インターネットでの不正行為 その傾向と対策

エンドユーザーは、自分を守るために不正アクセスに関する最新知識を知っておく必要があります。それには、自分を守るための知識だけではちょっと不十分ではないでしょうか。今回は、エンドユーザーからすればちょっと裏方であるシステム側が使うセキュリティーツールのお話です。不正アクセスの危険は無数にありますが、それを防ぐ方法も無数にあるということを知っておいてください。

第8回 セキュリティーを高めるツールあれこれ

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

知っててソンはない知識

今回は、不正アクセス対策としてセキュリティーを高めるツールについていくつか紹介と説明をします。今回紹介するツールの必ずしもすべてがエンドユーザーが使うとは限りません。しかし、不正アクセスに関する周辺知識を増やし、知識を広げることによって、今までの不正アクセスに関する知識が有効に使えるようになるのではないかと期待します。

ここで取り上げているツールは、どのツールが良いとか悪いとかといった評価や、具体的な実践手順を示すものではありません。読者の方々には、「不正アクセスを検知したり予防したりするツールが存在し、それが有効に活用できる」という認識を持ってもらえればよいかと思います。

ここで紹介するツールは、UNIX上で動作し、ftpで取得可能なツールです。本稿は、これらのツールの紹介が目的であり、これらのツールを推薦するものではありません。

紹介するツールは、特に断りがなければ次のURLから入手できます。

URL ftp://info.cert.org/pub/tools/

ネットワーク監視 ～TCP WRAPPER

UNIX上で動作するTCP/IPのネットワークを監視するデーモンプログラムです。systat、finger、ftp、telnet、rlogin、rsh、exec、tftp、talkや他のネットワークサービスに対してフィルタリングとモニタリングを行うプログラムです。

多くのネットワークプログラムは、サーバー＝クライアントモデルに基づいた設計になっています。ネットワーク経由で遠隔地からログインするtelnetを使っている方も多いかと思いますが、これはクライアント側で動作し、端末の機能を提供するtelnetと、サーバ

ー側でクライアントからの接続を受け持つtelnetdからできています（図1参照）。

TCP WRAPPERは、サーバー側のデーモンプログラム（サーバーで使われるソフトウェア）に手を加えることなしに、クライアントとサーバーの間の通信が開始する際に、いろいろな詳細情報を自動的に記録します。一方、使う側のクライアントは、今までの接続とまったく同じなので、TCP WRAPPERを使っているかどうかは分かりません。

いろいろな情報を自動的に記録していく以外にも、いろいろな機能があります。いくつかの例を列挙すると、特定のマシンからの接続や特定のクライアントしか接続させないようなアクセス制御、ホスト名偽造に対する制御、ネットワークアドレス偽造に対する制御などなどです。基本的な不正アクセスに対して確実に対応するだけではなく、TCP/IPのシーケンスナンバー偽造のような高度な不正アクセスに対応しています。

ログファイル管理
～swatch

SWATCH（Simple WATCHer program）は、ログファイルを加工したり、モニターしたりするのを簡単に行えるツールです。ログファイルに記録される情報のうち、特定の情報に対して特定のアクションを指定することができます。たとえば、不正アクセスらしきログデータが発生するとすぐにポケベルで管理者へ伝えるというような設定が可能です。SWATCHをTCP WRAPPERと組み合わせることにより、より強力に不正アクセスを監視することが可能になります。たとえば、TCP WRAPPERでTCP/IPのシーケンスナンバー偽造を検知してログに書き込むと、今度はSWATCHがそのログを感知して、管理者へポケベルで警告するといったようなことが可能になります。

また、不正アクセスだけではなく、システムの異常を迅速に管理者へ伝えるためにも活用できる便利なツールでもあります。

URL ftp://ftp.stanford.edu/general/security-tools/swatch/

パスワード暗号化
～シャドウパスワード

ここでのシャドウパスワードとは、具体的なツールを指しているのではなく、シャドウパスワードを採用しているシステムかどうかを意味しています。伝統的なUNIXのパスワードファイルは、ユーザーアカウントと同時にユーザーのパスワードが暗号化された形で（正確には特殊なハッシュ値といいます）が格納されています。

この暗号化済みパスワードは、その値から元のパスワードを求めることはできません。そこで、パスワードの候補となる文字列を暗号化関数で処理し、そのデータを暗号化済みパスワードと同じかどうか比較します。以前なら、非常に時間がかかる処理だったので、そのためパスワードファイルを一般ユーザーに公開しても、大きな脅威にはなりませんでした。

現在では、コンピュータの計算速度の向上により暗号化関数を使っての時間当たりの試行回数が飛躍的に増え、英単語辞書や人名辞書に載っているような、いわゆる「弱いパスワード」の利用者だと数分～数十分、5文字以下の英単語を2つ並べたようなパスワードでも数十分～数時間、英字小文字からなる7文字のパスワードだと数日～数十日で解かれてしまう危険性があります。

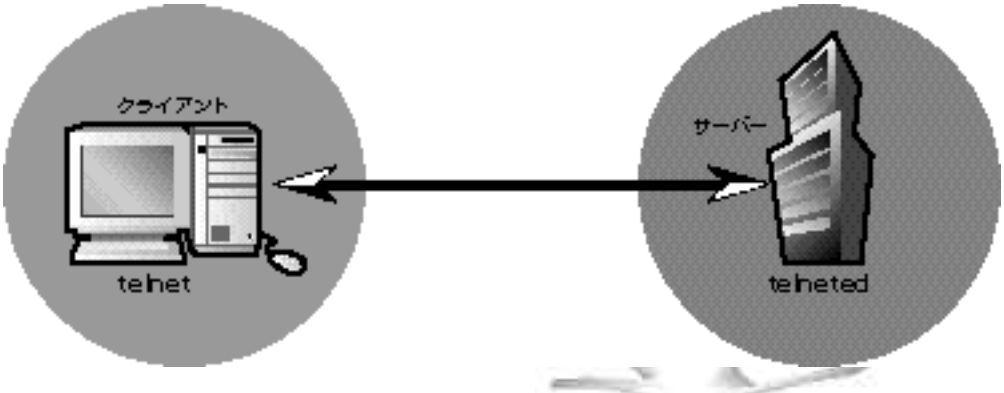
しかし、暗号化済みパスワードが見られなければ、パスワードの候補を暗号化関数で処理して比較するという処理が不可能になります。このような理由により、現在では、暗号済パスワードを通常の一般ユーザーに参照できるパスワードファイルから一般ユーザーが参照できないシャドウパスワードのファイルへ移動させるタイプのシステムがほとんどを占めています。

このように、シャドウパスワードが利用できる環境であれば、パスワードを破ろうとするものに、簡単には情報を与えないようにすることができます。

データチェック
～MD5

MD5は、一方向性ハッシュ関数MD5の名前です。一方向性ハッシュ関数とは、入力したデータを計算し、一定の長さの値を出力す

図1



る機能を持つ関数で、出力された値からは入力されたデータは逆算できないような構造になっています。

MD5は、チェックサムとしてデータの一貫性をチェックするのに使われますが、MD5は、暗号技術レベルでの安全性を持つため、通常のデータ一貫性のチェックよりも厳密にチェックができます。

電子署名
～PGP

PGPは、公開暗号方式および慣用暗号方式による暗号化、ならびに電子署名が行えるツールですが、ここでは電子署名について取り上げます。

紙の上に書かれた文字とは違い、電子的なテキストは、一部を書き換えても、その形跡は残らないため、内容の書き換え以前と書き換え以降の違いを発見できません。そこで必要になるのが電子署名です。

電子署名には2つ利点があります。まず1つは、データの一貫性をチェックすることによって内容の書き換えが行われていないかをチェックできることです。もう1つは、署名鍵に付属している所有者情報から誰が署名を行ったのかが分かることです。

CERT/CCやJPCERT/CCが公開しているドキュメントにもPGPを用いての電子署名が行われています。ユーザーが電子署名を使ってチェックできるようにしていれば、JPCERT/CCのドキュメントを改ざんして配布したり、あるいはJPCERT/CCをかたって偽ったドキュメントを作成して配布したりするような攻撃を避けることができます。

ドキュメントの場合、通常テキストのまま電子署名を行います。図2を参照してください。

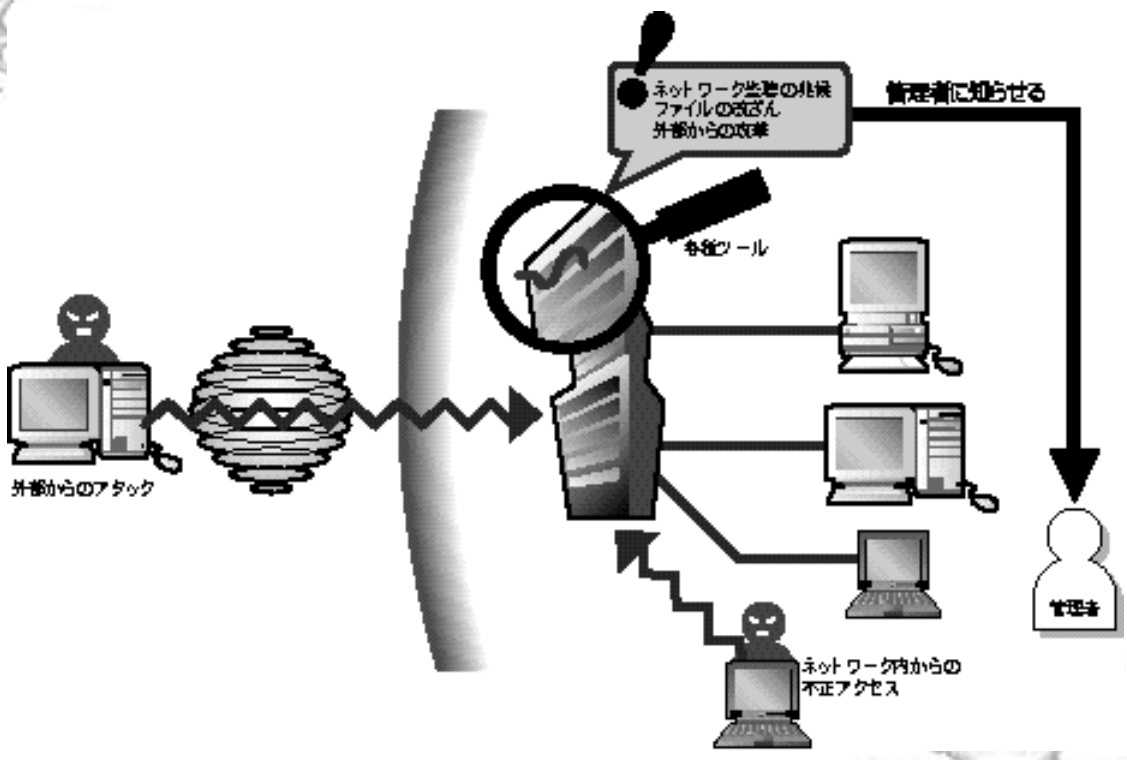
既存のバイナリーファイルに対して電子署名を行う場合、電子署名を行ったデータを別ファイルにセーブしている場合が多く、その場合は、チェックするためのバイナリーファイルと電子署名が含まれているファイルの2つを使用しなければなりません。電子署名には、いろいろな方法があります。そのつど、配布時に付属のドキュメントなどが用意されていると思いますので、確認して使用してください。

URL ftp://ftp.jpcert.or.jp/pub/

なお、JPCERT/CCでのドキュメントに対するPGP署名に関しては、最後に述べますのでそちらを参考にしてください。

データの一貫性チェック
～Tripwire

Tripwireは、ディレクトリやファイルのインティグリティチェック（一貫性チェック）を行うツールです。Tripwireは、あらかじめ登録しておいたファイルやディレクトリーに対して一方向性ハッシュ関数の処理を行い、ハッシュ値など各種の情報のデータベースを作ります（この記録は改ざんされないようにプロテクションされていなければなりません）。次に、Tripwireで検査を行うときに、検査対象となっているディレクトリーやファイルをデータベースの情報と照合し、改ざんが行われ



た場合、ただちに警告します。

不正アクセスによる、不正なファイル改ざん、削除などの発見や対応に効力を発揮します。

ステータスチェック ～ifstatus

ifstatusは、ネットワークインターフェイスのステータスをチェックするプログラムです。不正アクセスを行った侵入者が、ネットワークインターフェイスをデバッグモードやネットワークをモニターできるモード（プレミスクエスモード）で使用していたりした場合、警告します。cronなどを用いて一定時間ごとに自動チェックするようにします。

不正アクセスを行った侵入者は、さらにネットワーク上で盗聴するプログラムを仕掛けるために、ネットワークインターフェイスをネットワークをモニターできるモードで使用します。そのため、デバッグモードやネットワークをモニターできるモードで使用されている場合、不正アクセスを行っている可能性があります。その兆候をいち早くチェックするためのツールです。

図2：電子署名が入ったメールの例

```
=======

-----BEGIN PGP SIGNED MESSAGE-----

本文がここに記述されます。

-----BEGIN PGP SIGNATURE-----
Version: 2.6.2

iQCVAwUBNJqbZXVP+x0t4w7BAQFw1gP/a1k
4yDO7n4avxiaH3DkxHT2reC55uC8a
h+Z3qftD/EycwSXclzMZJnGcjlz22K2YCK1
wwjl53KR4HwQSUv8IJuwu8scMVDhL
Z59EPzqAXEQZOKjic7rtLZwembSWakgLgwY
PuT/3jMdmPzFZmatUw4zWVEgtRaIO
v+cUHc9Somo=
=Q276
-----END PGP SIGNATURE-----
=======
```

コマンド制限 ～smrsh

sendmail用の制限つきshellです。sendmailを設定するとき、通常のshの代わりに、smrshを設定します。UNIXのメールでは、.forwardの中に“| program”という記述ができ、任意のプログラムを実行することが可能ですが、これが悪意による攻撃に使われる可能性があります。smrshを使用すれば、/bin/adm/sm.bin（これはデフォルトです）のディレクトリーの中にあるコマンドしか実行できなくなります。つまり、システム側で用意した特定のコマンドしか実行ができなくなります。これにより.forwardを使用した悪意の攻撃に対応できるようになります。

通常のコマンドでの検査

通常のコマンドを使っても不正アクセスの兆候を探ることができます。例えば、不正アクセスを行うための、ルートの実行権限を持ったコマンドが隠されていないかを探すような場合は、ルート権限でfindを実行します。

コマンドの実行例

```
% find / -uid 0 -perm +4000 -print
```

ただし、この方法には事前にルートでの実行権限を持ったコマンドのリストを作成し、そのリストと比較することが必要です。また既存のコマンドに見せかけるような抜け道を作らないために、ルートでの実行権限を持った既存のコマンドに対してTripwireを使用するなども併用しなければいけません。

まとめ

既存の入手可能なツールを活用することによって、かなり強力にサイトを不正アクセスから防御したり、あるいは、不正アクセスの兆候や不正アクセスを受けたことを警告したりするといったことが可能になります。

管理のための手間が増えることは確かですが、不正アクセスを受けての深刻な被害を想定した場合を考えれば、決して無駄な手間にはならないと考えます。

サイトを管理しない方々も、このような対抗手段があることを知って、不正アクセスは、きちんと対応さえとっていれば、被害にあわないあるいは被害を最小限に抑えることができるのだということを認識して頂ければと思います。

JPCERT/CCの電子署名

さて、最後になりますが、JPCERT/CCの一部のドキュメントに対してPGPを用いての電子署名を行なっています。JPCERT/CCの電子署名であるかどうかをチェックするため公開鍵は、次のURLから入手できます。

URL http://www.jpcert.or.jp/jpcert-key.txt

この公開鍵が本物であるかどうかをチェックするためにPGPはKey fingerprint（鍵の指紋）という値を用意しています。JPCERT/CCのKey fingerprintは次の通りです。

```
Key fingerprint = BA F4 D9 FA B8
FB F0 73 57 EE 3C 2B 13 F0 48 B8
```