrlogind、またrshdといったリモートシェルを無効にし、代わりにSSHのような通信を暗号化するようなセキュリティーの高いサービスに入れ換えて使用することが望ましいといえます。ただし、ユーザーの使うマシン上で適当なSSHクライアントを用意できない場合があります。

接続を監視するフィルターをサーバーにも設定することによって、さらに安全を高めることができます。たとえば、フリーソフトのTCP Wrapperなどが有名です。これは、UNIXの上でフィルターとしてよく使われているものです。接続のフィルタリングだけではなく、詳しい接続記録を取れるようになります。

既存のＦＴＰサーバーやＰＯＰなどは、inetd.confというファイルでTCP Wrapperを使えるように設定できます。あるいは、最近のｓｅｎｄｍａｉｌのようにコンパイル時にtcpwrapperのライブラリーを指定し、内部に組み込むタイプのものもあります。

ＦＴＰやウェブサーバーの設定に気を付けよう

外部に公開しているFTPサイトはさまざまな注意が必要です。常に、バグのない最新のFTPサーバーを使うようにしてください。参照するディレクトリーのユーザーの権限の許可に注意してください。ユーザーの権限を間違えて設定すると、外部からの侵入に使われたり、ファイルを破壊されたり、あるいは不正中継に使われたりする危険があります。

ウェブサーバーが参照するファイルのユーザーIDは、CGIを実行するときに使われる実行時IDと異なるユーザーIDにしておく必要があります。さらに、このファイルに対して書き込み不可というユーザーの権限を与えておくとよいでしょう。また、CGIで入力されたデータで外部に対しては隠しておくべきものを、HTTP経由で参照できるような、たとえばHTMLファイルが納められるディレクトリーに一緒に保存するようなことは、絶対にしてはなりません。それはほかのHTMLファイルと同じように、データをサーバー上で公開したことになります。